

# ЕКОНОМІЧНА ІНФОРМАТИКА

## Центральний офіс

INTERNET  
"зовнішнє"  
підключення

маршрутизатор

ШЛЮЗ

## Журнал адміністратора



журнал подій

керування шлюзами



керування маршрутизаторами

маршрутизатор



робочі станції

робочі станції

робочі станції

сервер

## Віддалений офіс

INTERNET

ШЛЮЗ



робочі станції

робочі станції

робочі станції

сервер

- Модуль фільтрації потоків
- Лист доступу маршрутизатора

Є.В.Кочура, І.М. Пістунов, Т.В. Борщ, Ю.І. Рогов

2007

Міністерство освіти і науки України  
Національний гірничий університет

**Є.В. Кочура, І.М. Пістунов, Т.В. Борщ, Ю.І. Рогов**

# **ЕКОНОМІЧНА ІНФОРМАТИКА**

Навчальний посібник

Дніпропетровськ  
НГУ  
2007

**УДК 33:004(075.8)**

**ББК 65:32.97я73**

**E45**

Затверджено вченою радою університету як навчальний посібник по дисципліні „Економічна інформатика” для студентів очної та заочної форм навчання в циклі професійної підготовки бакалавра за напрямками підготовки 0501 Економіка і підприємництво та 0502 Менеджмент (Протокол № від                   ).

Рецензенти:

*О.М.Марюта*, д-р техн. наук, проф., завідувач кафедри економічної інформатики і статистики (Дніпропетровський національний університет);

*Б.І.Мороз*, д-р. техн. наук, проф., начальник кафедри інформаційних систем і технологій (Академія митної служби України).

Автори:

Є.В. Кочура, вступ, розділи 1-2, висновки

І.М. Пістунов, розділи 3-7, 12

Т.В. Борщ, розділ 9

Ю.І. Рогов, розділи 3, 8, 10-11

**Економічна інформатика:**

**E45** Навч. посібник/ Є.В. Кочура, І.М. Пістунов, Т.В. Борщ, Ю.І. Рогов. – Дніпропетровськ: Національний гірничий університет, 2007.– 238 с.

У посібнику розглянуто теоретичні і практичні аспекти застосування сучасних набутоків інформатики для автоматизації офісної діяльності у всіх її аспектах. Наведено інформацію про можливості для пересічного користувача персонального комп'ютера у створенні баз даних, веб-сторінок, обчислювальних програм тощо. В посібнику подано завдання для самостійного виконання, тому він може слугувати і для практичних чи лабораторних занять.

Посібник базується на літературних джерелах вітчизняних та зарубіжних авторів та на досвіді викладання дисциплін „Інформатика та комп'ютерна техніка”, „Комп'ютерні мережі та телекомунікації”, „Web-дизайн та програмування”, „Проектування баз даних”, „Захист інформації”, „Електронна комерція”, „Програмні оболонки і пакети”, „Інформаційні системи і технології в економіці” в Національному гірничому університеті. Призначено для студентів вищих учбових закладів і може бути корисним для економістів, плановиків, менеджерів та маркетологів.

**ББК 65:32.97я73**

© Є.В. Кочура, І.М. Пістунов, Т.В. Борщ, Ю.І. Рогов, 2007

© Національний гірничий університет, 2007

# ЗМІСТ

ВСТУП.....	9
1. ОСНОВНІ ОБ'ЄКТИ І МЕТОДИ ВИВЧЕННЯ НАУКИ ІНФОРМАТИКИ.....	13
2. ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ.....	16
2.1. Історія розвитку інформатики.....	16
2.2. Основні підходи до визначення поняття "інформація" .....	17
2.3. Носії інформації. Види і властивості інформації .....	17
2.4. Вимірювання інформації.....	20
2.5. Теорія інформації.....	22
2.6. Надійність інформації.....	22
2.7. Тривалість передачі інформації .....	23
2.8. Індивідуальне завдання №1. Розрахунок обсягу інформації.....	24
3. ТЕХНІЧНІ ЗАСОБИ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ.....	25
3.1. Принципи архітектури і історія розвитку комп'ютерів .....	25
3.1.1. Архітектура комп'ютера .....	25
3.1.2. Основні тенденції розвитку апаратного забезпечення комп'ютера .....	27
3.1.3. Магістрально-модульний принцип архітектури персональних комп'ютерів.....	28
3.1.4. Покоління ЕОМ і їх основні характеристики.....	28
3.2. Представлення інформації в комп'ютері.....	30
3.2.1. Представлення текстової інформації.....	30
3.2.2. Представлення графічної інформації.....	33
3.2.3. Представлення звукової інформації.....	36
3.2.4. Представлення числової інформації.....	37
4. МЕРЕЖНІ ТЕХНОЛОГІЇ.....	39
4.1. Локальні мережі.....	39
4.1.1. Апаратне забезпечення локальних мереж.....	39
4.1.2. Топологія мереж.....	40
4.1.3. Поняття сервера та робочої станції .....	42
4.1.4. Команди настройки локальної мережі засобами Windows .....	43

4.1.5 Програми обміну письмовими повідомленнями поміж різними робочими станціями локальної мережі.....	45
4.1.5.1. WinPopUp.....	45
4.1.5.2. Intranet Chat.....	46
4.2. Глобальні мережі.....	47
4.2.1 Класифікація мереж .....	47
4.2.2. Апаратне забезпечення глобальних мереж .....	48
4.2.3. Відмінність в режимах роботи глобальних мереж “off-line” та “on-line” .....	48
4.2.4. Основні поняття і протоколи Internet .....	49
4.2.5. IP-адресація комп'ютерів у Internet.....	50
4.2.6. Постачальник послуг Інтернету (ISP або Провайдер).....	52
4.2.7. Способи з'єднання з Інтернетом .....	52
4.2.8 Доменна система адрес в Інтернеті .....	54
4.3. Індивідуальні завдання №2 .....	56
4.4. Індивідуальні завдання №3 .....	57
5. ЗАСТОСУВАННЯ ІНТЕРНЕТУ В ЕКОНОМІЦІ .....	59
5.1. Браузери Інтернету .....	59
5.1.1. Internet Explorer .....	59
5.1.2. Netscape Navigator .....	60
5.1.3. Mozilla Firefox.....	62
5.1.3.1. Інтерфейс .....	62
5.1.3.2. Основні прийоми роботи з браузером Mozilla Firefox.....	62
5.1.4 Opera .....	66
5.1.4.1. Інтерфейс браузера Opera .....	66
5.1.4.2. Ключові особливості браузера .....	66
5.2. Мейлери Інтернету .....	68
5.2.1. Програма Outlook Express.....	70
5.2.1.1. Інтерфейс Outlook Express.....	70
5.2.2. Програма The Bat .....	74
5.2.3. Поштовий агент браузера Opera.....	75
5.3. Можливості Інтернету .....	75
5.3.1. Віртуальна електронна пошта .....	75
5.3.1.1. Створення поштової скриньки .....	76
5.3.2. Пошук інформації в Інтернеті. Пошукові сервери .....	77
5.3.2.1. Українські пошукові сервери.....	78
5.3.2.2. Російські пошукові сервери .....	78

5.3.2.3. Міжнародні пошукові сервери .....	79
5.3.3. Поняття “чату” і форуму в Інтернеті.....	80
5.3.3.1. Чат та форум на сайтах .....	81
5.3.3.2. Програмно забезпечений чат .....	82
5.3.4. Програма ICQ.....	83
5.3.5. Безперервне отримання інформації (“викачування”).....	84
5.3.5.1. Програма Net Vampire .....	84
5.3.5.2. Програма REGET .....	86
5.3.5.3. Програма Opera .....	87
5.4. Індивідуальне завдання № 4 .....	88
5.5. Індивідуальне завдання №5 .....	88
5.6. Індивідуальні завдання №6 .....	89
6. БЕЗПЕКА ТА ЗАХИСТ КОМП’ЮТЕРНИХ МЕРЕЖ .....	90
6.1. Огляд шкідливих програм.....	90
6.1.1. Комп’ютерні віруси .....	91
6.1.2. Троянські програми .....	92
6.1.3. Мережні хробаки.....	93
6.2. Технології інформаційної безпеки.....	93
6.2.1. Захист операційних систем.....	93
6.2.2. Налаштування параметрів безпеки в браузері Internet Explorer .....	94
6.2.3. Антивіруси.....	95
6.2.4. Міжмережні екрани .....	97
6.2.5. Системи контролю змісту та антиспаму .....	99
6.2.6. Ідентифікація й аутентифікація.....	101
6.3. Індивідуальне завдання № 7 .....	102
7. WEB-ДИЗАЙН.....	103
7.1. Проектування структури сайту .....	104
7.2. HTML-коди .....	106
7.2.1. Основні теги.....	109
7.2.2. Спеціальні символи і колір.....	115
7.2.3. Приклади Web-сторінок .....	116
7.3. Таблиці стилів CSS .....	118
7.3.1. Базові поняття. Специфікація CSS.....	119

7.3.2. Включення в HTML.....	120
7.3.3. Групування.....	120
7.3.4. Успадкування.....	121
7.3.5. Клас як селектор.....	121
7.3.6. ID як селектор.....	122
7.3.7. Основні параметри таблиці стилів.....	122
7.4. JavaScript.....	124
7.4.1. Розміщення скриптів.....	124
7.4.2. Приклади застосування JavaScript.....	124
7.5. Програми для створення html-сторінок.....	125
7.5.1. Візуальні редактори html-сторінок.....	125
7.5.2. Редактори html-кодів.....	128
7.6. Публікація Web-вузла в Internet.....	130
7.7. Індивідуальні завдання №8.....	132
7.7.1. Створення HTML-сторінок за допомогою Netscape Composer, Front Page, Word і порівняння їх можливостей з особливостями роботи браузерів.....	132
7.7.2. Особливості інтерфейсу Web-редактора Macromedia Homesite 5.0.....	133
7.7.3. Проектування структури сайту та оформлення тексту на Web-сторінці.....	133
7.7.4. Розміщення гіперпосилань та графічних елементів на Web-сторінках.....	134
7.7.5. Створення таблиць у Web-документах. Управління розміщенням елементів Web-сторінки за допомогою таблиць. Оформлення Web-сторінок з використанням стилів.....	135
7.7.7. Публікація Web-вузла в Internet.....	135
<b>8. ПРОГРАМНІ ЗАСОБИ ДЛЯ РОБОТИ ЗІ СТРУКТУРОВАНИМИ ДОКУМЕНТАМИ.....</b>	<b>137</b>
8.1. Програмні засоби для роботи з текстом.....	137
8.2. Програмні засоби для обчислювальних робіт.....	138
8.3. Програмні засоби роботи з графікою.....	141
8.4. Програмні засоби роботи із звуком.....	143

9. ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ .....	146
9.1. Основні поняття реляційних баз даних .....	146
9.2. Складові частини SQL .....	147
9.3. Типи даних SQL.....	148
9.3.1. Символьні функції.....	148
9.3.2. Числові типи даних .....	148
9.3.3. Дата і час .....	149
9.3.4. Невизначені або пропущені дані (NULL) .....	150
9.4. Використані терміни і позначення .....	150
9.5. Навчальна база даних.....	151
9.6. Найпростіші SELECT-запити .....	152
9.7. Оператори IN, BETWEEN, LIKE, is NULL .....	155
9.8. Перетворення висновку і вбудовані функції .....	157
9.9. Агреговані і групові функції.....	158
9.10. Порожні значення (NULL) у функціях .....	158
9.11. Результат дії трізначних умовних операторів .....	159
9.12. Упорядкування вихідних полів (ORDER BY).....	161
9.13. Вкладені підзапити .....	162
9.14. Формування зв'язаних під запитів.....	163
9.15. Зв'язані підзапити в HAVING .....	164
9.16. Використання оператора EXISTS .....	164
9.17. Особливості застосування операторів ANY, ALL, EXISTS при обробці порожніх значень (NULL) .....	165
9.18. Використання COUNT замість EXISTS .....	166
9.19. Оператор об'єднання UNION .....	166
9.20. Усунення дублювання в UNION.....	166
9.21. Використання UNION з ORDER BY .....	167
9.22. Зовнішнє об'єднання.....	169
9.23. Команди маніпулювання даними.....	169
9.24. Використання підзапитів у INSERT.....	170
9.25. Індивідуальне завдання №9. Вивчення середовища СУБД Access .....	171
9.26. Індивідуальне завдання №10. Використання SQL для розв'язувань задач БД свого варіанту.....	172
10. ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ .....	179
10.1. Дані VBA: типи даних, змінні і константи .....	179
10.1.1 Змінні і константи .....	180



10.1.2 Типи даних.....	183
10.2. Синтаксичні конструкції мови VBA .....	188
10.2.1 Вирази і операції .....	188
10.2.1.1. Арифметичні операції.....	188
10.2.1.2. Логічні операції.....	189
10.2.1.3. Операції порівняння.....	189
10.2.2 Конструкції мови, що управляють.....	191
10.3. Основні прийоми роботи з редактором Visual Basic .....	195
10.4. Індивідуальне завдання №11 .....	199
11. ЕКСПЕРТНІ ТА НАВЧАЛЬНІ СИСТЕМИ.....	201
11.1. Системи з інтелектуальним інтерфейсом .....	201
11.2. Експертні системи (ЕС) .....	202
11.2.1. Особливості та класифікація ЕС .....	202
11.2.2. Структура ЕС .....	205
11.2.3. Технологія проектування і розробки ЕС .....	205
11.3. Самонавчальні системи .....	207
11.4. Адаптивні інформаційні системи .....	210
12. ПОНЯТТЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	211
12.1 Етапи розвитку інформаційних технологій .....	211
12.2 Інструментарій інформаційної технології.....	213
12.3. Інформаційна технологія опрацювання даних .....	214
12.4. Інформаційна технологія керування .....	215
12.5. Інформаційна технологія підтримки прийняття рішень .....	216
12.6. Інформаційна технологія експертних систем.....	217
12.7. Небезпека та труднощі використання інформаційних технологій.....	218
ПІДСУМКИ.....	219
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ .....	220
ПРЕДМЕТНИЙ ПОКАЖЧИК .....	223
ДОДАТОК А. Словник спеціальних термінів.....	225
ДОДАТОК Б. Фінансові ресурси Інтернету.....	233

# ВСТУП

Інформація - явище незрівнянно більш древнє, ніж сама людина. Уже природа у процесі своєї еволюції передавала закодовану інформацію в рослинах і живих організмах. З перших своїх кроків люди шукають і знаходять нові засоби передачі, збереження та обробки інформації. Однак ніколи раніше людство не накопичувало інформацію й знання настільки стрімкими темпами. Тому закономірним є те, що жодна галузь людської діяльності не зазнала такого розвитку як інформаційні технології.

Діяльність окремих людей, груп, колективів і організацій зараз усі в більшому ступені починає залежати від їхньої інформованості і здатності ефективно використовувати наявну інформацію. Відшукання раціональних рішень у будь-якій сфері вимагає обробки великих обсягів інформації, що часом неможливо без залучення спеціальних технічних засобів.

Зростання обсягу інформації особливо стало помітно в середині ХХ в. Лавиноподібний потік інформації впав на людину, не даючи йому можливості сприйняти цю інформацію повною мірою. У новому потоці інформації, що з'являється щодня, орієнтуватися стає усе важче. Часом вигідніше стало створювати новий матеріальний або інтелектуальний продукт, ніж вести пошук аналога, зробленого раніше.

Утворення великих потоків інформації обумовлюється:

- надзвичайно швидким ростом числа документів, звітів, дисертацій, доповідей і т.п., у яких викладаються результати наукових досліджень і дослідно-конструкторських робіт;
- постійно зростаючим числом періодичних видань у різних областях людської діяльності;
- появою різноманітних даних (метеорологічних, геофізичних, медичних, економічних і ін.), записуваних звичайно на магнітних стрічках і тому не попадають у сферу дії системи комунікації.

В історії розвитку цивілізації відбулося кілька інформаційних революцій перетворень суспільних відносин через кардинальні зміни в сфері обробки інформації. Наслідком подібних перетворень було придбання людським суспільством нової якості.

*Перша* революція зв'язана з винаходом писемності, що привело до гігантського якісного і кількісного стрибка. З'явилася можливість передачі знань від покоління до поколінь.

*Друга* (середина ХVІ в.) викликана винаходом друкарства, що радикально змінив індустріальне суспільство, культуру, організацію діяльності.

*Третя* (кінець ХІХ в.) обумовлена винаходом електрики, завдяки якому з'явилися телеграф, телефон, радіо, що дозволяють оперативно передавати і накопичувати інформацію в будь-якому обсязі.

*Четверта* (70-і рр. ХХ в.) зв'язана з винаходом мікропроцесорної технології і появою персонального комп'ютера. На мікропроцесорах і інтегральних схемах створюються комп'ютери, комп'ютерні мережі, системи передачі даних (інформаційні комунікації). Цей період характеризують три фундаментальні інновації:

- перехід від механічних і електричних засобів перетворення інформації до електронного;
- мініатюризація усіх вузлів, пристроїв, приладів, машин;
- створення програмно-керованих пристроїв і процесів.

Остання інформаційна революція висуває на перший план нову галузь - інформаційну індустрію, пов'язану з виробництвом технічних засобів, методів, технологій для виробництва нових знань. Найважливішими складовими інформаційної індустрії стають усі види інформаційних технологій, особливо телекомунікації.

Сучасна інформаційна технологія спирається на досягнення в області комп'ютерної техніки і засобів зв'язку.

Ускладнення індустріального виробництва, соціального, економічного і політичного життя, зміна динаміки процесів у всіх сферах діяльності людини привели, з одного боку, до росту потреб у знаннях, а з іншої до створення нових засобів і способів задоволення цих потреб. Бурхливий розвиток комп'ютерної техніки й інформаційних технологій послужило поштовхом до розвитку суспільства, яке побудоване на використанні різної інформації й отримало назву - інформаційне суспільство.

Сучасні вчені вважають, що в інформаційному суспільстві процес комп'ютеризації дасть людям доступ до надійних джерел інформації, позбавить їх від рутинної роботи, забезпечить високий рівень автоматизації обробки інформації у виробничій і соціальній сферах. Рушійною силою розвитку суспільства повинно стати виробництво інформаційного, а не матеріального продукту. Матеріальний же продукт стане більш інформаційно ємним, що означає збільшення частки інновацій, дизайну і маркетингу в його вартості.

В інформаційному суспільстві зміняться не тільки виробництво, але і весь уклад життя, система цінностей, зросте значимість культурного дозвілля стосовно матеріальних цінностей. У порівнянні з індустріальним суспільством, де усе спрямовано на виробництво і споживання товарів, в інформаційному суспільстві виробляються і споживаються інтелект, знання, що приводить до збільшення частки розумової праці. Від людини буде потрібно здатність до творчості, зросте попит на знання. Матеріальною і технологічною базою інформаційного суспільства стануть різного роду системи на базі комп'ютерної техніки і комп'ютерних мереж, інформаційній технології, телекомунікаційному зв'язку. Інформаційне суспільство - суспільство, у якому більшість працюючих зайнята виробництвом, збереженням, переробкою і реалізацією інформації, особливо вищої її форми знань.

У реальній практиці розвитку науки і техніки передових країн наприкінці ХХ в. поступово здобуває зримі обриси створена теоретиками картина інформаційного суспільства. Прогнозується перетворення усього світового простору в єдине комп'ютеризоване й інформаційне співтовариство людей, що проживають в електронних квартирах і котеджах. Будь-яке житло оснащено всілякими електронними приладами і комп'ютеризованими пристроями.

Діяльність людей буде зосереджена головним чином на обробці інформації, а матеріальне виробництво і виробництво енергії буде покладено на машини. Вже опублікований ряд фактичних матеріалів, що свідчать, що це не утопія, а неминуха реальність недалекого майбутнього.

Приклад 1.1. За даними соціологічного дослідження, проведеного в США, уже зараз 27 млн. працюючих можуть здійснити свою діяльність, не виходячи з будинку, а 1/3 усіх недавно зареєстрованих фірм заснована на широкому використанні самостійної зайнятості. У США до категорії самостійно зайнятих були віднесені: у 1980 р. 5,7 млн. чоловік, у 1989 р. 14,6 млн., а в 1995 р. 20,7 млн. чоловік.

При переході до інформаційного суспільства виникає нова індустрія переробки інформації на базі комп'ютерних і телекомунікаційних інформаційних технологій. Ряд учених виділяють характерні риси інформаційного суспільства:

- вирішено проблему інформаційної кризи, тобто дозволене протиріччя між інформаційною лавиною й інформаційним голодом;
- забезпечено пріоритет інформації з порівняння з іншими ресурсами;
- головною формою розвитку стане інформаційна економіка;
- в основу суспільства будуть закладені автоматизовані генерація, збереження, обробка і використання знань за допомогою новітньої інформаційної техніки і технології;
- інформаційна технологія придбає глобальний характер, охоплюючи всі сфери соціальної діяльності людини;
- формується інформаційна єдність усієї людської цивілізації;
- за допомогою засобів інформатики реалізований вільний доступ кожної людини до інформаційних ресурсів усієї цивілізації;
- реалізовано гуманістичні принципи керування суспільством і впливу на навколишнє середовище.

Крім позитивних моментів прогноуються і небезпечні тенденції:

- усе більший вплив на суспільство засобів масової інформації;
- інформаційні технології можуть зруйнувати приватне життя людей і організацій;
- існує проблема добору якісної і достовірної інформації;
- багатьом людям буде важко адаптуватися до середовища інформаційного суспільства.

Існує небезпека розриву між "інформаційною елітою" (людьми, що займаються розробкою інформаційних технологій) і споживачами. Ближче усіх на шляху до інформаційного суспільства коштують країни з розвинутою інформацій-

ною індустрією, до числа яких варто віднести США, Японію, Англію, Німеччину, країни Західної Європи. У цих країнах уже давно одним з напрямків державної політики є напрямок, пов'язаний з інвестиціями і підтримкою інновацій в інформаційну індустрію, у розвиток комп'ютерних систем і телекомунікацій.

У 80-х роках відбувся цілий ряд якісних змін в ІТ. Деякі з них усвідомлювалися поступово (наприклад, розвиток архітектури і стандартів відкритих систем), інші, як феномен персональних обчислень, входили в життя набагато більш революційним шляхом. Коротко розглянемо, як ці зміни усе більш обмежували застосування класичних методів системного проектування, вимагаючи нових підходів у розробці чисто "комп'ютерних" компонентів ІС.

Нарешті, до кінця 80-х - початкові 90-х в усім світі не тільки розроблювачами, але і користувачами були усвідомлені три дійсно революційних феномени. Вони стали усе ширше входити у вітчизняну практику, якісно змінюючи діяльність комп'ютеризованих підприємств:

1. Феномен персональних обчислень, заснований на постійній доступності працівникові можливостей ЕОМ, у першу чергу - на використанні персональних комп'ютерів. Феномен полягає в тому, що в багатьох видах інформаційних, проектних і управлінських робіт зникла необхідність у працівниках-виконавцях (друкарках, креслярах, діловодах і ін.), що є посередниками між постановкою задачі і її вирішенням.

2. Феномен кооперативних технологій, що складає в комп'ютерній підтримці спільної погодженої роботи групи працівників над одним проектом. Цей феномен виник на основі суми методів, що забезпечують керування доступом членів групи до різних частин проекту, керування версіями і редакціями проектною документацією і погодженим виконанням робіт у послідовній процедурі робіт, керування рівнобіжним конструюванням і ін.

3. Феномен комп'ютерних комунікацій, що складається в різкому збільшенні можливостей обміну будь-якою інформацією. Він виник, зокрема, на основі стандартизованих протоколів обміну даними прикладного рівня в локальних і глобальних мережах. Це дозволило виключити необхідність передачі паперових документів для одержання згоди або змістовних зауважень, непотрібні переїзди для проведення нарад, забезпечити постійну готовність працівника отримати і відіслати повідомлення або інформативні записи даних поза залежністю від місця його географічного розташування й ін.

Оцінка їхнього впливу на виробничу діяльність і організаційної структури, розробка відповідних методик вироблялися не тільки за кордоном, але і вітчизняними фахівцями, хоча тоді в нас час реального застосування цих методів ще не настало.

Насправді, перспективи розвитку інформаційних комп'ютерних технологій безмежні. Саме тому нами було обрано напрямок цього посібника, в якому було зроблено спробу дати відповіді на запитання що являє собою інформаційна технологія, які існують види інформаційних технологій. Також розглядаються етапи розвитку інформаційних технологій, перспективи їх використання. Не обми-

нули своєю увагою і ті проблеми та небезпеку, які пов'язані з використанням інформаційних технологій.

# 1. ОСНОВНІ ОБ'ЄКТИ І МЕТОДИ ВИВЧЕННЯ НАУКИ ІНФОРМАТИКИ

*Вивчивши матеріали цього розділу, студент має опанувати поняття основних об'єктів і методів у інформатиці.*

Існує багато визначень інформатики як науки і учбової дисципліни. Таке різноманіття думок відображає той факт, що інформатика – це наука, що бурхливо розвивається, знаходиться в періоді становлення. Сьогодні швидко змінюються не тільки апаратні засоби і програмне забезпечення, але і методи роботи з інформацією, а також розуміння ролі інформації в ефективному використанні обмежених матеріальних ресурсів і технічних можливостей сучасного виробництва.

Щоб визначитися з основними об'єктами вивчення в інформатиці, приведемо декілька визначень інформатики, які представляють спектр основних думок.

У 1978 р. на Міжнародному конгресі з інформатики було запропоновано наступне формулювання визначення поняття "інформатика": "Поняття інформатики охоплює області, пов'язані з розробкою, створенням, використанням і матеріально-технічним обслуговуванням систем обробки інформації, включаючи машини, устаткування, математичне забезпечення, організаційні аспекти, а також комплекс промислової, комерційної, адміністративної і соціальної дії".

У наступному визначенні в явному вигляді підкреслюється подвійний характер інформатики – фундаментальність і технологічність: "Інформатика – в даний час одна з фундаментальних областей наукового знання, що формує системно-інформаційний підхід до аналізу навколишнього світу, вивчає інформаційні процеси, методи і засоби отримання, перетворення, передачі, зберігання і використання інформації. Разом з тим інформатика – це сфера практичної діяльності людини, що стрімко розвивається і постійно розширюється, зв'язана з використанням інформаційних технологій".

**Методологічне визначення: "Інформатика – комплекс наукових напрямів і наукоємких технологій, об'єднаних загальним методом дослідження – інформаційним підходом і загальним предметом дослідження – інформаційними процесами".**

На наш погляд, *інформатика – це наука про закономірності протікання інформаційних процесів в системах різної природи, про методи, засоби і технології автоматизації інформаційних процесів, про закономірності створення і функціонування інформаційних систем.*

У даному формулюванні підкреслюється, що об'єктом вивчення інформатики як науки є саме *закони, закономірності*, оскільки цілі будь-якої науки не тільки пояснювальні, але і прогностичні. Основний предмет вивчення – інформаційні процеси, але не самі по собі, а в їх прив'язці до "носія" – інформаційних систем. Прагматизм інформатики обумовлений тим, що предметом її дослідження є також методи, засоби і технології, що забезпечують ефективну організацію інформаційних процесів і їх автоматизоване виконання.

З погляду вузівської освіти, *інформатикові слід розглядати як істотний елемент гуманітарної культури людини, яка вносить вирішальний внесок до формування сучасного наукового світогляду, дає ключ до розуміння багатьох явищ нашому життю, забезпечує уміннями, необхідними для орієнтування в сучасному інформаційному просторі і успішній діяльності в ньому.*

Таким чином, **основними об'єктами** вивчення в інформатики є:

- *інформаційні об'єкти* (тексти, сукупності даних, алгоритми і т.п.), тобто інформація, зафіксована на якому-небудь носіїві;
- *інформаційні ресурси* суспільства як сукупність створених людством інформаційних об'єктів;
- *інформаційні процеси* (пошук, відбір, зберігання, передача, обробка, зберігання, кодування, захист);
- *інформаційні моделі* (зокрема, структури даних, алгоритми, програми);
- *інформаційні технології*, як сукупність методів, способів, протоколів реалізації інформаційних процесів;
- *інформаційні системи*, як сукупність інформаційних ресурсів разом з допустимими для них інформаційними процесами, що реалізуються за допомогою інформаційних технологій;
- комп'ютер і комп'ютерні системи як універсальні *засоби автоматизації* інформаційних процесів.

У будь-якому науковому дослідженні важливо не тільки те, *що* досліджується, але і те, *як* досліджується, тобто

важливо мати уявлення про основні методи дослідження, а також про специфіку застосування загальнонаукових методів в даній області знання. Загальними для всіх наук методами дослідження є спостереження, теоретичний аналіз, експеримент, зокрема обчислювальний, моделювання і ін. У кожній конкретній науці ці методи набувають своєї специфіки.

**Основними методами** дослідження в інформатиці є:

– системно-інформаційний аналіз як конкретизація системного підходу;



– інформаційне моделювання як конкретизація загальнонаукового методу моделювання;

– комп'ютерний експеримент як різновид властивого всім наукам обчислювального експерименту.

Швидке збільшення об'єму існуючої і циркулюючої в суспільстві інформації ставить сучасну людину перед проблемою уміння працювати з нею: знаходити, відбирати потрібне, зберігати, упаковувати і швидко витягувати з сховища, обробляти і перетворювати. Причому, інформація все частіше може бути представлена не тільки в текстовому, найбільш звичному вигляді, але і як відео- і аудіоматеріали, схеми і анімаційна графіка і т.п. Володіння методами, прийомами і засобами роботи з інформацією стає одним з основних професійно важливих якостей фахівця будь-якого профілю.

### **Контрольні запитання**

1. Чи входять поняття інформатики до гуманітарного циклу освіти?
2. Що таке інформатика?
3. Що таке інформаційний процес?
4. Комп'ютер і інформаційна модуль – це одне і те саме?
- 5.

*У цьому розділі студенти отримали поняття об'єктів, засобів та методів інформатики.*

## 2. ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТКИ

*Вивчивши матеріали цього розділу, студент опанує історію розвитку інформатики, поняття інформації та навіччиться розраховувати величину інформації.*

### 2.1. Історія розвитку інформатики

Інформатика як самостійна галузь наукового знання і область практичної діяльності з'явилася в другій половині ХХ в. Її витоками можна вважати:

– теорію інформації, тісно пов'язану з теорією зв'язку в технічних системах (телефон, телеграф, радіозв'язок), зокрема, математичну теорію зв'язку (Клод Шеннон, 1948 р.);

– кібернетику, що досліджує загальні закони управління в системах різної природи, основи якої заклав Ноберт Вінер (1948 р.);

– теорію автоматів, основи якої заклав Джон фон Нейман (1946 р.);

– теорію алгоритмів (Н. Вірт, Э. Дейкстра, А.П. Ершов, Д. Кнут і ін.).

Інформатику традиційно пов'язують з розвитком комп'ютерної техніки. Але комп'ютер послужив лише каталізатором тенденцій, які виникли в науці і практиці задовго до його появи. Які ж ці тенденції? Перш за все, різке наростання об'єму інформації, яка з розвитком засобів зв'язку стає доступною окремій людині. Приходить усвідомлення того, що людські можливості сприйняття і переробки інформації обмежені. В той же час науково-технічний прогрес, що супроводжується високими швидкостями, розвитком складних технічних комплексів, ставить людину в умови, коли йому необхідно навчитися швидко і безпомилково переробляти інформацію, щоб ефективно управляти технікою. Рішення часто доводиться ухвалювати в умовах дефіциту часу, які, до того ж, вони можуть бути чреваті великими наслідками. Завдання управління вимагають застосування таких засобів, які допомагають зібрати повнішу інформацію, надійно її зберігати, швидко поширювати і безпомилково обробляти. Таким засобом і став комп'ютер. По суті, інформатика покликана займатися вирішенням проблем, пов'язаних з необхідністю організації інформаційних процесів (збір, зберігання, передача, обробка, захист і т.п.) і ефективного використання інформаційних ресурсів. Комп'ютерні засоби роботи з інформацією як основний об'єкт вивчення сучасної інформатики є нерозривна єдність трьох частин – технічного, програмного і алгоритмічного забезпечення інформаційних процесів і інформаційних систем. Основне питання інформатики – які інформаційні процеси і як вони можуть бути ефективно автоматизовані? Інформатика широко використовує досягнення таких дисциплін, як математика, лінгвістика, семіотика (наука про знакові системи), логіка, computer science, робототехніка, інженерія знань і ін.

## 2.2. Основні підходи до визначення поняття "інформація"

**Інформація** – від латинського informatio – відомості, роз'яснення, виклад.

Однозначно визначити, що ж таке інформація так само неможливо, як неможливо це зробити для понять "час", "енергія", "якість" і ін.

1 Найбільш загальним **філософським** визначенням є наступне: *інформація – це відображена різноманітність, що виникає в результаті взаємодії об'єктів.*

Ми звикли до того, що мир навколо нас мінливий, ми і помічаємо його саме в процесі зміни, тобто інформація виникає, коли порушується одноманітність і це порушення якимсь чином відбивається, проявляє себе в сигналах.

2 СENS чоґо-небудь у людини часто асоціюється з впорядкованістю, організованістю. Роль "сенсу" може грати структура, спосіб взаємозв'язку. Це відображено в наступному визначенні: *інформація – це атрибут матерії, виступаючий в пасивній формі як віддзеркалення організованості (дезорганізованості) матерії, а в активній формі як засіб організації (дезорганізації) матерії.*

3 Під інформацією **в побуті** (життєвий аспект) розуміють відомості про навколишній світ і процеси, що протікають в нїм, сприймані людиною або спеціальними пристроями (С.И.Ожегов. Тлумачний словник російської мови). Щоб людина сприймала ці відомості як інформацію, вони повинні володіти новизною.

4 **В теорії зв'язку** під інформацією прийнято розуміти *будь-яку послідовність сигналів, яка зберігається, передається або обробляється за допомогою технічних засобів, не враховуючи їх сенси.*

5 **В кібернетиці** під інформацією розуміється тільки *та частина знань, яка використовується для орієнтування, активної дії, управління, тобто в цілях збереження, вдосконалення, розвитку системи.*

6 Під інформацією **в теорії інформації** розуміють не будь-які відомості, а лише ті, які знімають повністю або зменшують *невизначеність, що існує до їх отримання. Інформація – це знята невизначеність.*

7 Під інформацією **в документалістиці** розуміють все те, що так чи інакше зафіксоване в знаковій формі у вигляді документів.

8 В **інформатиці** інформацію розглядають як *продукт взаємодії зареєстрованих сигналів – даних і методів їх обробки, адекватних вирішуваним задачам.*

## 2.3. Носії інформації. Види і властивості інформації

Інформація завжди пов'язана з матеріальним носієм. **Носій інформації** – середовище для запису і зберігання інформації. Носієм інформації може бути будь-який *матеріальний предмет* (папір, камінь, стіна, поверхня дискети), *хвилі* різної природи (акустична – звук, електромагнітна – світло, радіохвиля, гравітаційна, – тиск, тяжіння), особливі *стани речовини* (концентрація молекул

в рідкому розчині, температура і тиск газу, розташування молекул в кристалі процесора і ін.).

*Спосіб передачі інформації – сигнал.*

*Інформаційний сигнал* – фізичний процес, що має для людини або технічного пристрою інформаційне значення. Він може бути безперервним (аналоговим) або дискретним.

*Аналоговий сигнал* – сигнал, що безперервно змінюється по амплітуді і в часі.

Сигнал називається *дискретним*, якщо він може приймати лише *кінцеве число значень* в певний період часу.

### **Види інформації:**

– за способом сприйняття: візуальна, аудіальна, тактильна, смакова, нюхова;

– за суспільним значенням:

– *особиста* – це знання, досвід, інтуїція, уміння, емоції, спадкова пам'ять конкретної людини;

– *спеціальна*, значуща для певної групи людей: наукова, виробнича, технічна, управлінська;

– *суспільна*, значуща для більшості членів суспільства: суспільно-політична, науково-популярна (науково осмислений досвід всього людства, історичні, культурні і національні традиції і ін.), буденна (якою ми обмінюємося в процесі повсякденного спілкування), естетична (образотворче мистецтво, скульптура, музика, театр і ін.).

– за сферою застосування інформації (економічна, географічна, соціологічна і ін.);

– за характером джерел інформації (первинна, вторинна, узагальнювальна і ін.);

Існує виділення видів інформації і по інших підставах, але в інформатиці найбільш важливим є розділення інформації за формою уявлення, а саме:

– текстова, – числова, – графічна, – звукова і їх всілякі комбінації.

Для кожного з видів інформації існують особливі методи кодування, спеціальні програмні засоби (текстові, графічні, звукові редактори, електронні таблиці і т.п.) і спеціальні пристрої (клавіатура для введення текстової і числової інформації, сканер для введення графіки і т.п.).

В даний час основною в обчислювальній техніці стає *мультимедійна* форма представлення інформації. Кольорова графіка поєднується із звуком і текстом, числові розрахунки супроводжуються діловою графікою, текст – рухомими відеозображенням і тривимірними образами.

Інформація нам потрібна для того, щоб орієнтуватися в навколишньому оточенні і ухвалювати правильні рішення. Але чи будь-яка інформація допомагає нам в цьому? Знання *властивостей інформації* може допомогти людині оцінити у кожному конкретному випадку наскільки рішення, прийняті на основі наявної інформації, можуть бути вірними.

1 Інформація *об'єктивна*, якщо вона не залежить від чиєї-єбудь думки, думки. Об'єктивну інформацію можна отримати за допомогою *справних*

датчиків, вимірювальних приладів. Але, відбиваючись в свідомості конкретної людини, інформація перестає бути об'єктивною, оскільки перетворюється (більшою чи меншою мірою) залежно від думки, думки, досвіду, знання, пристрастей конкретного суб'єкта.

2 Інформація **достовірна**, якщо вона відображає дійсне положення справ.

**Недостовірною** інформація може бути з наступних причин:

- навмисне спотворення (*дезинформація*);
- спотворення в результаті дії перешкод ("*зіпсований телефон*");
- у разі, коли значення реального факту применшується або перебільшується (*чутки, "рибальські історії", реклама, політичні дебати*).

3 Інформація **повна**, якщо її досить для розуміння і ухвалення рішення.

**Неповна** інформація може привести до помилкового висновку або рішення. Не дарма російське прислів'я свідчить: "Недоучений гірше неписьменного".

Надлишок інформації може бути також шкідливий при ухваленні рішення, як і її недолік, оскільки для аналізу і обробки додаткової інформації потрібний час. А час для людини – один з найдефіцитніших і дорожчих "ресурсів" його життя, не говорячи вже про екстремальні ситуації, коли від швидкості ухвалення рішення може залежати дуже багато що.

4 Інформація **актуальна** (своєчасна), якщо вона важлива, істотна для теперішнього часу.

Тільки вчасно отримана інформація може принести необхідну користь.

Неактуальною інформація може бути по трьом причинам, вона може бути:

- застарілою (*торішня газета*);
- передчасною (*прогноз погоди на літо, даний в січні*);
- незначущою, непотрібною (*наприклад, повідомлення про те, що в Італії понижені ціни на проїзд в транспорті на 5 %*).

5 Інформація може бути **корисною** або **даремною** (**цінність** інформації).

Але, оскільки чіткої межі між цими поняттями немає, то слід говорити про ступінь корисності стосовно потреб конкретних людей. Корисність інформації оцінюється по тих завданнях, які ми можемо вирішити з її допомогою. Оцінка корисності інформації завжди суб'єктивна. Те, що корисно для однієї людини, може бути здійснене марно для іншого.

6 Інформація **зрозуміла**, якщо вона виражена на мові, доступній для одержувача.

**Соціальна** інформація володіє ще і **додатковими властивостями**, а саме:

7 Має **семантичний** (сисловою) **характер**. Як правило, зміст, сенс для людини важливіший, ніж форма представлення інформації.

8 Має **мовну природу**. Один і той же зміст може бути виражений на різних природних (розмовних) або спеціальних мовах.

9 З часом кількість інформації **росте**, інформація **накопичується**, відбувається її **систематизація, оцінка і узагальнення**. Цю властивість назвали **зростанням і акумулюванням** інформації (кумуляція – від латинського *cumulatio* – збільшення, скупчення).

10 Властивість *старіння* інформації полягає в зменшенні її цінності з часом.

Старить інформацію не сам час, а поява нової інформації, яка уточнює, доповнює або відкидає повністю або частково ранішу інформацію. Науково-технічна інформація старіє швидше, естетична (витвори мистецтва) – повільніше.

11 *Логічність, компактність, зручна форма представлення* інформації полегшує розуміння і засвоєння інформації.

Приклад. Грамотний, доказовий виступ, коли оратор логічно переходить від одного питання (факту, припущення) до іншого, сприймається краще, ніж сумбурна мова. Використання схем нерідко краще прояснює принцип роботи технічного пристрою, чим багатосторінкові описи.

12 При сприйнятті і розумінні текстів людиною важливою властивістю інформації виявляється її *визначеність* (однозначність).

## 2.4. Вимірювання інформації

Перш, ніж вимірювати значення якої-небудь фізичної величини, треба ввести одиницю вимірювання. У інформації теж є така одиниця – **біт**, але сенс її різний при вимірюванні інформації в рамках різних підходів до *визначення* поняття "інформація".

**ПЕРШИЙ ПІДХІД. Невимірюваність інформації в побуті (інформація як новизна для одержувача).**

Ви отримали деяке повідомлення, наприклад, прочитали статтю в журналі. У цьому повідомленні міститься якась кількість нових для Вас відомостей. Як оцінити, скільки інформації Ви отримали? Іншими словами, *як виміряти інформацію?* Чи можна сказати, що чим більше стаття, тим більше інформації вона містить? Різні люди, що отримали одне і те ж повідомлення, по-різному оцінюють його новизну і інформаційну місткість. Це відбувається тому, що знання людей про події, явища, про які йдеться в повідомленні, *до отримання* повідомлення були різними. Тому ті, хто знав про це мало, визнають, що отримали багато інформації, ті ж, хто знав більше, можуть сказати, що інформації не отримали зовсім. Кількість *інформації в повідомленні, таким чином, залежить від того, наскільки новим є це повідомлення для одержувача.* У такому разі, кількість інформації в одному і тому ж повідомленні повинна визначатися окремо для кожного одержувача, тобто мати суб'єктивний характер. Але суб'єктивні речі не піддаються порівнянню і аналізу, для їх вимірювання важко вибрати одну загальну для всіх одиницю вимірювання. Таким чином, з погляду інформації як новизни, ми не можемо однозначно і об'єктивно оцінити кількість інформації, що міститься навіть в простому повідомленні. Що ж тоді говорити про вимірювання кількості інформації, що міститься в науковому відкритті, новому музичному стилі, новій теорії суспільного розвитку. Тому, коли інформація розглядається як новизна повідомлення для одержувача, питання про вимірювання кількості інформації не ставиться.

## ***ДРУГИЙ ПІДХІД – технічний або об'ємний.***

У техніці, де інформацією вважається будь-яка послідовність знаків, що зберігається, оброблювана або передавана, сигналів, часто використовують простий спосіб визначення кількості інформації, який може бути названий **об'ємним**. Він заснований на *підрахунку числа символів в повідомленні*, тобто пов'язаний тільки з довжиною повідомлення і не враховує його змісту.

Довжина повідомлення залежить від числа знаків, що вживаються для запису повідомлення. Наприклад, слово "мир" в російському алфавіті записується трьома знаками, в англійському – п'ятьма (peace), а в ДКОИ-8 (двійковий код обміну інформацією завдовжки 8) – двадцятьма чотирма бітами (111011011110100111110010).

У обчислювальній техніці застосовуються дві стандартні одиниці вимірювання: **біт** (англ. **binary digit** – двійкова цифра) і **байт (byte)** і похідні від них одиниці – *кілобайт (Кб)*, *мегабайт (Мб)*, *гігабайт (Гб)*, *терабайт (Тб)*.

1 біт – мінімально можливий сигнал, що зберігається і передаваний. Умовно два його можливих стани позначаються 0 і 1. Насправді ці стани можуть мати різну фізичну природу: для оперативної пам'яті це наявність або відсутність напруги в електронній схемі; для компакт-дисків це виступ або западина на поверхні і т.д.

Історично склалося так, що 1 байт рівний 8 бітам. Саме вісьма бітами кодувався один символ в програмах для перших ЕОМ.

Зазвичай приставка "кіло" означає тисячу, а приставка "мега" – мільйон, але в обчислювальній техніці все "прив'язується" до прийнятої двійкової системи кодування.

Через це один кілобайт рівний не тисячі байт, а  $2^{10} = 1024$  байт.

Аналогічно,  $1 \text{ Мб} = 2^{10} \text{ Кб} = 1024 \text{ Кб} = 2^{20} \text{ байт} = 1\,048\,576 \text{ байт}$ .

$1 \text{ Гб} = 2^{10} \text{ Мб} = 2^{20} \text{ Кб} = 2^{30} \text{ байт} = 1\,073\,741\,824 \text{ байт}$ .

## ***ТРЕТІЙ ПІДХІД – імовірностний.***

Вимірювання інформації в теорії інформації, коли інформація визначається як знята невизначеність.

Отримання інформації (її збільшення) одночасно означає збільшення знання, що, у свою чергу, означає зменшення незнання або інформаційної невизначеності. Говорять, що повідомлення, яке зменшує невизначеність, що існувала до його отримання, рівно в 2 рази, несе 1 біт інформації. По суті, 1 біт інформації відповідає вибору одного з двох рівноімовірних повідомлень.

Дуже приблизно можна вважати, що кількість інформації в повідомленні про якусь подію співпадає з кількістю питань, які необхідно задати і відповіддю на які можуть бути лише "так" чи "ні", щоб отримати ту ж інформацію. Причому подія, про яку йдеться, повинна мати *рівноімовірні* результати. Саме тому, якщо число рівноімовірних результатів події, про яку йдеться в повідомленні, кратно ступені числа 2 ( $4 = 2^2$ ,  $8 = 2^3$ ,  $32 = 2^5$ ), то повідомлення несе ціла кількість біт інформації. Але в реальній практиці можуть зустрічатися самі різні ситуації. Наприклад, повідомлення про те, що на світлофорі червоний сигнал, несе в собі інформації більше, ніж битий.

З точки зору на інформацію як на зняту невизначеність *кількість інформації залежить від вірогідності отримання* даного повідомлення. Причому, чим більше вірогідність події, тим менше кількість інформації в повідомленні про таку подію. Іншими словами, кількість інформації в повідомленні про якусь подію залежить від вірогідності звершення даної події.

## 2.5. Теорія інформації

Теорія інформації — це наука, що вивчає комерційні закономірності, зв'язані з одержанням, передачею, обробкою і збереженням інформації. Її основні положення були сформульовані Сенноном у 40-х роках н.е. Теорія інформації є необхідним складовою керування і відіграє важливу роль у науковому менеджменті.

Інформація — наявність деяких зведень про стан об'єкта і навколишнє зовнішнє середовище. Будь-яка інформація, щоб бути переданою, повинна бути закодована. Сигналами можуть бути світлові або світлові імпульси, сигнали і т.д. Щоб вирішити задачі інформації, потрібно навчитися вимірювати кількість обсягу переданої інформації, пропускну здатність каналу. Інформація протилежна ентропії — міри невизначеного стану фізичної системи.

Ступінь невизначеності фізичної системи визначається не тільки числом його можливих станів, але і числом його імовірностей.

$x_i$	$x_1$	$x_2$	...	$x_n$
$P_i$	$P_1$	$P_2$	...	$P_n$

Розглянемо деяку величину  $X$ , що характеризує стан системи. Тоді імовірність того, що система знаходиться у стані  $x_i$

$$P_i = P(X \sim x_i) \quad (2.1)$$

Як міру апіорного не визначення в теорії застосовується спеціальна одиниця виміру – ентропія. Поняття ентропії є основним.

$$H(X) = - \sum_{i=1}^n P_i \log P_i \quad (2.2)$$

Ентропія володіє властивостями:

1. Обертається в нуль, коли одне зі складових вірогідно, а інший стан неможливий;
2. При заданому числі станів вона звертається в максимум, коли ці стани рівноімовірні;
3. Має властивість адитивності, тобто коли кілька незалежних систем об'єднані в одну, то ентропія складається.

## 2.6. Надійність інформації

Надійність передачі інформації визначається надійністю з'єднання елементів в інформаційній системі.



При послідовному з'єднанні елементів вихід з ладу хоча б одного елемента приводить до відмовлення всієї системи. Імовірність безвідмовної роботи буде дорівнювати добуткові імовірностей безвідмовної роботи кожного елемента системи.

$$P_c = P_1 * P_2 * \dots * P_n; \quad (2.3)$$

$$P_c = \prod_{i=1}^n P_i; \quad (2.4)$$

$$P_c(t) = \prod_{i=1}^n P_i(t); \quad (2.5)$$

Якщо в рівнянні імовірність безвідмовної роботи однакова, тоді

$$P_c(t) = \prod P_i(t), \quad (2.6)$$

де  $(a = n)$ .

При рівнобіжному з'єднанні елементів система в цілому відмовляє при виході з ладу всіх елементів. Імовірність відмовлення системи дорівнює імовірності відмовлення всіх елементів.

$$q_c(t) = q_1(t) * q_2(t) * \dots * q_n(t); \quad (2.7)$$

$$q_c(t) = \prod_{i=1}^n q_i(t); \quad (2.8)$$

Імовірність безвідмовної роботи:

$$P_c(t) = \prod_{i=1}^n q_i(t). \quad (2.9)$$

## 2.7. Тривалість передачі інформації

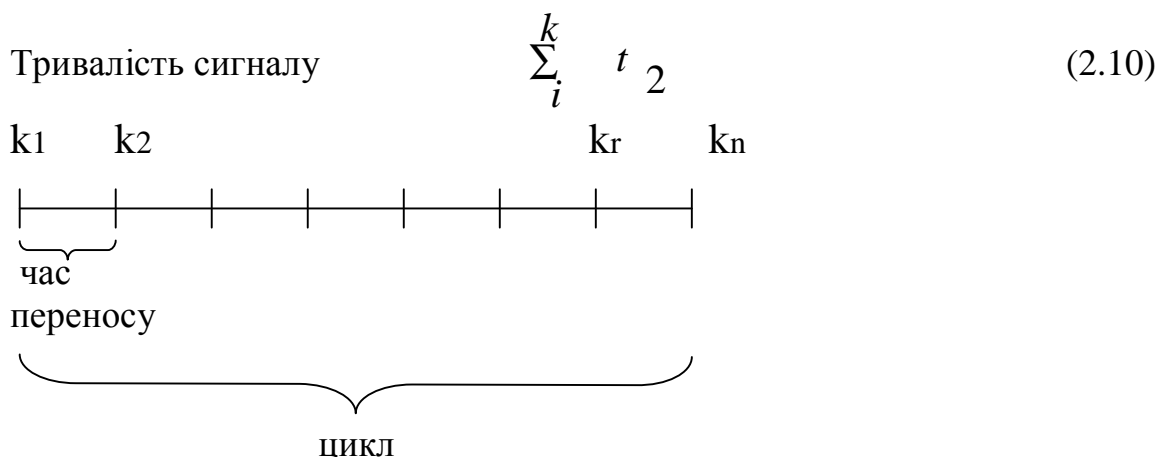


Рис. 2.1. Тривалість передачі інформації

Час процесу передачі інформації:

$$\sum_i^k t_2 + \sum_i^{k-1} = t_g, \quad (2.11)$$

де  $\sum_i^{k-1}$  — час переносу інформації.

## 2.8. Індивідуальне завдання №1. Розрахунок обсягу інформації

1. Система може знаходитися тільки у чотирьох станах з імовірністю  $A$ ,  $B$ ,  $C$  або  $D$ . Знайти ентропію системи, як що відомі чисельні значення  $A$ ,  $B$ ,  $C$ .

Студент обирає свій варіант за номером у списку групи.

№ вар	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	0,28	0,03	0,11	0,28	0,27	0,22	0,29	0,38	0,35	0,24	0,09	0,3	0,32	0,19	0,06
B	0,02	0,2	0,28	0,15	0,04	0,35	0,12	0,29	0,21	0,11	0,31	0,12	0,26	0,16	0,01
C	0,38	0,28	0,16	0,28	0,31	0,06	0,02	0,33	0,15	0,27	0,25	0,17	0,34	0,33	0,34
№ вар	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	0,03	0,17	0,02	0,21	0,21	0,24	0,29	0,15	0,16	0,01	0,17	0,08	0,15	0,38	0,33
B	0,33	0,22	0,24	0,27	0,29	0,32	0,14	0,22	0,07	0,16	0,07	0,32	0,28	0,35	0,06
C	0,15	0,26	0,06	0,01	0,2	0,09	0,26	0,13	0,02	0,04	0,29	0,32	0,36	0,27	0,38

2. За даним з п.1 завдання, знайти ймовірність безвідмовної роботи такої системи.

### Контрольні запитання

1. Що таке „носії інформації”?
2. Чим інформація відрізняється від знань?
3. Чому ентропія є основою для виміру інформації?
4. Назвіть властивості інформації.
5. Як визначити надійність інформації?

*У розділі подано історію розвитку інформатики, поняття про основні підходи до визначення поняття "інформація", про носії інформації, види і властивості інформації. Подано формули для вимірювання інформації, розрахунку надійності інформації.*

# 3. ТЕХНІЧНІ ЗАСОБИ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ

*Вивчивши матеріали цього розділу, студент отримає інформацію про електронні елементи, які складають комп'ютер та принципи їх взаємодії.*

## 3.1. Принципи архітектури і історія розвитку комп'ютерів

### 3.1.1. Архітектура комп'ютера

**Архітектура ЕОМ** – загальний опис структури і функцій ЕОМ, її ресурсів. У цей опис входить:

- загальна конфігурація основних пристроїв;
- основні можливості і характеристики пристроїв;
- способи взаємозв'язку основних пристроїв комп'ютера.

**Ресурси ЕОМ** – засоби обчислювальної системи, які можуть бути виділені процесу обробки даних на певний інтервал часу. До ресурсів ЕОМ традиційно відносять об'єм доступної пам'яті, процесорний час і ін.

До **центрального** (системним) **пристроїв** комп'ютера відносять, перш за все, центральний процесор, оперативну пам'ять, системну магістраль.

**Периферійними пристроями** комп'ютера є: дисплей, клавіатура, маніпулятори – миша, джойстик, світлове перо і т.п., вінчестер, дисководи для гнучких і компакт-дисків, принтер, плоттер, сканер, модем і ін.

**Порт** – пристрій, через який периферійні пристрої підключаються до системної магістралі.

При розробці принципів архітектури комп'ютерів широко використовується ідея про розділення окремих операцій процесу рішення задачі (процесу обчислень) між окремими "спеціалізованими" пристроями.

Коли Чарльз Беббідж розробляв аналітичну рахункову машину в 1830-х рр. він припустив, що для успішної роботи необхідні як мінімум наступні пристрої:

- пристрій для обробки даних, в якому безпосередньо здійснюються обчислення ("млин");
- пристрій для зберігання даних ("склад");
- пристрій для управління процесом обчислень ("контора").

Розробці Беббіджа не призначено було утілитися в моделі, що діє, але ідеї про розділення окремих операцій процесу обчислень між окремими "спеціалізованими" пристроями отримали подальший розвиток в принципах архітектури комп'ютерів, традиційно званих принципами фон Неймана (1940-і рр.). Ці принципи такі:

- **принцип програмного управління.** Всі пристрої працюють під управлінням програм. Програми складаються з окремих кроків – команд. Послідовність команд і є програмою;

- **принцип умовного переходу.** Існує можливість міняти послідовність обчислень залежно від отриманих проміжних результатів;
- **принцип програми, що зберігається.** Програми і дані до них зберігаються в одній тій же пам'яті.

Команди представляються в числовій формі і зберігаються в тому ж ОЗУ, що і дані для обчислень.

Таким чином, команди можна посилати в арифметичній пристрій і перетворювати як звичайні числа. Це дозволяє створювати програми, здатні в процесі обчислень змінювати самі себе;

- **принцип ієрархічності пристроїв, що запам'ятовують,** – пам'ять ділиться на оперативну (швидка, невеликого розміру) і довготривалу (велику, а тому повільну). Найбільш часто використовувані дані зберігаються в швидкому ЗУ порівняно малій місткості, а більш рідко використовувані – в повільному, але набагато більшій місткості;
- **принцип двійкового кодування** – вся інформація в комп'ютері зберігається і обробляється в двійковому коді.

Починаючи з перших ЕОМ (1940-і рр.), реалізовувалася схема взаємодії пристроїв комп'ютера, заснована на цих принципах, представлена на рис. 3.1.

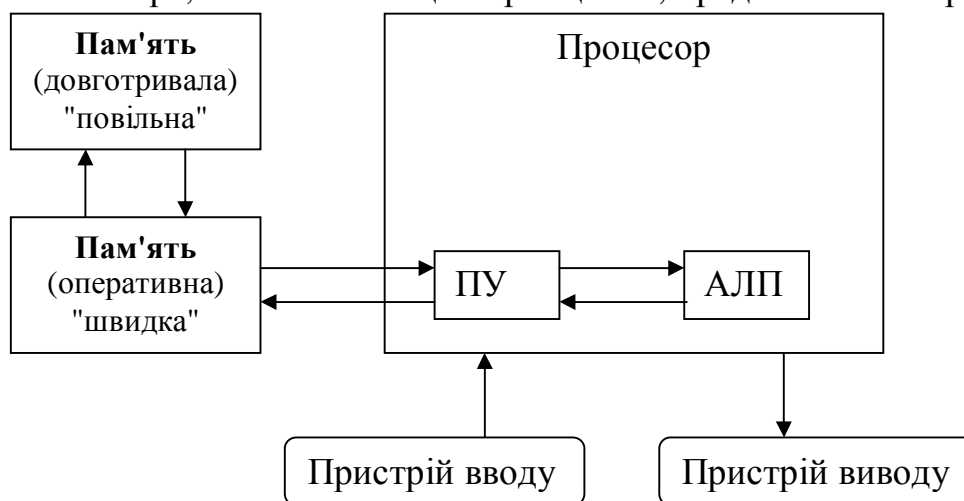


Рис. 3.1. Схема взаємодій пристроїв комп'ютера згідно архітектурі фон Неймана. ПУ – пристрій управління; АЛП – арифметико-логічний пристрій

Що стосується персональних комп'ютерів (кінець 1970-х – початок 1980-х рр.), то їх основу складає системна ("материнська") плата, що знаходиться в системному блоці, на якій розміщені **системні (центральні)** пристрої комп'ютера – **процесор і пам'ять** (оперативна і постійна), сполучені між собою **системною шиною** (інформаційною магістраллю), до якої під'єднуються **контролери всіх периферійних пристроїв**, що підключаються до комп'ютера (див. рис. 3.2). При цьому периферійними вважаються і клавіатура, і монітор, і вінчестер, і дисководи, і модем, і маніпулятори, і сканер, і відеокамера, і т.д. Додаткові пристрої, що дозволяють користувачеві комп'ютера слухати музику, дивитися відеоролики, працювати в мережі і т.д., підключаються через спеціальні **плати розширення**. Неможлива робота комп'ютера і без таких

допоміжних (з погляду процесу обробки інформації) пристроїв, як блок живлення, система охолодження і ін.

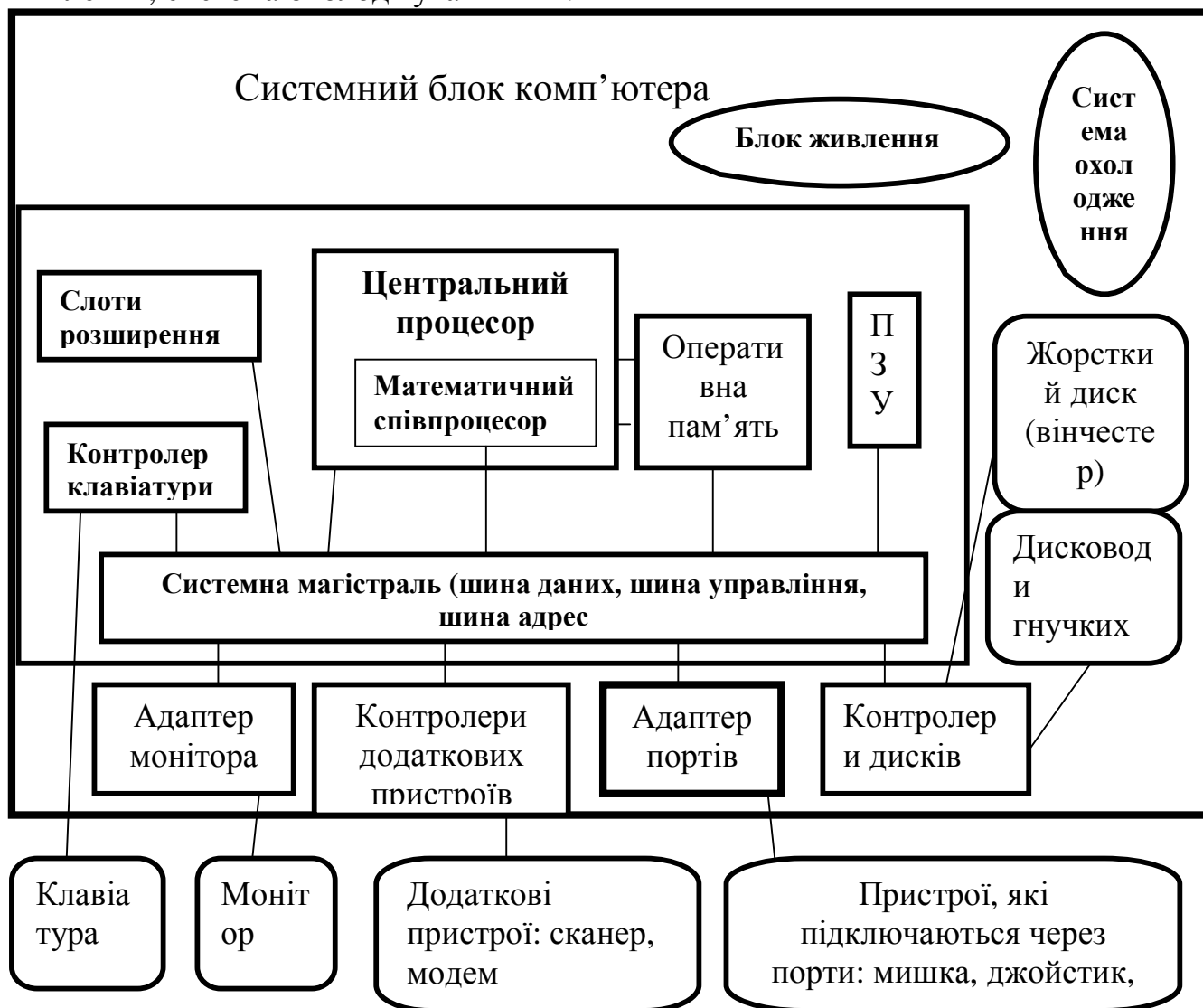


Рис. 3.2 - Схема архітектури персонального комп'ютера

### 3.1.2. Основні тенденції розвитку апаратного забезпечення комп'ютера.

Розвиток ЕОМ з моменту їх появи відбувається швидкими темпами. Модернізуються існуючі пристрої і розробляються нові, з'являються більш довершені конструктивні рішення для забезпечення взаємозв'язку окремих пристроїв між собою – тобто архітектура ЕОМ постійно удосконалюється. На зміну великим ЕОМ прийшли МІНІ-ЕОМ, а потім і персональні комп'ютери (ПК). Зберігаючи загальні принципи архітектури, кожна нова модель комп'ютерів володіє певними відмітними ознаками.

**Інтеграція пристроїв.** Наприклад, якщо в перших моделях математичний співпроцесор, кешпам'ять, таймер і ряд інших пристроїв виготовлялися і розміщувалися на материнській платі як окремі пристрої, то в даний час вони все частіше об'єднуються в одному кристалі з центральним процесором.

**Розширення спектру периферійних пристроїв.** В даний час користувачеві пропонуються самі різні моделі принтерів, дисплеїв, клавіатур, декілька десятків видів маніпуляторів, сенсорні системи і т.д.

**Уніфікація портів** – перехід від спеціалізованих портів для різних пристроїв (наприклад, LPT – Line PrinTer – для підключення принтера і COM – communicate – для модему і т.п.) до універсальних портів – USB – universal serial bus (універсальна послідовна шина). До одного USBпорту можна підключити до 127 пристроїв різного призначення.

**Уніфікація двійкового кодування символів** – перехід від множини однобайтних таблиць кодувань (ASCII, ЯКІ-8, CP1251 і т.п.) до єдиної двобайтової таблиці Unicode, що містить коди  $2^{16} = 65536$  різних символів.

### 3.1.3 Магістрально-модульний принцип архітектури ПК

Архітектура сучасних персональних комп'ютерів заснована на **магістрально-модульному принципі** організації обміну інформації. Відповідно до цього принципу центральні пристрої комп'ютера взаємодіють між собою (обмінюються інформацією) і з периферійними пристроями через системну (інформаційну) магістраль.

**Системна магістраль** призначена для передачі даних, адрес, команд управління, а тому включає шину даних, адресну шину і шину управління.

Центральні пристрої приєднані до шини безпосередньо, а периферійні – через **пристрої сполучення** (контроллери або адаптери).

### 3.1.4 Покоління ЕОМ і їх основні характеристики.

Існує своєрідна періодизація розвитку ЕОМ, зв'язана, в основному, з типом використовуваної елементної бази, яка визначає у свою чергу швидкодію комп'ютера, місткість ЗП. Часові межі поколінь сильно розмиті: в один і той же час випускалися і використовувалися машини різних типів. Для окремих же машин питання про їх приналежність до того або іншого покоління вирішується досить просто.

З переходом до серійного виробництва **лампових** ЕОМ з програмою, що зберігається, починається період машин **першого покоління**.

Як зовнішні носії інформації при введенні і виведенні даних використовувалися перфострічки і перфокарти. Типова (середнє) швидкодія машин першого покоління вимірювалася декількома тисячами арифметичних операцій в секунду.

У 1948 р. був винайдений транзистор і починаючи з середини 1950-х рр. на зміну ламповим машинам прийшли **транзисторні** машини **другого покоління**, в яких основними елементами були напівпровідникові тріоди-транзистори (габарити транзистора, що замінив електронну лампу, приблизно в 40 разів менше). Транзисторні машини володіли значно вищою надійністю, ніж лампові ЕОМ (середній термін служби транзисторів на два-три порядки перевершує термін служби електронних ламп), меншим споживанням енергії,

вищою швидкодією, яка досягалася не тільки за рахунок підвищення швидкості перемикання рахункових елементів, що запам'ятовують, але і за рахунок змін в структурі машин. Найбільш могутні машини другого покоління (МІНСЬК, МИР, БЭСМ-6) досягали швидкодії до ста тисяч операцій в з.

У 1964 р. були винайдені інтегральні схеми (ІС – електронна схема, витравлена на поверхні кремнієвого кристала, містить декілька сотень транзисторів). У інтегральних схемах елементи створюються за спеціальною технологією в самій речовині матеріалу. Основою для таких схем служать напівпровідникові матеріали, найчастіше кремній.) **Інтегральні схеми** стали елементною базою для машин **третього покоління**. Початок періоду машин третього покоління пов'язаний з розробкою серії ІВМ-360 (США) і ЄС ЕОМ (країни соціалістичної співдружності). Перехід на ІС впливав на збільшення надійності роботи ЕОМ, зменшення габаритних розмірів, зменшення споживання енергії. ЕОМ третього покоління оперують з довільною буквено-цифровою інформацією (з'явилася можливість вводити інформацію з клавіатури, а не з перфострічок і перфокарт, як це було раніше). Вони побудовані за принципом незалежної паралельної роботи різних пристроїв: *процесора, засобів зовнішньої пам'яті*, завдяки чому ЕОМ одночасно могла виконувати серію операцій: вводити інформацію з *магнітної стрічки*, вирішувати задачі, виводити інформацію на *магнітний диск* або *друкуючий пристрій*. Починаючи з третього покоління комп'ютери працюють під управлінням *операційних систем*. Учені постійно працювали над зменшенням розмірів апаратури, що привело до появи так званих міні- і МІКРОЕОМ.

У 1970 р. були розроблені **великі інтегральні схеми** (ВІС). На одному кристалі вдалося розмістити всі основні електронні частини процесора – з'явився **мікропроцесор**. Постійне вдосконалення технології виробництва ВІС спричинило за собою швидку зміну поколінь мікропроцесорів, що приводить до стрімкого вдосконалення електронної обчислювальної техніки – появи **четвертого покоління** ЕОМ. Продуктивність машин четвертого покоління досягає декількох мільйонів операцій в секунду. Об'єм оперативної пам'яті досягає декілька мегабайт.

З мікропроцесорною революцією безпосередньо пов'язана одна з найважливіших подій в історії ЕОМ – створення і широке застосування персональних комп'ютерів. ПК володіють такими привабливими властивостями для користувачів, як мале споживання енергії, відносно мала вартість, невеликі габаритні розміри, підвищена надійність в експлуатації, великі можливості обробки інформації різного типу.

Звичайно, розвиток ОТ йде не тільки по лінії зміни елементній-технологічної бази. Продуктивність традиційних обчислювальних систем підвищується двома шляхами: **розвитком елементної бази і архітектури ЕОМ**.

**Однопроцесорна** архітектура має межу продуктивності, визначувану швидкістю розповсюдження електричного сигналу по фізичних лініях зв'язку між пристроями комп'ютера.

Інший підхід підвищення продуктивності обчислювальних систем заснований на використанні **принципу паралелізму**, тобто обробка інформації розбивається на декілька одночасно виконуваних послідовних процесів, кожний з яких може обмінюватися інформацією з іншими процесами. У 1985 р. було розроблено новий обчислювальний пристрій, що отримав назву **трансп'ютер**.

**Трансп'ютер** – це обчислювальний пристрій, що виконаний у вигляді однієї **надвеликої інтегральної схеми**, що містить процесор, запам'ятовує пристрій і **комунікаційні зв'язки** для з'єднання з іншими комп'ютерами. Трансп'ютер використовується як елементарний блок для побудови паралельних високопродуктивних систем. Вони можуть бути сполучені один з одним різними способами, утворюючи різноманітну архітектуру паралельної обробки.

В даний час створюються мікропроцесори сьомого покоління (Intel IV, Athlon і ін.). Їх швидкодія оцінюється в 1,5-3 ГГц або 1,5-3 мільярда операцій в секунду. Оперативна пам'ять має місткість 256,512,1024 Мб . Є засоби мовного введення команд. Багато програм включають модулі, що традиційно відносяться по своїх функціях до систем штучного інтелекту. На спеціальних серверах використовується відразу декілька процесорів, що розпаралелюють процес виконання програм. ПК, оснащені такого роду пристроями, можна віднести до комп'ютерів п'ятого покоління.

## **3.2. Представлення інформації в комп'ютері**

Представлення даних в комп'ютері зводиться, по суті, до різних способів їх кодування в двійковому алфавіті (рис. 3.3). А саме існування різних способів обумовлюється тим, що обробка даних різного типу включає різні операції, і доцільно вибирати такий спосіб кодування, при якому операції над даними виконуються найшвидше і "універсально".

### **3.2.1. Представлення текстової інформації**

Під текстом розуміється послідовність знаків, що входять в деяку множину (алфавіт) і володіють в цій множині певним сенсом, тобто кожен знак розглядається в такому контексті як символ. Тому часто дані такого вигляду називають ще символічними.

У вужчому сенсі текстом називатимемо деяку послідовність закодованих в двійковому коді знаків, яку можна зберігати, передавати, перетворювати за допомогою комп'ютера.

Якщо спочатку комп'ютери були призначені для проведення всіляких розрахунків ("обчислювальні машини"), тобто виконання операцій над числами, то в даний час значна частка комп'ютерних ресурсів витрачається саме на введення, зберігання, передачу і обробку текстів.





Рис. 3.3. Форми представлення інформації

Які ж операції можна проводити над текстом чи, він вже знаходиться в пам'яті комп'ютера? Перш за все, пошук заданої послідовності символів і заміна на іншу послідовність символів, впорядкування символів (відмітимо, що додавання, заміна, видалення, копіювання і тому подібні операції відносяться в рівній мірі до обробки даних всіх типів). І пошук, і впорядкування зводяться до виконання операції порівняння символів. При порівнянні символів найбільш важливими аспектами є унікальність коду для кожного символу і довжина цього коду, а сам вибір принципу кодування в даний час практично не має значення. Тому можна використовувати різні таблиці кодування, лише б при кодуванні і декодуванні символів використовувалася одна і та ж таблиця.

У обчислювальних машинах для представлення інформації використовуються ланцюжки байтів. Тому для перекладу інформації з машинного уявлення в людський необхідні таблиці кодування символів – таблиці відповідності між символами певної мови і кодами символів. Їх ще називають **таблицями кодування**, **кодovими сторінками** або застосовують англійський термін *character set* (який іноді скорочують до *charset*).

Найвідомішою таблицею кодування є код ASCII (Американський стандартний код для обміну інформацією). Спочатку він був розроблений для передачі текстів по телеграфу, причому у той час він був 7-бітовим, тобто для кодування символів англійської мови, службових символів, що управляють, використовувалися тільки 128 7-бітових комбінацій. При цьому перші 32 комбінації (коду) служили для кодування сигналів, що управляли (початок тексту, кінець рядка, переведення каретки, дзвінок, кінець тексту і т.д.). При розробці перших комп'ютерів фірми IBM цей код був використаний для представлення символів в комп'ютері. Оскільки в початковому коді ASCII було всього 128 символів, для їх кодування вистачило значення байта, у яких 8-ою біт рівний 0. Значення байта з 8-м бітом, рівним 1, стали використовувати для представлення символів псевдографіки, математичних знаків і деяких символів з мов, відмінних від англійського (грецького, німецьких умляутив, французьких діакритичних знаків і т.п.).

Коли стали пристосовувати комп'ютери для інших країн і мов, місця для нових символів вже не стало хапати. Для того, щоб повноцінно підтримувати крім англійського і інші мови, фірма IBM ввела у вживання декілька кодових таблиць, орієнтованих на конкретні країни. Так для скандинавських країн була запропонована таблиця 865 (Nordic), для арабських країн – таблиця 864 (Arabic), для Ізраїлю – таблиця 862 (Israel) і так далі. У цих таблицях частина кодів з другої половини кодової таблиці використовувалася для представлення символів національних алфавітів (за рахунок виключення деяких символів псевдографіки).

З російською мовою ситуація розвивалася особливим чином. Очевидно, що заміну символів в другій половині кодової таблиці можна провести різними способами. Ось і з'явилися для російської мови декілька різних таблиць кодування символів кирилиці: KOI8-R, IBM-866, CP-1251, ISO-8551-5. Всі вони однаково зображають символи першої половини таблиці (від 0 до 127) і розрізняються представленням символів російського алфавіту і псевдографіки.

Що стосується *довжини коду*, то вона по можливості повинна бути мінімальною і в той же час достатньою для кодування всіх символів, які можуть потрібно при введенні текстів будь-якої складності.

ЕОМ перших поколінь через обмеженість оперативної пам'яті і невисокої за сучасними мірками швидкодії процесорів використовували мінімальний набір символів. Зростання потужності комп'ютерних пристроїв привело до того, що в даний час при роботі з одним ПК можна використовувати декілька різних таблиць тієї, що перекодувала. По суті, багато сучасних програмних засобів, будь то текстові процесори або програми для роботи в Internet, надають користувачеві можливість вибору, яку з таблиць тієї, що перекодувала використовувати в своїй роботі, автоматично проводячи перетворення існуючого текстового файлу з одного способу кодування в іншій по команді користувача. Відповідні програмні модулі називають конверторами.

Зараз, коли об'єм пам'яті комп'ютерів надзвичайно виріс, вже немає потреби сильно економити при кодуванні тексту. Можна дозволити собі "розкіш" витратити для зберігання тексту удвічі більше пам'яті (виділяючи для

кожного символу не 1, а 2 байти). При цьому з'являється можливість розмістити в кодовій таблиці – кожний на своєму місці – не тільки букви європейських алфавітів (латинського, кириличного, грецького), але і букви арабського, грузинського і багатьох інших мов і навіть велику частину японських і китайських ієрогліфів. Адже два байти можуть зберігати вже число від 0 65535.

Двобайтове міжнародне кодування UNICODE, розроблене кілька років тому, тепер починає упроваджуватися на практиці. У стандарті UNICODE окрім певного двійкового коду (ці коди прийнято позначати буквою U, після якої слідує знак + і власне код в шістнадцятковому уявленні) кожному символу привласнено певне ім'я. Повна специфікація стандарту Unicode є досить товстою книгою і включає всі існуючі, вимерлі і штучно створені алфавіти миру, а також безліч математичних, музичних, хімічних і інших символів.

Відзначимо, що і ASCII і UNICODE і інші стандарти кодування символів не визначають зображення символів, а тільки склад набору символів і спосіб його уявлення в комп'ютері. Крім того, дуже важливий порядок перерахування символів в наборі, оскільки він впливає найістотнішим чином на алгоритми сортування.

### 3.2.2. Представлення графічної інформації

Людина сприймає інформацію за допомогою всіх органів чуття і самими різними способами, але все таки основними формами інформації, з якими нам найзвичніше "працювати", є вербальний і образний, причому останньому часто віддається перевага. Сьогодні важко уявити собі екран монітора без графічного оформлення: піктограми стали невід'ємними компонентами призначеного для користувача інтерфейсу багатьох програмних засобів, за допомогою програм-аніматорів створити мультиплікаційний фільм може зараз не тільки колектив професійних художників, але і будь-яка людина, якій цікаво цим займатися.

Правда, так було не завжди. Найперші комп'ютери могли працювати тільки з числами – проводити обчислення. Потім із збільшенням швидкодії процесорів, появою дисплеїв і зовнішньої пам'яті на магнітних носіях з'явилася можливість обробляти текстову інформацію. Але вже тоді хотілося створювати таблиці і графіки, малювати на комп'ютері. Але на алфавітно-цифрових дисплеях графіка була можлива тільки як "псевдографіка" (взагалі, для комп'ютера "псевдо" – широко поширене явище: графіка – точкова, простір – віртуальний, інтелект – штучний і т.д.).

Графічні дисплеї з'явилися на початку восьмидесятих років. До цього часу досягнута швидкодія процесорів, об'єми пам'яті, обширні бібліотеки алгоритмів вже дозволяли створювати, виводити на екран, перетворювати навіть дуже складні графічне образи в реальному часі, тобто без помітних для людини "затримок" на їх обробку комп'ютерними пристроями.

Комп'ютерна графіка зараз розповсюджується все ширше. Навіть програми, призначені для створення і роботи з текстами, – текстові процесори –

мають, як правило, вбудовані можливості по створенню малюнків, креслень, схем і вбудовуванню їх в текст.

Мабуть, саме для уявлення в двійковому коді інформації графічного вигляду розроблено найбільше число різних способів. Непрямим свідомством цього можна вважати наявність великого числа форматів графічних файлів: *\*.bmp*, *\*.jpg*, *\*.tif*, *\*.psx*, *\*.aca*, *\*.cdr* і т.д. Частково це пов'язано з тим, що для зберігання малюнків в двійковому коді потрібно багато місця в пам'яті комп'ютера, і програмісти завжди намагалися винайти "економний" спосіб кодування графічних зображень.

Важливим етапом кодування графічного зображення є розбиття його на дискретні елементи (дискретизація).

Основними способами представлення графіки для її зберігання і обробки за допомогою комп'ютера є **растрові і векторні зображення**.

**Векторне зображення** є графічний об'єкт, що складається з елементарних геометричних фігур (найчастіше відрізків, дуг, прямокутників, еліпсів). Положення цих примітивів визначається координатами характерних точок і величиною радіусу. Для кожної лінії вказується її тип (суцільна, пунктирна, штрих-пунктирна), товщина і колір. Інформація про векторне зображення кодується як звичайна числа і обробляється спеціальними програмами.

**Графічний примітив** – геометрична фігура, що є елементом графічного зображення, що створюється за допомогою графічного редактора.

Графічним примітивом можуть бути прямокутники, еліпси, прямі лінії, стрілки різної конфігурації і т.п.

**Матричний принцип** кодування графічних зображень полягає в тому, що зображення розбивається на задану кількість рядків і стовпців. Потім колір кожного елемента (клітки, крапки) отриманої сітки кодується за вибраним правилом.

Правила можуть бути найрізноманітнішими, розглянемо найбільш просте з них.

Якщо малюнок чорно-білий, то досить ставити у відповідність клітинці, у якої "закрашена" більше половини площі, одиниці, інакше – нуль. Якщо малюнок кольоровий, то для кожної крапки потрібно зберігати код її кольору.

Крапки-клітки, на які розбивається зображення, називаються **пікселями**.

Pixel (picture element – елемент малюнка) – мінімальна одиниця зображення, колір і яскравість якої можна задати незалежно від решти зображення.

Відповідно до матричного принципу будуються зображення, що виводяться на принтер, відображаються на екрані дисплея, отримувані за допомогою сканера.

Якість зображення буде тим вище, чим "щільніше" розташовані пікселі, тобто ніж більше роздільна здатність пристрою, і чим точніше закодований колір кожного з них. Дозвіл пристроїв зазвичай вимірюють в "крапках на дюйм" (dpi).

Про графічне зображення, коли кожен піксель кодується  $n$  бітами, говорять як про графіку завглибшки в  $n$  розрядів.

**Глибина кольору** – довжина коду, використовуваного при кодуванні кольору одного пікселя.

При глибині коду, рівній 1, можна передати  $2^1 = 2$  кольори (наприклад, чорний і білий).

При глибині коду, рівній 4, можна передати  $2^4 = 16$  кольорів.

При глибині коду, рівній 8 (1 байт), можна передати  $2^8 = 256$  кольорів.

При глибині коду, рівній 16 (2 байти), можна передати  $2^{16} = 65536$  кольорів (режим High Color).

При глибині коду, рівній 24 (3 байти), можна передати  $2^{24} = 16777216$  відтінків кольору (режим True Color).

Яким чином кодується колір пікселя або графічного примітиву? Якщо Ви подивитесь на екран дисплея через сильну лупу або збільшувальне скло, то побачите або безліч різноколірних прямокутників, або безліч різноколірних кружечків (залежно від марки і моделі техніки). Кожен екранний піксель складається з трьох таких елементів, один з яких червоного (Red), інший зеленого (Green), третій синього (Blue) кольору (RGB-модель утворення кольорів).

Відомо, що, якщо на зображенні є близько розташовані кольорові деталі, то з великої відстані ми не розрізняємо кольору окремих деталей – відбувається зміщення світлових потоків, що передають кольори цих деталей. Відомо також, що будь-який бажаний колір може бути отриманий в результаті складання (зміщення) червоного, зеленого і синього світлових потоків. Яскравість (інтенсивність) кожного кольору може бути різна.

**Тривимірна комп'ютерна графіка** Створення і редагування тривимірних об'єктів відбувається на базі використання достатньо складного математичного апарату. Він використовується для перетворення тривимірних координат об'єкту, що зображається, в їх проекцію на площину, а також для перерахунку кодів відтінків кольору кожного пікселя при віддзеркаленні на плоскому екрані світлотіней, рельєфу для створення реалістичнішої "об'ємності" зображення.

Спеціальні алгоритми дозволяють масштабувати, нахилити, дзеркально відображати об'єкти в тривимірному просторі, а також створювати ефекти перспективи, скручування, вигину, нахилу і розгойдування тіл. Використовуються різні методи розрахунку освітленості і тіней на викривлених поверхнях тіл. За допомогою ефектів серпанку і дистанційних тіней моделюються різні атмосферні явища: туман, хмарність.

**Анімаційні ефекти** Анімація – штучне представлення руху в кіно, на телебаченні або в комп'ютерній графіці шляхом відображення послідовності малюнків або кадрів з частотою, при якій забезпечується цілісне зорове сприйняття образів.

Як правило, для плавного відтворення анімації необхідна швидкість, або частота кадрів, не меншого 10 кадрів в секунду – інертність зорового сприйняття.

Різниця між анімацією і відео полягає в тому, що відео використовує безперервний рух і розбиває його на безліч дискретних кадрів. Анімація використовує безліч незалежних малюнків або графічних файлів, які

виводяться в певній послідовності для створення ілюзії безперервного руху. Крім того, в традиційній анімації принципово розділена двовимірною, мальованою, і тривимірною, ляльковою, анімація. Комп'ютер стирає ці грані: створене площинне зображення можна перевести в об'єм і накласти на складну поверхню.

Для створення комп'ютерної анімації існує безліч програмних засобів і способів.

Тривимірною графікою і анімація корисні не тільки при виробництві телевізійних анімаційних роликів або спецефектів в кінофільмах, але і у виробничій діяльності при створенні різного роду дво- і тривимірних моделей.

### 3.2.3. Представлення звукової інформації

З курсу фізики відомо, що звук – це коливання повітря. За своєю природою **звук є безперервним сигналом**. Якщо перетворити звук в електричний сигнал (наприклад, за допомогою мікрофону), ми можемо зафіксувати напруження, що плавно змінюється з часом.

Для комп'ютерної обробки аналоговий сигнал потрібно якимсь чином перетворити в послідовність двійкових чисел – **дискретизувати** його (англ. discrete – роздільний, такий, що складається з окремих частин) і **оцифрувати** таким чином: вимірювати амплітуду сигналу через рівні проміжки часу і записувати набуті числові значення в пам'ять комп'ютера.

Пристрій, що виконує процес дискретизації і оцифровки аналогових сигналів, називається **аналого-цифровим перетворювачем** (АЦП).

Результат вимірювань не є точним аналогом безперервного електричного сигналу. Чим частіше відбуваються вимірювання і чим вони точніші. Частота, з якою проводяться вимірювання, називається **частотою дискретизації**. Коли ми говоримо, що частота дискретизації 44,1 кГц, то це означає, що сигнал вимірюється 44100 разів за секунду.

А точність вимірювань амплітуди визначається числом битий, вимірювань, що використовуються для представлення результату. Цей параметр називають **розрядністю** або **рівнем квантування**.

Чим вище частота дискретизації і рівень квантування, тим точніше буде представлений і потім відтворений звук. Проте, пропорційно збільшенню частоти зростають:

а) інтенсивність потоку цифрових даних, а пропускні можливості інтерфейсів не безмежні, особливо якщо записується/воспроизводиться одночасно декілька каналів;

б) обчислювальне навантаження на процесори, а їх можливості також обмежені;

в) об'єм пам'яті, необхідної для зберігання сигналу в цифровій формі.

Тому, залежно від характеру звуку і вимог, що пред'являються до його якості і об'єму займаної пам'яті, вибирають деякі компромісні значення.

Наприклад, при запису музики на компакт-диски використовуються 16-бітове кодування при частоті дискретизації 44,032 кГц, при роботі ж з мовою достатньо 8-бітового кодування при частоті 8 кГц.

Вважається, що діапазон частот, які чує людина, складає від 20 Гц до 20000 кГц.

Згідно теоремі Найквіста-Котельникова, для того, щоб аналоговий (безперервний за часом) сигнал можна було точно відновити по його відліках, частота дискретизації повинна бути як мінімум удвічі більше максимальної звукової частоти. Таким чином, якщо реальний аналоговий сигнал, який ми збираємося перетворити в цифрову форму, містить частотні компоненти від 0 Гц до 20 кГц, то частота дискретизації такого сигналу повинна бути не менше, ніж 40 кГц. Сьогодні найпоширенішими частотами дискретизації є 44,1 кГц і 48 кГц. Останнім часом з'ясовано, що обертони, розташовані понад 20 кГц, вносять чималий внесок до звучання. Тому з'являються перетворювачі, що використовують частоти дискретизації 96 кГц і 192 кГц, а в недалекому майбутньому очікується поява систем і з частотою 384 кГц.

Для запису і зберігання звукового сигналу в цифровій формі потрібний великий об'єм дискового простору. Чим вище за вимогу до якості записуваного звуку, тим більше повинна бути місткість носія.

Істотно понизити об'єм цифрових даних, необхідних для представлення звукового сигналу із заданою якістю, можна за допомогою **компресії**, тобто шляхом зменшення кількості відліків і рівнів квантування. Головне завдання методів компресії – досягнення максимального стиснення звукового сигналу при мінімальних суб'єктивно чутних спотвореннях відновленого сигналу. При цьому використовуються досить складні кодуючі пристрої і програми стиснення – кодеки (від *кодування-декодування*).

### 3.2.4. Представлення числової інформації

Схожість в кодуванні числової і текстової інформації полягає в наступній: щоб можна було *порівнювати* дані цього типу, у різних чисел (як і у різних символів) повинен бути різний код. Основна відмінність числових даних від символічних полягає в тому, що над числами окрім операції порівняння проводяться різноманітні *математичні операції*: складання, множення, обчислення кореня, обчислення логарифма і ін. Правила виконання цих операцій в математиці детально розроблені для чисел, представлених в позиційній системі числення. Багатовікова історія розвитку математики показує, що саме позиційний принцип дозволяє використовувати ці правила як універсальні алгоритми, справедливі для системи числення з будь-якою основою: 2, 3, 8, 10, 16, 60 і ін.

**Система числення** – сукупність прийомів позначення чисел, спосіб запису чисел.

**Основа системи числення** – це:

- число різних цифр, використовуваних для запису чисел;
- кількість одиниць молодшого розряду, відповідних одній одиниці наступного старшого розряду.

Найбільш звична для нас система числення – десяткова. Для запису чисел в ній використовується 10 різних цифр; одиниці будь-якого розряду відповідає 10 одиниць попереднього розряду.

У двійковій системі числення для запису чисел використовується всього дві цифри – 0 і 1, а одиниці будь-якого розряду відповідає дві одиниці попереднього розряду.

У шістнадцятиричній системі числення використовується 16 цифр: перші десять звичні арабські цифри, а для позначення шести цифр, що залишилися, використовуються перші шість прописних букв латинського алфавіту (A, B, C, D, E, F).

У шістдесятиричній системі числення, так добре знайомій нам по нашому годиннику при "численні" секунд і хвилин, в даний час для запису чисел використовується десять цифр, звичних нам по десятковій системі, але вага одиниці кожного розряду складає 60 одиниць розряду попереднього.

Відмітимо, що основа системи числення вказується як нижній індекс, наприклад,  $123,5_{10}$   $1101,101_2$   $120,7_8$   $9A07, C8_{16}$ .

У позиційних системах числення кожне число може бути записане в *цифровій* і *багаточленній* формі.

*Перевести число з r-ричної в десяткову систему числення можна, записавши його в багаточленній формі і обчисливши значення отриманого результату. Здійснити переклад з десяткової системи числення в будь-яку іншу позиційну систему можна також, використовуючи позиційний принцип. Корисним в цьому випадку буде таке поняття, як **базис системи числення**, тобто послідовність так званих ключових чисел, кожне з яких задає значення цифри по її місцю в записі числа.*

### Контрольні запитання

1. Що таке аналого-цифровий перетворювач?
2. Для чого використовується двійкова система числення?
3. Поясніть принцип квантування аналогового сигналу.
4. Для чого потрібні контролери пристроїв?
5. Назвіть основні відзнаки різних поколінь ЕОМ.
6. Яке кодування використовується для тексту?

*У цьому розділі понано інформацію про принципи архітектури і історія розвитку комп'ютерів, архітектуру комп'ютера, основні тенденції розвитку апаратного забезпечення комп'ютера, покоління ЕОМ і їх основні характеристики, представлення інформації в комп'ютері.*



# 4. МЕРЕЖНІ ТЕХНОЛОГІЇ

*З матеріалів цього розділу студент зрозуміє основи використання локальних та глобальних мереж.*

## 4.1. Локальні мережі

*Комп'ютерною мережею називається декілька комп'ютерів, об'єднаних один з одним. Такі системи характеризуються типом з'єднання, апаратним забезпеченням, ієрархією зв'язку і т. ін.*

Якщо комп'ютери з'єднані в систему, яка має кінцеву кількість робочих станцій, часто розташованих в межах одного підприємства (фірми), то така мережа називається *локальною (Local Area Network, LAN)*. Локальні мережі характерні тим, що відстані між компонентами мережі порівняно невеликі, як правило, не перевищують декількох кілометрів. Число робочих станцій такої системи наперед відоме, кожна станція має власний рівень в ієрархії системи, наперед відомі функції кожної робочої станції, тобто, існує головний комп'ютер, а також підлеглі йому комп'ютери.

### 4.1.1. Апаратне забезпечення локальних мереж

Мінімальним апаратним комплектом мереж є персональний комп'ютер, в материнську плату якого вставлена додаткова електронна плата, яка називається "мережною картою".

Мережна карта відповідає за підготовку даних до передачі по мережевому кабелю, передає або приймає дані, управляє потоком даних між комп'ютером та кабельною системою.

Всі ці дії кожна мережна плата виконує у строго визначеній послідовності, за встановленими правилами, які називаються – протоколами.

Кабель приєднується до мережної карти роз'ємом, що має назву "конектор". Типи конекторів залежать від типу кабелю.

Найчастіше це коаксіальний роз'єм та/або роз'єм для так званої "крученої пари". Також є плати з роз'ємами під оптоволоконний та мідний кабель. Перший дозволяє з'єднувати комп'ютери коаксіальним кабелем, схожим на телевізійний. Таке з'єднання забезпечує надійний зв'язок і є стійким до електромагнітних полів, утворених іншою електротехнічною апаратурою. "Кручена пара" часто представляє собою звичайний двожильний телефонний дріт. Найбільш дорогим є оптоволоконний кабель, який здійснює передачу інформації за допомогою світлових імпульсів.

Для підсилення сигналу, що слабшає при передачі на значні відстані, застосовуються лінійні підсилювачі або повторювачі (repeaters), які монтуються

через кожні 300-500 м для коаксіального кабелю, та через кожні 50-250 м для “крученої пари”.

Ще одним важливим елементом апаратного забезпечення локальних мереж є “концентратор”, тобто пристрій, що з’єднує кабелі з різних комп’ютерів в одну точку, сигнал з якої іде далі. Інколи він ще називається за аббревіатурою англійської назви – *HUB*. Завдяки цьому пристрою з’явилася можливість поєднувати групи близько розташованих комп’ютерів в єдину мережу, що значно здешевлює апаратну частину локальних мереж. *HUB* може бути активним чи пасивним.

Активні концентратори повинні бути ввімкнені до джерела електроенергії, вони можуть відновлювати і ретранслювати сигнали, мають різну кількість портів, тобто, до них можливі підключити різну кількість кабелів (сегментів).

Пасивні концентратори просто виконують з’єднання (монтажні панелі, комутуючі блоки).

Гібридні концентратори – це такі, до яких можна ввімкнути кабелі різних типів: коаксіальні, „кручену пару”, оптоволоконні.

Інтелектуальні концентратори, або свічі (switch), не тільки передають сигнали за адресою, а й слідкують за подіями в мережі та управляють ними. Вони, як правило, оснащені додатковим програмним забезпеченням, яке надає можливості адміністратору мережі слідкувати та програмувати кожний порт – тобто кожне з’єднання з робочою станцією.

Основними параметрами концентраторів, не враховуючи специфічних технічних характеристик, є швидкість передачі даних і кількість портів (вихідних роз’ємів, до яких підключається комп’ютер за допомогою мережного кабелю). Якщо говорити про швидкість передачі, то випускаються пристрої на 10 Мбіт/с, на 10/100 Мбіт/с та 1 Гбіт/с. Кількість портів також буває різною — 4, 5, 8, 12, 16, 24, 32.

При проектуванні мережі насамперед необхідно враховувати кількість комп’ютерів, що підключаються до неї, задачі, які повинні виконуватися мережею, врахувати ступінь безпеки і, виходячи з цього, обирати топологію мережі та обладнання.

#### **4.1.2. Топологія мереж**

Кожна мережна технологія має характерну тільки для неї топологію з’єднання вузлів мережі та метод доступу до середовища передачі даних. *Фізична топологія* визначає правила фізичних з’єднань вузлів. *Логічна* – напрямок потоків даних між вузлами мережі. Логічна та фізична топології незалежні одна від одної.

Існує три фізичні базових топології: зірка, шина та кільце.

Топологія “*Зірка*” (рис. 4.1) – всі комп’ютери безпосередньо з’єднуються з мережею через один вузол (концентратор).

Дана топологія потребує багато кабелю. Вся мережа виходить з ладу при проблемах центрального вузла, але вилучення окремих робочих станцій на ро-

боту решти комп'ютерів та мережі не вплине. "Зірка" на сьогодні є однією з найпопулярніших схем з'єднання.

Топологія "Шина" (рис. 4.2) – комп'ютери з'єднуються через загальний кабель. В минулому, була одна з найпопулярніших схем з'єднання, оскільки є більш економною по витратам кабелю, простою та водночас надійною, легко розширюється.

Недоліком є те, що при значних обсягах трафіку (передачі даних в одиницю часу) зменшується пропускна здатність мережі. Вихід з ладу кабелю припиняє роботу мережі.

Топологія „кільце” (рис. 4.3) – послідовне з'єднання персональних комп'ютерів (ПК), при якому останній з'єднаний з першим, за допомогою кабелю. Дані переміщуються кільцем в одному напрямку від ПК до ПК, причому кожен з них працює як повторювач, який підсилює сигнал і передає далі. Оскільки сигнал проходить через всі ПК, вихід з ладу одного з них веде до відмови всієї мережі. У кільце можна вмонтувати додаткові засоби, що вимикають несправний ПК, для продовження роботи мережі.

*Переваги:*

- усі комп'ютери мають рівний доступ до мережі;
- кількість користувачів істотно не впливає на продуктивність.

*Недоліки:*

- вихід з ладу одного комп'ютера може привести до відмови всієї мережі;
- кільцеві мережі чуттєві до розриву кабелю;
- важко локалізувати несправності;
- підключення нового користувача

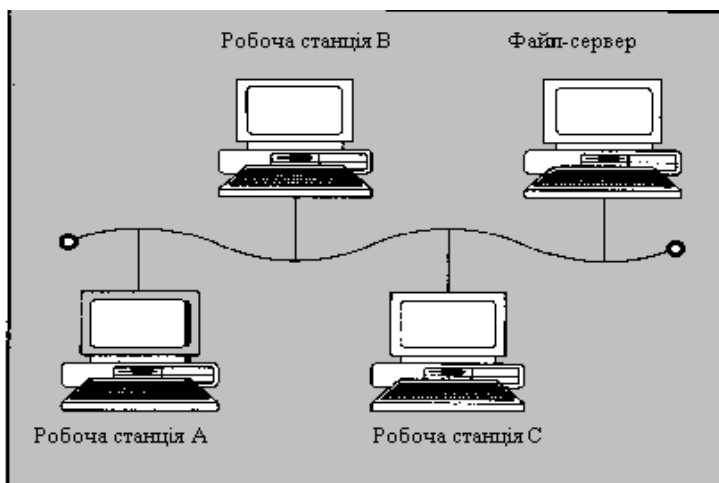


Рис. 4.2. Топологія „шина”

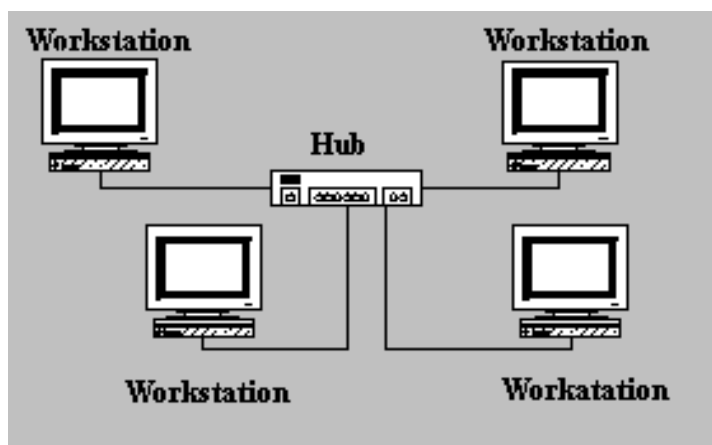


Рис. 4.1. Топологія „зірка”

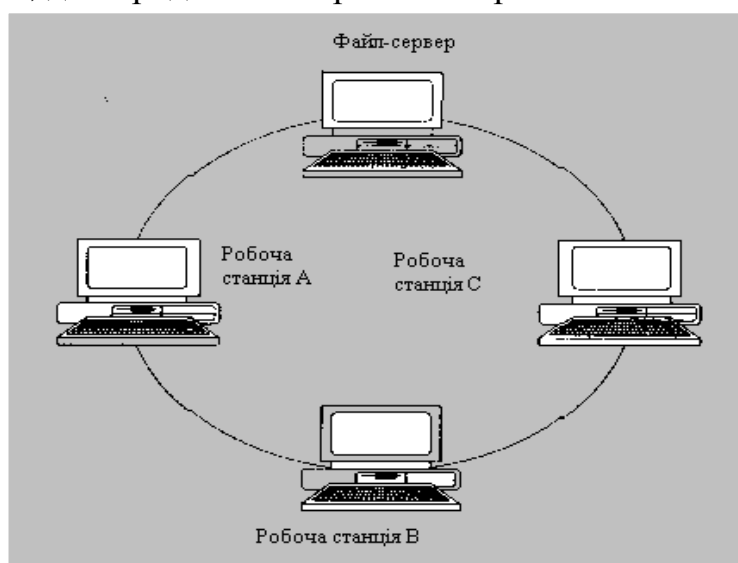


Рис. 4.3. Топологія „кільце”

чи зміна конфігурації мережі вимагає зупинки роботи всієї мережі.

Для підсилення переваг і усунення недоліків окремих різновидів використовують комбіновані топології.

„Зірка – шина” (рис. 4.4) – це комбінація топологій „зірки” та „шини”. Концентратори декількох мереж з топологією „зірка” об’єднуються за допомогою магістральної лінійної шини. Вихід з ладу одного ПК не позначається на роботі мережі, а концентратора – викликає вимкнення від мережі лише ввімкнених до нього комп’ютерів та концентраторів.

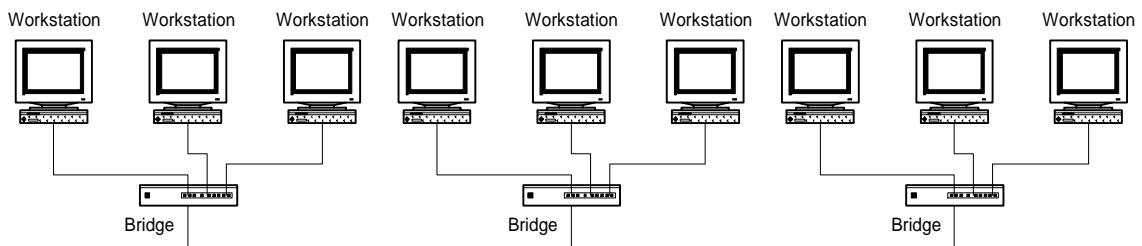


Рис. 4.4. Локальна мережа з топологією „зірка-шина”

„Зірка-кільце” – аналогічно попередній топології, але всі концентратори ввімкнені до головного концентратора, усередині якого реалізується кільце.

### 4.1.3. Поняття сервера та робочої станції

Існують мережі трьох типів:

- однорангові;
- з виділеним сервером;
- комбіновані.

*Однорангова* – мережа, в якій усі комп’ютери рівноправні. Кожен користувач передає в мережу певні ресурси свого ПК. Такі мережі найчастіше об’єднують небагато комп’ютерів (не більш 10).

*Переваги:*

- менші витрати на облаштування мережі;
- простота обслуговування;
- можливість використання кожним користувачем ресурсів інших ПК;
- зручність і простота роботи користувачів у мережі.

*Недоліки:*

- число ПК у мережі не перевищує 25-30;
- невеликий ступінь захисту даних.

Мережа з *виділеним сервером* – мережа, де є головний комп’ютер (сервер), який надає користувачам доступ до мережних ресурсів своєї мережі.

*Сервер* (server, служник – англ.) використовується для визначення користувачів, розподілу доступу до ресурсів, встановлення черги на доступ до інформації, забезпечує дублювання інформації, дотримання режиму секретності та зв’язок з іншими серверами, що мають власні групи даних. У мережі з виділеним сервером можуть бути визначені комп’ютери, до яких буде обмежений доступ з інших комп’ютерів. Крім того, є можливість організувати доступ до спіль-

них мережних принтерів, модемів та інших пристроїв, з будь-якого комп'ютера. На сервері можуть бути записані програми, якими користуються всі комп'ютери мережі.

Сервер рідко використовується як робоча станція, оскільки всі його ресурси по швидкодії, пам'яті, місткості довготермінових запам'ятовуючих пристроїв забирає робота по обслуговуванню мережі.

Всі інші комп'ютери мережі називаються робочими станціями.

Користувач робочої станції (клієнт, client, workstation) споживає мережні ресурси, які надає сервер.

Для роботи сервера може використовуватися спеціалізоване програмне забезпечення типу EtherNet, ArcNet та ін. Але для звичайного користувача системи на робочій станції його присутність не помітна, оскільки працює воно в фоновому режимі.

Переваги мереж з виділеним сервером:

- більш ефективно централізоване керування мережею;
- робочі станції, за апаратним та програмним забезпеченням, можуть бути досить простими і дешевими;
- спільне використання периферійного обладнання: принтерів, модемів, сканерів, що також здешевлює мережу;
- спільне використання програмного забезпечення;
- спільне використання даних.

Недоліки:

- більш висока вартість установки;
- складне налаштування системи.

*Комбіновані мережі* – суміщають кращі якості однорангових мереж та мереж з виділеним сервером. Вони використовують два типи операційних систем – серверні операційні системи (FreeBSD, Red Hat, Windows NT 4.0 Server, Windows 2000 Server, Windows 2003 Server Advanced, OS/2 та ін.) та операційні системи робочих станцій (Fedora Core, Red Hat, Windows 95,98,2000 Pro, XP та ін.).

Серверні операційні системи відповідають за спільне використання програм, периферії та даних у мережі, а операційні системи робочих станцій здійснюють доступ до ресурсів мережі та надають користувачам доступ до своїх ресурсів.

#### **4.1.4. Команди настройки локальної мережі засобами Windows**

Коли робоча станція (комп'ютер, на якому ви працюєте) знаходиться в локальній мережі, то всі інші робочі станції доступні для вас. Цей доступ забезпечується через будь-який файл-менеджер, за допомогою якого необхідно звернутися до розділу **Сетевое окружение** (рис. 4.5).

Там ви побачите перелік груп, в які з'єднані робочі станції. Зверненням до групи є просте натискання лівої кнопки миші або кнопки **Enter**.

Для кожної групи відкриється перелік робочих станцій, кожна зі своїм іменем (рис. 4.6). Якщо ми виберемо те чи інше ім'я, нам стануть доступні ті папки чи диски, які на цій робочій станції зроблені доступними для використання в мережі.



Рис. 4.5. Вікно „Сетевое окружение”

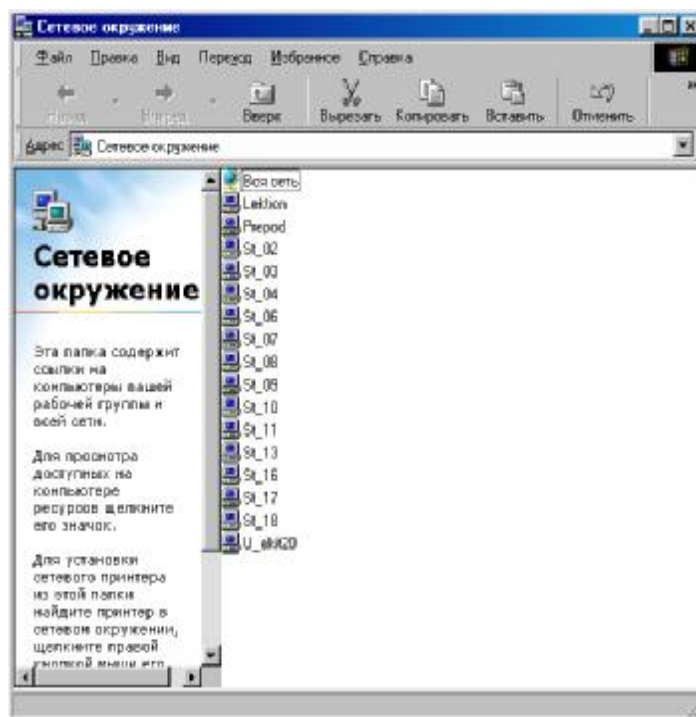


Рис. 4.6. Вікно з переліком комп'ютерів (робочих станцій) Вашої мережі


В деяких випадках, на ці елементи встановлений пароль, який обмежує доступ. Операції по зміні статусу того чи іншого елементу робочої станції виконуються натисканням правої кнопки миші. В контекстovому меню треба вибрати пункт “Свойства”. У вікні, що з'явиться, можна настроїти параметри об'єкту таким чином, щоб він був доступний при звертанні по мережі.

Для того, щоб зробити диски та папки доступними для використання у мережі, треба вибрати у вікні **Свойства** вкладку **Доступ** (рис. 4.7). Потім у вікні, що відкриється треба вказати опцію **Общий ресурс**, ввести мережне ім'я чи погодитись з обраним за замовчуванням та ввести нотатки, що необов'язково. У полі **Имя** необхідно ввести імена користувачів, яким дозволено доступ до даного об'єкта. Для цього треба натиснути кнопку **Добавить...**, у результаті відкриється вікно **Добавление пользователей**, в якому зі списку користувачів треба вибрати необхідних, наприклад, **Все пользователи**. З полів **Только чтение**, **Полный доступ** та **Особый** треба вибрати потрібне, наприклад, **Полный доступ**, натиснувши відповідну кнопку. Після цього закрити вікно, натиснувши кнопку **ОК**.

**Полный доступ** надає можливість відкривати диски, папки, файли, які знаходяться на мережнім диску, проглядати їх, редагувати файли, копіювати та переміщати об'єкти з одного ПК на інший.

Доступ **Только чтение** надає всі перераховані вище можливості, але не дозволяє редагувати файли, а тільки читати їх.

Аналогічні дії робляться для того, щоб зробити доступним для використання в мережі принтер, (тобто буде можливість з будь-якого ПК роздрукувати будь-який файл). При цьому, змінюється статус принтера, який з локального стає мережним.

Якщо внизу значку об'єкта (диск чи папка) знаходиться рука  – він доступний для роботи, а саме є мережним.

Копіювання та переміщення об'єктів у мережі провадиться так само, як у програмах Windows, таких як **Проводник** чи **Мой компьютер**.

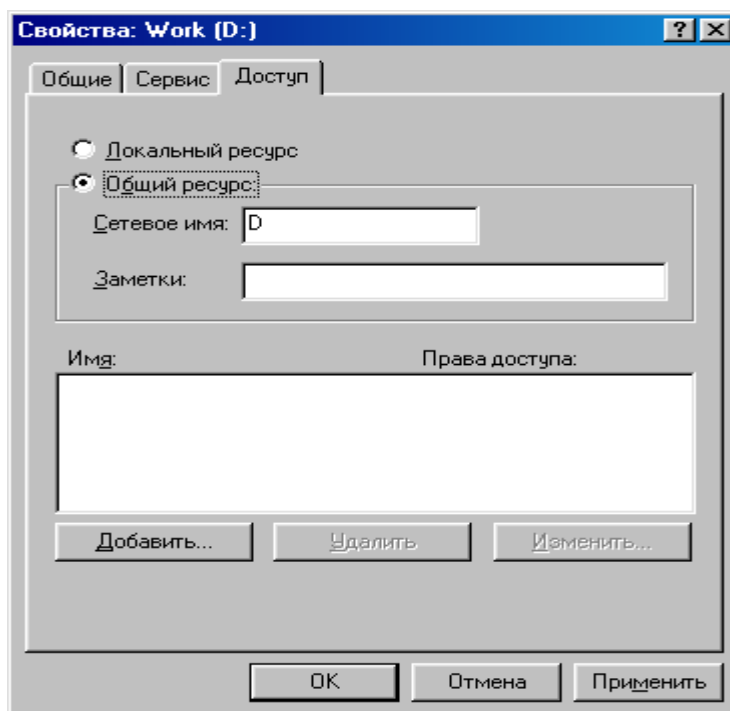


Рис. 4.7. Вікно „Свойство”

#### 4.1.5. Програми обміну письмовими повідомленнями поміж різними робочими станціями локальної мережі

##### 4.1.5.1. WinPopup

WinPopup входить в стандартну поставку Windows та має досить простий інтерфейс (рис. 4.8).

Для відправки повідомлення потрібно послідовно виконати наступні дії:

1. Вибрати **Отправить** в меню **Сообщение**.

2. Вказати тип адресата: конкретний комп'ютер чи всі станції робочої групи.

3. Ввести ім'я комп'ютера чи робочої групи.

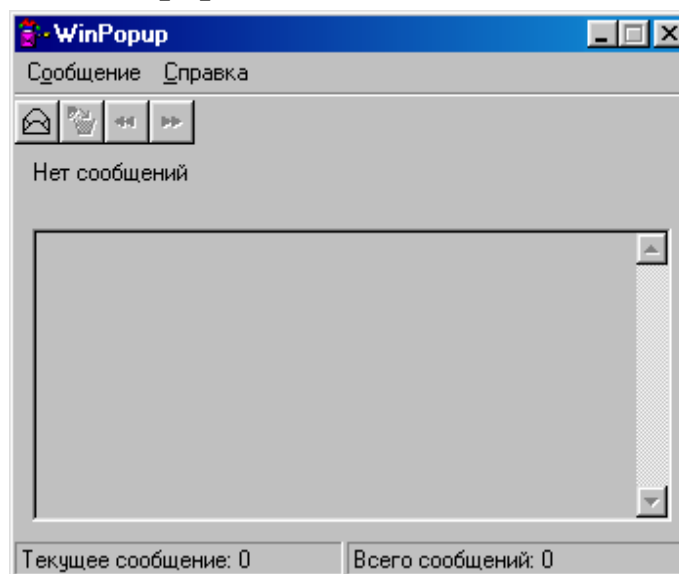


Рис. 4.8. Вікно програми WinPopup

#### 4. Набрати текст повідомлення і натиснути кнопку **Отправить**.

При відправленні повідомлення з'являється додаткове вікно, яке містить назви можливих адресатів ваших повідомлень.

Якщо WinPopUp не завантажена, повідомлення на цю робочу станцію не надійде, а просто зникне без будь-якого сповіщення. Щоб переглянути спливаюче повідомлення, вкажіть на значок WinPopUp на панелі задач і натисніть кнопку миші.

На екрані з'явиться поточне повідомлення. Для перегляду інших повідомлень виберіть **Следующее** чи **Предыдущее** в меню **Сообщение**.

Щоб видалити поточне повідомлення, виберіть **Удалить** в меню **Сообщение**. Щоб видалити всі повідомлення, виберіть **Очистить все** в меню **Сообщения**.

### 4.1.5.2. Intranet Chat

Ця програма більш зручна, хоча і виконує аналогічну з WinPopUp функцію. Вона розташована за кнопкою "Пуск-Программы-Intranet Chat".

Основні елементи інтерфейсу (Рис. 4.9):

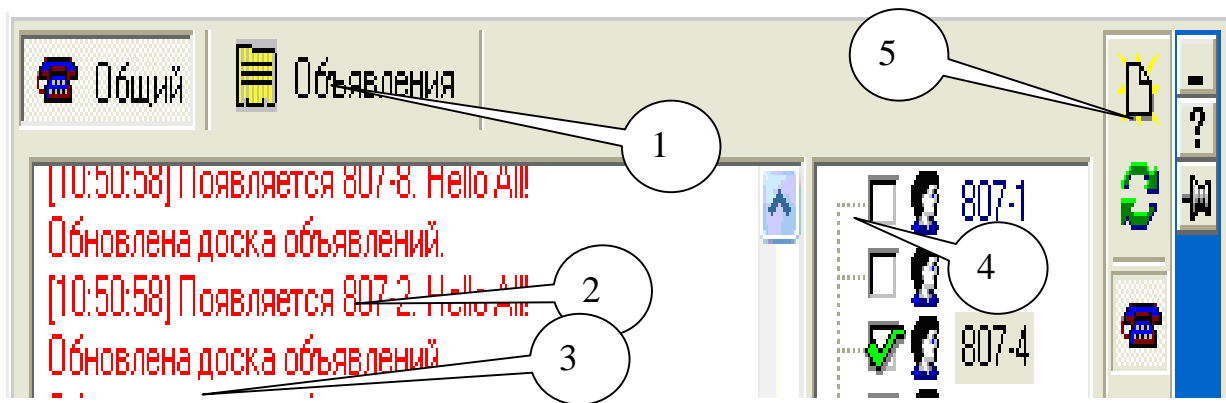


Рис. 4.9. Інтерфейс програми Intranet Chat

1 – Перемикач режимів доступу.

2 – Вікно повідомлень.

3 – Зона написання повідомлення.

4 – Вікно адресації повідомлень, які надійдуть до відмічених комп'ютерів.

5 – Панель інструментів, значення яких з гори до низу наступні: Очистити, Відновити список користувачів, Стан – ніяких обмежень, Стан – не турбувати на масові повідомлення, Стан – не турбувати по всім повідомленням, Стан – мене немає, Вибрати швидке повідомлення, Налаштування, Про програму, Вихід.

При виборі інструменту „настройки”, з'являється додаткове вікно (рис. 4.10).



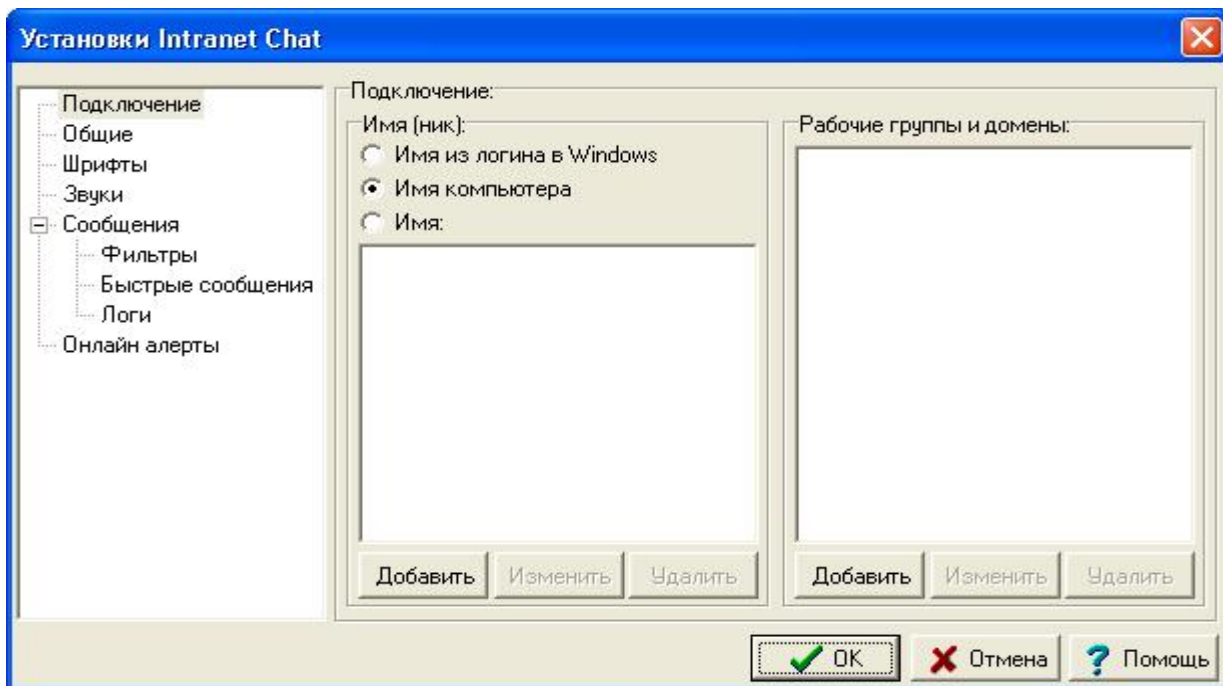


Рис. 4.10. Интерфейс настроек программы Intranet Chat

## 4.2. Глобальні мережі

Глобальні (Wide Area Network, WAN) мережі не мають точно визначеної кількості робочих станцій, не мають і чіткої ієрархії. До глобальних мереж може приєднатися і від'єднатися невизначена кількість комп'ютерів. Причому, фізичне місцезнаходження кожного комп'ютера в глобальній мережі теж не визначено.

Простіше кажучи, Інтернет - це комплекс зв'язаних між собою комп'ютерних мереж, який дозволяє компаніям, організаціям, окремим користувачам, навчальним закладам і урядам, організувати загальний доступ до інформації з усього світу.

Інтернет включає Всесвітню Павутину (World Wide Web), яка дозволяє переглядати документи з різним форматуванням і малюнками, багато веб-сторінок містять посилання на інші веб-сторінки, тому можна легко переглядати великі обсяги інформації, просто клацаючи мишею і мандруючи по мережі.

Internet – глобальна комп'ютерна мережа, що охоплює увесь світ. Щомісяця розмір мережі збільшується на 7-10%. Internet утворює ядро, що забезпечує зв'язок різних інформаційних мереж, що належать різним установам в усьому світі.

### 4.2.1. Класифікація мереж

Глобальні мережі можна класифікувати за наступними ознаками:

1. За типом засобів комунікації: наземні багатовузлові мережі; супутникові радіомережі; комбіновані мережі.
2. За способом комутації повідомлень: комутація каналів; комутація повідомлень; комутація пакетів; адаптивна комутація.

3. За вибором маршруту передачі повідомлення: фіксовані шляхи; спрямований вибір шляху; випадкові шляхи; лавинний спосіб.

#### 4.2.2. Апаратне забезпечення глобальних мереж

Мінімальним апаратним комплектом мереж є персональний комп'ютер, в материнську плату якого вставлена додаткова електронна плата, яка називається "модем" (це скорочення від фрази "модулятор-демодулятор"). Інколи модем є зовнішнім пристроєм, що з'єднується з комп'ютером системним шнуром через послідовний порт COM2.

Інший вихід цього пристрою з'єднується з телефонною мережею. Саме телефонні канали зв'язку дозволяють поєднати ваш комп'ютер з глобальною мережею. Це з'єднання встановлюється з одним з тих комп'ютерів, що тримають постійний зв'язок з іншими такими ж постійно діючими комп'ютерами. Вони називаються серверами мережі, але інколи їхні функції відрізняються від серверів локальних мереж.

Сервери глобальної мережі працюють як поштові скриньки: приймають повідомлення від своїх клієнтів та передають далі за адресами до інших клієнтів чи груп клієнтів, користуючись своїми зв'язками з іншими серверами. Ці зв'язки утримуються не з одним, а з декількома іншими комп'ютерами, тому структура глобальної мережі нагадує павутину. Саме тому англійською мовою і вживається термін "net", тобто "павутина".

Щоб стати клієнтом того чи іншого сервера, потрібно укласти угоду на підключення вашого комп'ютера до глобальної мережі. Ви отримаєте паролі та номери телефонів для доступу до інформації в глобальній мережі.

#### 4.2.3. Відмінність в режимах роботи глобальних мереж "off-line" та "on-line"

Термін "off-line" дослівно означає "вимкнути з лінії". Для глобальних мереж це означає такий режим роботи сервера, коли він, отримавши інформацію від клієнта, чекає сеансу зв'язку з потрібним йому сервером, що лежить на маршруті передачі повідомлення. Наступний сервер працює так само. Отже, сама передача інформації нагадує роботу звичайної пошти.

Очевидно, що незважаючи на велику швидкість передачі інформації, сам час її проходження в межах не те що земної кулі, а навіть в межах України, може розтягуватися на одну-дві доби. В такому режимі працює глобальна банківська мережа України і тому час зарахування грошей на рахунок, згідно існуючим нормативам, може досягати чотирьох діб.

Для миттєвого з'єднання будь-якого комп'ютера з іншим у межах глобальних мереж було винайдено режим роботи "on-line". Всі сервери в мережі одночасно підтримують зв'язок із усіма іншими серверами, або мають можливість негайно за запитом клієнта, встановити цей зв'язок.

Фізично це означає зустрічну передачу інформації в межах глобальної мережі. Для забезпечення надійності в таких складних умовах, кожна глобальна

мережа розробляє і використовує певну систему структури представлення і кодування інформації, так званий “мережний протокол”.

#### 4.2.4. Основні поняття і протоколи Internet

Робота мережі Internet базується на використанні сімейства комунікаційних протоколів TCP/IP (Transmission Control Protocol/Internet Protocol – Протокол керування передачею/ Протокол Internet). Протокол – правила передачі даних у мережі, яких повинні дотримуватися всі компанії, щоб забезпечити сумісність використовуваного апаратного і програмного забезпечення. Кожне повідомлення для передачі по мережі поділяється на короткі пакети однакового розміру і визначеної структури. *Пакет* – частина повідомлення, що задовольняє стандарту TCP та IP. TCP/IP – це множина комунікаційних протоколів, що визначають, як комп'ютери різних типів можуть спілкуватися між собою.

Протокол TCP розбиває повідомлення на пакети і нумерує їх, щоб при одержанні інформації можна було правильно скласти повідомлення. Мережа передає пакети по черзі за допомогою протоколу IP. До кожної одержаної порції інформації протокол IP додає службову інформацію, яка містить адреси відправника і одержувача інформації. Оскільки окремі пакети можуть передаватися різними шляхами, порядок надходження пакетів може бути порушений. Після одержання всіх пакетів, TCP розміщує їх в певному порядку і складає в єдине ціле. Крім того, після пересилання пакета, кожен вузол (комутаційний сервер) очікує підтвердження того, що пакет одержано належним чином, інакше відбувається повторна передача. Це дає змогу запобігти ситуації, коли велике повідомлення передається знову і знову через єдину помилку.

Насправді дані переміщуються:

- зверху вниз від прикладного рівня до фізичного;
- в рамках фізичного рівня горизонтально по мережному кабелю до комп'ютера - приймача даних;
- потім одержані дані рухаються вгору по рівнях мережної моделі.

Різні сервіси, що включаються в TCP/IP, і функції цього сімейства протоколів можуть бути класифіковані за типом виконуваних задач. Згадаємо лише основні протоколи, загальне їхнє число нараховує не один десяток:

- *Транспортні протоколи*: TCP (Transmission Control Protocol), UDP (User Datagram Protocol) – керують передачею даних між двома комп'ютерами;
- *Протоколи маршрутизації*: IP (Internet Protocol), ICMP (Internet Control Message Protocol), RIP (Routing Information Protocol) і ін. – обробляють адресацію даних, забезпечують фактичну передачу даних і визначають найкращі шляхи до адресата;
- *Протоколи підтримки мережної адреси*: DNS (Domain Name System), ARP (Address Resolution Protocol) і ін. – забезпечують ідентифікацію комп'ютера з унікальним номером та іменем;
- *Протоколи прикладних послуг*: FTP (File Transfer Protocol), TELNET і ін. – програми, котрі комп'ютер (чи користувач) використовує для одержання дос-

тупу до різних послуг: передача файлів між комп'ютерами, віддалений термінальний доступ до системи та ін;

- *Шлюзові протоколи:* EGP (Exterior Gateway Protocol), GGP (Gateway-to-Gateway Protocol) і IGP (Interior Gateway Protocol) – допомагають передавати по мережі повідомлення про маршрутизації, інформацію про стан мережі, а також обробляти дані для локальних мереж;

- *Інші протоколи:* SMTP (Simple Mail Transfer Protocol), NFS (Network File System) та ін. – вирішують важливі задачі, що не відносяться до категорій, згаданих вище. Це передача повідомлень електронної пошти, робота з каталогами і файлами віддаленого терміналу та ін.

Згідно стандарту OSI протоколи розділяються на рівні: фізичний, каналний, мережний, транспортний, сеансовий, представницький та прикладний. За концептуальними ознаками їх можна розділити на чотири види:

**1-й вид – прикладний.** Він включає такі типи:

HTTP (Hyper Text Transfer Protocol) – для передачі інформації у світову павутину. Саме в ньому передаються веб-сторінки.

FTP (File Transfer Protocol) – для пересилання бінарних (текстових) файлів.

SMTP (Simple Mail Transfer Protocol) – для пересилання електронної пошти.

POP (Post Office Protocol) – для отримання і зберігання пошти.

NNTP (Net News Transfer Protocol) – для передачі новин у телеконференціях.

TELNET – емуляція терміналу для роботи з віддаленими серверами.

**2-й вид – транспортний.** Тобто такий, яким проводиться передача зазначених вище даних.

TCP (Transmission Control Protocol) – для керування процесом передачі інформації поміж серверами.

UDP (User Datagram Protocol) – альтернативний TCP-протоколу.

**3-й вид – для передачі даних між різними мережами.**

IP (Internet Protocol) – його завдання забезпечити якісну передачу блоків інформації.

**4-й вид – для передачі даних у локальних мережах.**

SPX/IPX – для мереж типу Novell.

#### 4.2.5. IP-адресація комп'ютерів у Internet

Кожен ПК, підключений у Internet, має свою IP-адресу. Вона завжди має довжину 32 біта і складається з чотирьох частин по 8 бітів, іменованих відповідно до мережної термінології *октетами* (octet). Це означає, що кожна частина може мати значення в межах від 0 до 255. Чотири частини поєднують у запис, у якому кожне восьмибітове значення відокремлюється крапкою. Наприклад, 147.120.3.28 – це IP-адреса. Коли йде мова про мережну адресу, мають на увазі саме IP-адресу. У новому протоколі IP6 адреси довші, але загальний принцип залишається тим же.

Якби використовувалися всі 32 біта в IP-адресі, то вийшло б понад чотири мільярди можливих адрес – більш ніж досить для майбутнього розширення Internet. Однак, деякі комбінації бітів зарезервовані для спеціальних цілей, що зменшує число потенційних адрес. Крім того, 8-бітні четвірки згруповані спеціальними способами в залежності від типу мережі, так що фактичне число можливих адрес ще менше.

Будь-яка IP-адреса складається з двох частин: адреси мережі (ідентифікатора мережі, Network ID) і адреси хоста (ідентифікатора хоста, Host ID) у цій мережі. Завдяки такій структурі IP-адреси ПК у різних мережах можуть мати однакові номери. Але оскільки адреси мереж різні, ці ПК ідентифікуються однозначно і не можуть бути поплутані один з одним.

IP-адреси виділяються в залежності від розміру організації і типу її діяльності. Якщо це невелика організація, то швидше за все в її мережі небагато ПК (а, отже, IP-адрес). У великій корпорації навпаки, можуть бути тисячі (а то і більше) ПК, об'єднаних у безліч з'єднаних між собою локальних мереж. Для забезпечення максимальної гнучкості IP-адреси, в залежності від кількості мереж і ПК в організації, розділяються на класи А, В і С. Ще існують класи D і E, але вони використовуються у специфічних службових цілях.

Три класи IP-адрес дозволяють розподілити їх у залежності від розміру мережі організації. Оскільки 32 біта – припустимий повний розмір IP-адреси, класи розбивають чотири 8-бітні частини адреси на адресу мережі й адресу хоста в залежності від класу. Так, адреса мережі класу А визначається першим октетом IP-адреси. Значення першого октету, що знаходиться в межах 1-126, зарезервовано для гігантських транснаціональних корпорацій і найбільших провайдерів. Таким чином, у класі А може існувати усього лише 126 великих мереж компаній, кожна з яких може містити майже 17 мільйонів комп'ютерів.

Клас В використовує два перших октети як адресу мережі, а перший октет тут може приймати значення в межах 128-191. У кожній мережі класу В може бути 65 тис. ПК, такі мережі мають найбільші університети й інші великі організації. Відповідно, у класі С під адресу мережі вже приділяється три перших октети, а значення першого октету можуть бути в межах 192-223. Це найпоширеніші мережі, їх число може перевищувати два мільйони, а число ПК (хостів) у кожній мережі – до 254. Слід зазначити, що «розриви» у припустимих значеннях першого октету між класами мереж з'являються через те, що один чи декілька бітів зарезервовані на початку IP-адрес для ідентифікації класу.

У табл. 4.1 підсумовано вищесказане.

Якщо IP-адреса деякого ПК – 147.14.87.23, це означає, що він знаходиться в мережі класу В, мережний ідентифікатор – 147.14, а його унікальний номер в даній мережі – 87.23. Якщо IP-адреса – 221.132.3.123, то ПК знаходиться в мережі класу С з мережним ідентифікатором 221.132.3 і ідентифікатором хосту 123. Кожного разу, коли надсилається повідомлення якому-небудь хост-комп'ютеру в Internet, IP-адреса використовується для вказівки адреси відправника й одержувача.

Таблиця 4.1.

## Систематизація класів мереж та IP-адрес комп'ютерів

Мережа	Діапазон адрес
A	0.0.0.0 – 127.255.255.255
B	128.0.0.0 – 191.255.255.255
C	192.0.0.0 – 223.255.255.255
D	224.0.0.0 – 239.255.255.255
E	240.0.0.0 – 255.255.255.255

### 4.2.6. Постачальник послуг Інтернету (ISP або Провайдер)

*Сервіс-провайдери – установи, які надають комерційні послуги з підключення до Internet (Internet Service provider, ISP).*

Сервіс-провайдер обов'язково має комп'ютерну мережу з постійним сполученням з Інтернет, у складі якої є сервери доступу. З їх допомогою здійснюється підключення до Інтернет абонентів – окремих користувачів або користувачів локальних мереж організацій і установ.

Бекбон (Backbone) – загальна, первинна лінія зв'язку первинних провайдерів. Сервер – потужний комп'ютер, який виконує роль координатора в мережі. В залежності від застосування виділяють декілька типів серверів. Web-сервери працюють зі світовою павутиною (World Wide Web). Поштові сервери забезпечують прийом та передачу електронної пошти. FTP-сервери зберігають на собі інформаційні файли. NNTP-сервери використовуються для обробки новин в мережі. DNS-сервери перекладають буквені адреси в цифрові. Останні три типи названі по типу протоколів передачі даних в мережі.

Загальна схема роботи провайдера показана на рис. 4.11. Тут, К – комп'ютери клієнтів, П – провайдери, ПМ – поштова машина – комп'ютер, який з'єднується з бекбоном, БК – база клієнтів – комп'ютер, який містить в собі дані клієнтів (веб-сторінки, поштові повідомлення, тощо), КМ – клієнтська машина, комп'ютер з додатковими електронними пристроями, який забезпечує одночасне з'єднання декількох клієнтів з провайдером.

Різні постачальники послуг Інтернету пропонують різні послуги за різними цінами, наприклад, електронну пошту чи власну домашню веб-сторінку. Оплата за Інтернет – погодинна або за місяць, без обмежень в часі користування.

### 4.2.7. Способи з'єднання з Інтернетом

Способи з'єднання з Інтернетом бувають двох типів.

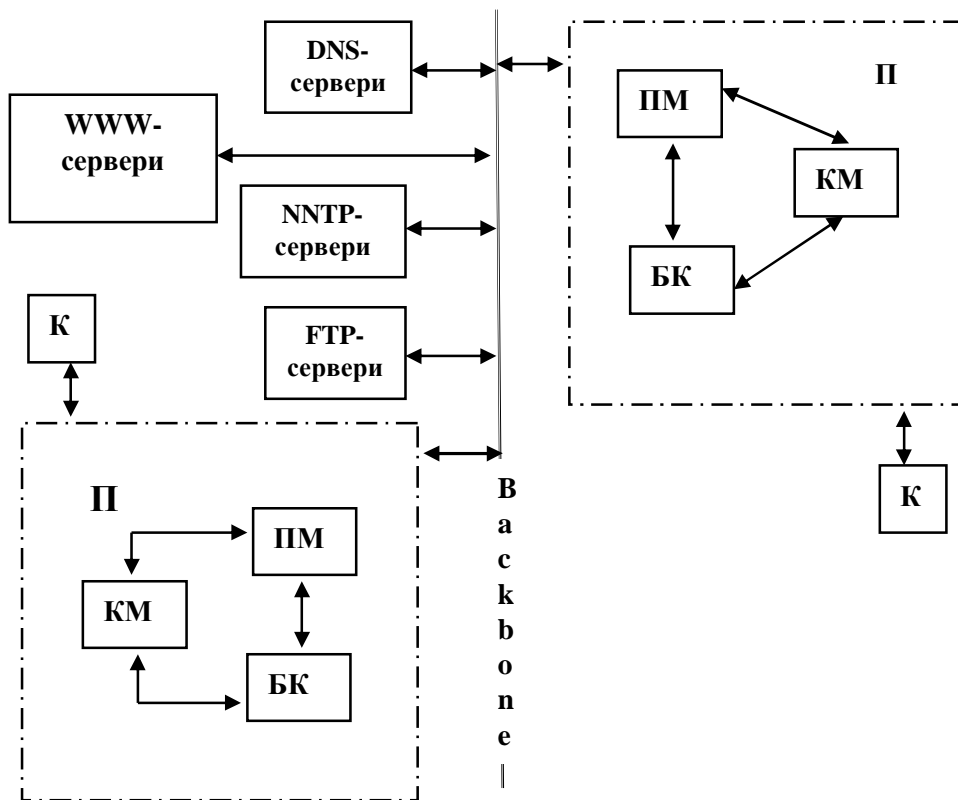


Рис. 4.11. Загальна схема роботи провайдера

*Постійний*, коли користувач постійно зв'язаний з комп'ютером провайдера. Кожен клієнт Інтернету повинен мати свою адресу (так звану, *IP-адресу*). Якщо ви постійно зв'язані з провайдером, ваша IP-адреса є постійною. Наприклад, 121.123.221.

Для постійного зв'язку використовується будь-яка лінія, сигнал в якій має не аналоговий, а цифровий характер. Такі лінії називаються ISDN-лініями і вони мають найбільшу швидкість проходження інформації, але вони багато коштують, тому частіше користуються тимчасовим зв'язком. Цей спосіб з'єднання називається *dial up* (англ., набирати номер телефону) і потребує модему, який сам набере номер провайдера, передасть потрібні коди і паролі. В цьому випадку ви отримаєте тимчасову IP-адресу, яка називається *динамічною*. Часто, провайдери призначають постійні IP-адреси, навіть при *dial up*-з'єднанні. Ця адреса необхідна тільки для первинного приєднання до Інтернету.

*Виділена лінія* – приєднання проходить за допомогою мережних карт, які використовуються і в локальних мережах. Те ж, коли ваш комп'ютер знаходиться в локальній мережі. Якщо ви приєднані через модем, то використовуються протоколи SLIP (Serial Line Internet Protocol) – протокол для серійного підключення та PPP (Point to Point Protocol) – протокол з точки на точку. Другий тип протоколу має більше можливостей порівняно з першим, навіть забезпечує одночасну передачу інформації по різних протоколах.

Для Windows існує власна термінологія протоколів: CSLIP – стиснений SLIP, стандартний SLIP, PPP та MRN – для з'єднання з сервером Novell NetWare. DNS (Domane Name Server) – сервери, де IP адреса вашого сервера порівнюється з адресою сервера, написаною літерами. Адреси цих серверів треба

знати, інакше ви не зможете користуватися адресами серверів, складених з літер. Всі сервери мають унікальні адреси згідно протоколу TCP/IP – 32-бітові, типу: 124.172.10.27. Щоб краще запам'ятати і легше оперувати з такими адресами, застосовуються адреси, складені з літер, наприклад: [www.gov.ua.net](http://www.gov.ua.net) – адреса серверу Уряду України. WWW – аббревіатура від World Wide Web, GOV– (від слова government) уряд, UA – Україна, NET – мережа.

В усіх випадках, комп'ютер спочатку шукає потрібну адресу в локальній мережі, а потім застосовує так званий Default Gateway – стандартний шлюз, який підказує, де шукати потрібну адресу в Інтернеті.

#### 4.2.8. Доменна система адрес в Інтернеті

Крім IP-адрес, для ідентифікації конкретних хостів у мережі використовується так зване доменне ім'я хоста (Domain host name). Так само як і IP-адреса, це ім'я є унікальним для кожного комп'ютера (хоста), підключеного до Internet, тільки тут замість цифрових значень адреси застосовуються слова. У даному випадку поняття домену означає сукупність хостів Internet, об'єднаних за якоюсь ознакою (наприклад, за територіальною). Зрозуміло, що використання доменного імені хосту було введено тільки для того, щоб полегшити користувачам задачу запам'ятовування імен потрібних їм комп'ютерів.

Доменна система імен являє собою метод призначення імен шляхом покладання на різні групи користувачів відповідальності за підмножини імен. Кожен рівень у цій системі називається доменом. Домени відокремлюються один від одного крапками. Приклади доменних імен:

dsau.dp.ua;            nic.ddn.mil; yoyodyne.com.

В імені може бути будь-яке число доменів, але більш п'яти зустрічається рідко. Кожен наступний домен в імені (якщо дивитися зліва – праворуч) вище попереднього. В імені **dsau.dp.ua** елемент **dsau**– ім'я реального комп'ютера з IP-адресою (рис. 4.12).

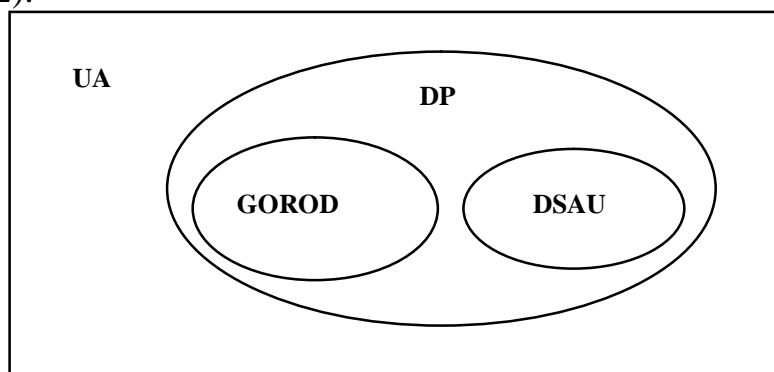


Рис. 4.12. Доменна структура імен

Головним доменом є UA – тобто, Україна. Підлеглим йому є домен DP тобто – Дніпропетровськ. Домен ще нижчого рівня це DSAU – домен Дніпропетровського державного аграрного університету. Рівнозначним йому є домен GOROD.



Кожна група може створювати і змінювати імена, що знаходяться під її контролем. Якщо **dp** вирішить створити нову групу і назвати її **dsau**, вона може ні в кого не запитувати дозволу. Усе, що потрібно зробити – це додати нове ім'я у свою частину всесвітньої бази даних, і рано чи пізно той, кому потрібно, довідається про це ім'я (**dsau.dp.ua**). Аналогічним чином **dsau** може купити новий комп'ютер, привласнити йому ім'я і включити в мережу, не запитуючи ні в кого дозволу. Якщо всі групи, починаючи з **ua** і нижче, будуть дотримуватись правил і забезпечувати унікальність імен, то ніякі дві системи в Internet не будуть мати однакового імені. У Вас можуть бути два комп'ютери з ім'ям **fred**, але лише за умови, що вони знаходяться в різних доменах (наприклад, **fred.cso.uiuc.edu** і **fred.ora.com**).

При запровадженні мережі, вона в основному локалізувалась на території США і більш важливою була організаційна чи галузева характеристика. Коли Internet стала міжнародною мережею, виникла необхідність надати закордонним країнам можливість контролю за іменами систем, що знаходяться в них. Для цієї мети створений набір доменів з двох літер, що відповідають доменам вищого рівня для цих країн. Оскільки **ua** – код України, то комп'ютер на території України може мати таке ім'я: **hockey.guelph.ua**. Загальне число кодів країн – близько 300, комп'ютерні мережі існують приблизно у 170 з них.

Остаточний план розширення системи присвоєння імен ресурсів у Internet був оголошений комітетом ІАНС (International Ad Hoc Committee). Крім того, у рішеннях ІАНС зазначено, що засновується 28 уповноважених агентств по присвоєнню імен в усьому світі. Як заявлено, нова система дозволить успішно перебороти монополію, що була нав'язана єдиним уповноваженим – компанією Network Solutions. Усі нові домени будуть розподілені між новими агентствами, а колишні будуть відслідковуватися спільно Network Solutions і National Science Foundation.

Перетворенням доменних імен у зрозумілі для ПК IP-адреси займається Domain Name System (DNS, Доменна система імен) – сервіс, забезпечуваний TCP/IP. Саме завдяки роботі DNS ви можете не запам'ятовувати IP-адресу, а використовувати набагато більш просту доменну адресу. Система DNS транслює символічне доменне ім'я ПК у IP-адресу.

Доменна структура імен дозволяє створити 2 типи адрес:

- e-mail – електронну адресу, яка зліва від доменного імені містить ще символ «@», що відділяє „логін” (умовне, обране вами ім'я) для листування через електронну пошту. Наприклад: **info@dsau.dp.ua** – офіційна електронна адреса ДДАУ.

URL – адресу веб-сторінки, яка знаходиться на якомусь віддаленому комп'ютері. Спочатку в цій адресі йде конструкція, яка визначає протокол передавання даних веб-сторінок “http” та тип сервера, який знайде цю адресу “www”. Далі йде доменна адреса, яка завершується іменем папки на комп'ютері провайдера, в якій знаходиться потрібна нам веб-сторінка, та ім'я цієї сторінки. Наприклад: <http://www.nmu.org.ua/ekit/index.html> – адреса веб-сторінки кафедри

ЕКІТ. Розділові знаки, такі як двокрапка та похила риска (так званий „прямий слеш”) є обов’язковими елементами URL.

Наведемо деякі галузеві домени верхнього рівня: com – Комерційні організації; edu – Навчальні заклади (університети, середні школи і т.д.); gov – Урядові заклади (крім військових); mil – Військові установи (армія, флот і т.д.); org – Інші організації; net – Мережні ресурси; firm – Для ділових ресурсів мережі; store – Для торгівлі; web – Для організацій, що мають відношення до регулювання діяльності в WWW; rec – Ігри і розваги; info – Надання інформаційних послуг; nom – Для індивідуальних ресурсів, а також тих, хто шукає свої шляхи реалізації, що відсутні в приведеному скороченому списку.

### 4.3. Індивідуальні завдання №2

1. Візьміть у викладача ім'я своєї ланки, номер свого варіанта, а також мережне ім'я та пароль для роботи в мережі.
2. Перевантажте, якщо це необхідно, ПК у режимі переініціалізації мережного імені і паролю. (Пуск–Завершение работы–Войти в систему под другим именем.)
3. Уведіть мережне ім'я і пароль для входу в локальну мережу.
4. Увійдіть в мережне оточення і перегляньте структуру локальної мережі. Вивчіть поняття сервера, домена і робочої групи.
5. Створіть на своєму ПК папку D:\Student\ Ім'я\_ланки\_Прізвище. Визначте для неї спеціальний доступ: *запис у файли, читання файлів, надання доступу до файлів* робочим станціям своєї ланки.
6. Створіть, у створеній вище папці, текстовий документ, де опишіть дії для входу в локальну мережу і режимах доступу до диска (папки).
7. Перегляньте інформацію, записану у файлах студентів вашої ланки.
8. Визначте для своєї папки повний доступ робочим станціям своєї ланки.
9. Підключіть мережні диски D: колег по ланці.
10. Перепишіть файли з робочих станцій своєї ланки на свій ПК.
11. Визначте повний доступ для диска A: свого ПК, робочим станціям своєї ланки. Запишіть інформацію, що зібралася в робочій папці, на свою дискету, використовуючи при цьому дисковод A: іншої робочої станції ланки.
12. Надайте повний доступ всім користувачам до принтера для роботи в мережі. Роздрукуйте файл.
13. Зробіть доступ *Только для чтения* всім користувачам до дисків, папок та файлів для роботи в мережі.
14. Скопіюйте папку зі свого ПК на інший.
15. Скопіюйте свої файли зі свого ПК на інший.
16. Перемістіть папку зі свого ПК на інший.
17. Перемістіть свої файли зі свого ПК на інший.
18. Скопіюйте папку з сусіднього ПК на свій.
19. Скопіюйте файли з сусіднього ПК на свій.
20. Перемістіть папку із сусіднього ПК на свій.

21. Визначте, в чому різниця між копіюванням та переміщенням?
22. Подивіться зміст диска D:\ сусіднього комп'ютера.
23. Завантажте із сервера документ Base.
24. Виберіть дані по кварталу (номер кварталу – порядковий номер у ланці) і скопіюйте їх у документ на свою робочу станцію.
25. Завантажте програму WinPopUp або Intranet Chat.
26. На підставі скопійованих даних, побудуйте гістограму *Доходів-Витрат* по місяцях.
27. Порахуйте прибуток для кожного місяця (курси валют по місяцях знаходяться у файлі в адміністратора ланки).
28. Порахуйте загальний прибуток за квартал.
29. Наприкінці створеного звіту внесіть дату і час створення вашого документу.
30. Використовуючи засіб спілкування по мережі, розішліть колегам по ланці повідомлення про завершення свого звіту.
31. При одержанні від усіх колег по ланці відповідних повідомлень, сформууйте зведений документ (з чотирьох частин: 1-й квартал, 2-й квартал, 3-й квартал, 4-й квартал).
32. Розрахуйте загальний прибуток за рік, і доповніть документ гістограмою *Доходів-Витрат* по кварталах.
33. Запишіть свій зведений документ у папку на сервері і на власну дискету.
34. Відправте адміністратору своєї ланки повідомлення про закінчення роботи.
35. В особистій робочій папці створити папку «Мережні додатки», куди помістити ярлики запуску наступних програм: MS Word, MS Excel, MS Paint.
36. Створіть папку «Уведення висновку», куди внесіть у виді ярликів: мережний принтер, дисковод (іншої робочої станції користувача своєї ланки), робочі станції своєї ланки і папку, що знаходиться на сервері.
37. Створіть в меню Пуск/Програми свою групу програм для запуску «мережних додатків».

#### 4.4. Індивідуальні завдання №3

##### Засвоєння порядку формування адрес в Інтернеті

Розрахувати приклади IP-адреси, доменного ім'я та адреси електронної пошти, користуючись наведеною нижче таблицею 4.2, як основою для створення адреси.

Таблиця 4.2

№ п/п	1	2	3	4	5	6	7	8	9	10
Домен типу	com	edu	gov	mil	org	net	firm	store	web	arts
Домен країни	ua	ru	mo	gb	az	kaz	la	li	est	fr
Мережа	A	B	C	D	E	B	C	E	A	D
№ п/п	11	12	13	14	15	16	17	18	19	20

№ п/п	1	2	3	4	5	6	7	8	9	10
Домен типу	rec	info	nom	com	edu	gov	mil	org	net	firm
Домен країни	cha	au	ua	ru	mo	gb	az	kaz	la	li
Мережа	D	C	A	B	C	D	E	B	C	E
№ п/п	21	22	23	24	25	26	27	28	29	30
Домен типу	store	web	arts	rec	info	nom	com	edu	gov	mil
Домен країни	est	fr	cha	au	ua	ru	mo	gb	az	kaz
Мережа	A	D	D	C	A	B	C	D	E	B
№ п/п	31	32	33	34	35	36	37	38	39	40
Домен типу	org	net	firm	store	web	arts	rec	info	nom	com
Домен країни	la	li	est	fr	cha	au	ua	ru	mo	gb
Мережа	C	E	A	D	D	C	A	B	C	D

### Контрольні запитання

1. Що таке комп'ютерна мережа? Дайте визначення локальної комп'ютерної мережі.
2. Яке обладнання потрібне для функціонування локальної мережі?
3. Концентратори. Види та призначення.
4. Що таке топологія мережі? Види топологій.
5. Комбіновані топології мереж.
6. Поясніть значення термінів «клієнт» та «сервер».
7. Порівняти однорангові, комбіновані та мережі з виділеним сервером.
8. Перелічити необхідні умови для створення файлів на іншому ПК.
9. Який принтер називається мережним, а який локальним?
10. Як роздрукувати файл, використовуючи ярлик мережного принтера?
11. Internet, призначення, коротка характеристика.
12. Режими роботи мереж.
13. Класифікація мереж.
14. Протокол TCP/IP – визначення та призначення.
15. Адресація в Internet. IP- адреса. URL- адреса.

*У розділі подано систему електронних адрес та прийоми використання локальними та глобальними мережами.*

# 5. ЗАСТОСУВАННЯ ІНТЕРНЕТУ В ЕКОНОМІЦІ

*У розділі викладено прийоми застосування Інтернету при пошуку, зберіганню та розповсюдженню економічної інформації.*

## 5.1. Браузери Інтернету

Програма чи програмний пакет, який дозволяє переглядати веб-сторінки □ Інтернеті – це так званий браузер (to browse, англ. – переглядати, швидко перегортати сторінки). Тобто дослівно browser – переглядач, тобто програма для перегляду Web-сторінок.

В Інтернеті є проблеми з передачею кирилиці. Для її трансляції потрібен 8-бітний код, а більшість серверів у США настроєно на передачу тільки 7-бітного коду, зручного для передачі латиниці. До того ж, немає єдиного стандарту для передачі кирилиці. Після подорожей по численним серверам, текст стільки ж разів і перекодується. Оскільки майже всі сервери працюють під операційною системою UNIX, основною системою кодування кирилиці на них є KOI8U (U – для української чи R – російської мов).

Для комп'ютерів звичайних користувачів з операційною системою Windows, прийнята система кодування CP-1251. Для операційної системи SunOS прийняте кодування кирилиці ISO-8869-5. Ті ж, хто і досі працює під управлінням MS DOS, користуються кодуванням CP-866. Для комп'ютерів фірми Apple-Macintosh прийняте кодування в системі MacCyrillic. Але в більшості випадків, браузери потрібне перекодування роблять автоматично. Інколи, перекодування треба обирати самостійно.

Браузери під час інсталяції вимагають адресу і телефон провайдера, вашу адресу та TCP/IP-код, який повідомляє вам провайдер.

### 5.1.1. Internet Explorer

Найбільш поширений браузер. Однією з причин цього є те, що він входить в склад стандартних додатків операційної системи Windows. З розвитком операційної системи та технологій представлення даних вдосконалювався і браузер. Деякі сервісні можливості сторінок розраховані якраз на останні версії Internet Explorer.

Вікно програми (рис. 5.1) має головне меню та панель інструментів, що дублює деякі позиції цього меню.

Кнопки **Вперед** і **Назад** дозволяють переміщатися поміж Web-сторінками, які ми вже переглядали раніше. **Останов** призводить до зупинки передачі змісту веб-сторінки. **Обновить** застосовується, коли веб-сторінка має змінну частину. Натискання кнопки **Домой** відправляє на стартову сторінку браузера. **Поиск** – відкриває додаткове вікно, в якому можна задати параметри пошуку потрібного

сайту чи товару. Кнопка **Избранное** відкриває меню раніше обраних і відмічених нами сайтів. **Почта** вмикає програму Outlook для передачі листів.

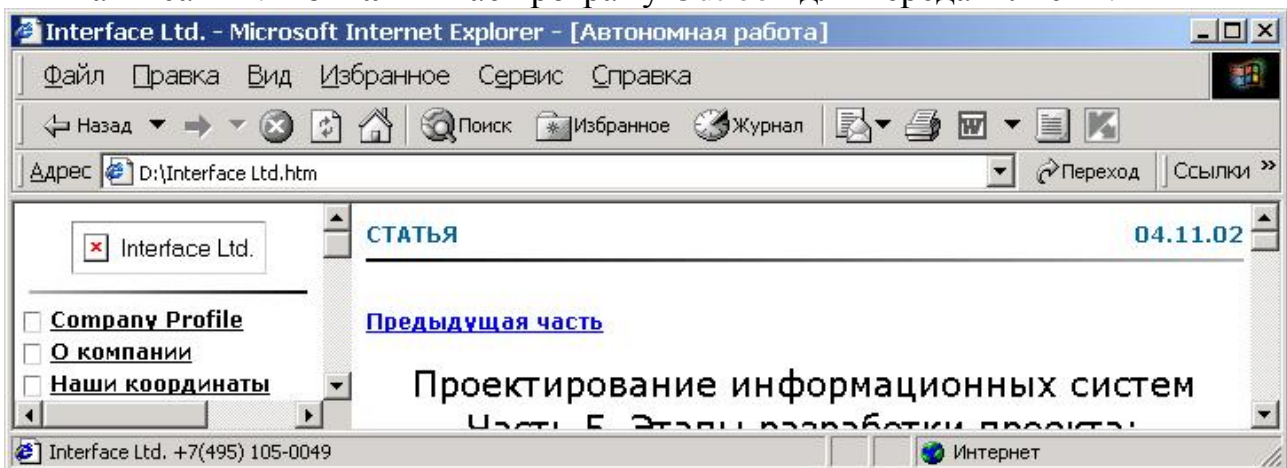


Рис. 5.1. Інтерфейс браузера Internet Explorer

Нижче панелі інструментів знаходиться адресний рядок. Він призначений для введення і відображення адреси Web-сторінки. Щоб перейти до якої-небудь сторінки вам навіть не потрібно набирати її адресу цілком, почніть друкувати, а засіб автозавершення доповнить передбачувану адресу, ґрунтуючись на адресах вузлів, які ви раніше відвідували.

Досить просто перетворити веб – сторінку в формат Word, Excel, Acrobat Reader чи Power Point, а також змінити кодування зображення таким чином, щоб текст веб–сторінки можна було прочитати.

### 5.1.2. Netscape Navigator

Програма поєднує в собі функції мейлера, браузера, чат-менеджера та редактора html-файлів, тому її роботу розберемо детальніше.

На початку роботи треба створити так званий профіль, тобто початкові настройки програми: записуємо власне ім'я, електронну адресу, а також найменування профілю. Окрім того, вказуємо адресу, де зберігається інформація відносно цього профілю. В наступному вікні треба вказати адресу сервера для пошти, що надходить, а також ваше ім'я для цієї пошти. Останній пункт меню настройки – це адреса сервера новин. Адреси всіх серверів повідомляє ваш провайдер.

Після ввімкнення браузера, він робить спробу зв'язатися з сервером вашого провайдера.

При перегляді веб-сторінки курсор на деяких місцях тексту перетворюється на вказівний палець. Це відбувається на ділянках тексту синього кольору з підкресленням де є гіперпосилання на іншу веб-сторінку. Як і в текстовому редакторі Word, натиснувши це місце лівою кнопкою миші, можна перейти на іншу сторінку. Використовується також термін *гіперлінк* (hyperlink – гіперпосилання).

Найбільш популярними інтерактивними сайтами є “пошукові машини” – сервери, які дозволяють знайти адресу потрібного вам сайту або електронну адресу конкретної фірми чи особи. Для цього треба вказати ключові слова: ім'я,

прізвище, тему пошуку (наприклад, слово акція, Шевченко, Україна тощо). Не треба забувати, що пошук може йти на різних мовах і в різних кодуваннях, але краще всього користуватися англійською.

Якщо якийсь сайт вам потрібен часто, можна зробити закладку по схемі, а потім звернутися до нього через перелік цих закладок. Закладки – це такі ж файли, як і інші у Windows, тому порядок роботи з переліком закладок тотожний з порядком роботи з файлами. В програмі наперед створені деякі папки з закладками, але ви можете створювати власні (рис. 5.2).

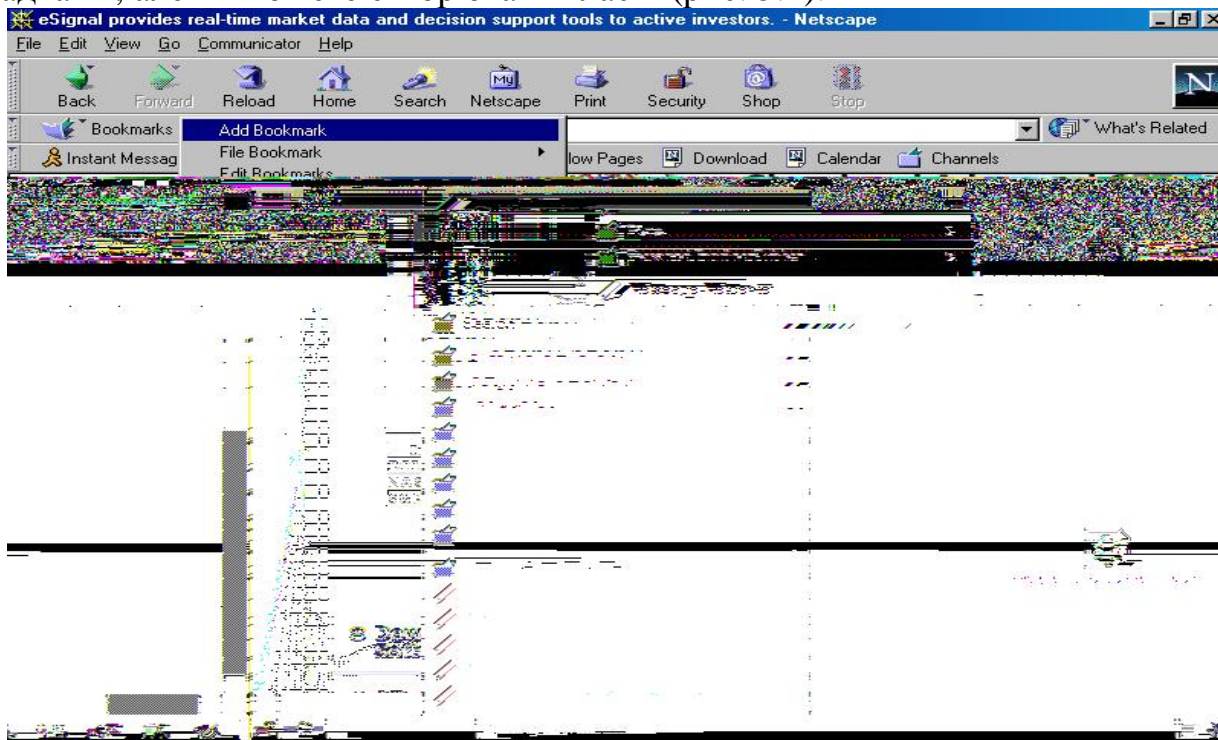


Рис. 5.2 Створення закладок

Окрім програми Navigator, Netscape має програму мейлер. Його теж спочатку треба настроїти. Перші три пункти стосуються цитованого тексту. Коли ви отримаєте листа і захочете відповісти, то мейлер змінить той текст, який ви отримали згідно настройок і вмістить у ваш лист відповідь. Водночас, він вкаже вашу адресу в полі FROM та адресу вашого дописувача в полі TO.

Наступні два пункти стосуються шрифту, яким ви будете отримувати листи та новини з телеконференцій. Третя група настройок стосується характеру листування в телеконференціях.

Після встановлення зв'язку з повайдером, можна “піти” на будь-яку сторінку в Інтернеті, вказавши її адресу і натиснувши **Enter**. Меню, часто зверху ліворуч чи праворуч, дозволяє швидко перейти на різні частини цієї сторінки.

Повернутися до попередньої сторінки можна за допомогою кнопок **Back** чи **Forward** на панелі інструментів. У випадку, якщо треба одразу перейти на якусь сторінку, можна застосувати пункт меню **Go**, де наведено повний перелік всіх сторінок, що були в перегляді і одразу перейти на потрібну.

### 5.1.3. Mozilla Firefox

#### 5.1.3.1. Інтерфейс

Програму розробляло близько 30 програмістів, а права на неї має Gecko™. Повний список авторів можна побачити в підпункті меню **Help-About Firefox - Credits**.

На рис. 5.3 наведено інтерфейс основного вікна програми.

Цифрами на ньому показано наступні основні елементи:

1 – Головне меню, яке включає в себе пункти: **File**, **Edit**, **View**, **Go**, **Bookmarks**, **Tools**, **Windows** та **Help**.

2 – Навігаційна панель інструментів має наступні кнопки (зліва направо): „Повернутися на попередню сторінку”, „Відмінити повернення”, „Повторити завантаження поточної сторінки”, „Припинити завантаження сторінки”, „Перейти на домашню сторінку”, смуга, в якій треба записати URL- адресу сторінки, кнопка виконання вказаного переходу, смуга пошукової машини Google.

3 – Панель вибраних сторінок, яка включає в себе кнопки „Зробити стартовою сторінку, яку ви зараз переглядаєте” та „Наступна головна сторінка”.

4 – Додаткове вікно, в якому можна побачити перелік папок по вибраним сторінкам або історію перегляду сторінок.

5 – Основне вікно, де зображається зміст Web-сторінок.

6 – Поле додаткової інформації.



Рис. 5.3. Інтерфейс основного вікна браузера Mozilla Firefox

#### 5.1.3.2. Основні прийоми роботи з браузером Mozilla Firefox

##### Переміщення на іншу сторінку

Ви можете перейти на нову веб-сторінку, набираючи її URL-адресу в смузі адрес. URL частіше всього починаються з "http://" з одним або більше іменами,



які ідентифікують адресу. Наприклад – <http://www.mozilla.org/>. Якщо ви раніше набирали цю адресу, то достатньо клацнути смугу, щоб вибрати потрібний URL. В іншому випадку наберіть URL сторінки, яку ви хочете відвідати. URL, який ви набираєте, замінює той текст, що був перед цим у смугі. Натисніть *Enter (Return)*. Щоб швидко добратися до смуги, натисніть CtrlCmd+L.

Якщо ви не знаєте URL, то наберіть ключові слова (як наприклад "подарунки" або "квітки") у полі пошуку і натисніть *Enter (Return)*. Це запустить пошук сторінки, тема якої містить вказані вами ключові слова.

### **Перехід по гіперпосиланням**

Більшість веб-сторінок містить посилання, що полегшують переміщення на інші сторінки.

1. Перемістіть курсор миші, поки він не зміниться на зображення вказівного пальця. Це трапляється кожного разу, коли курсор опиняється над посиланням. Більшість посилань мають вигляд підкресленого тексту, кнопки і малюнки також можуть бути посиланнями.
2. Клацніть на посиланні один раз. Процес переходу за гіперпосиланням відображається внизу основного вікна.

### **Повторний перегляд сторінок**

Є декілька шляхів повторного перегляду сторінки:

- щоб перейти назад або вперед одну сторінку, клацніть кнопку *Back* або *Forward*;
- для переміщення назад або вперед більш ніж на одну сторінку, клацніть маленькі трикутники на кнопках *Back* і *Forward* та зі списку недавно відвіданих сторінок виберіть потрібну;
- щоб побачити список URL, які Ви вже набирали в смугі адрес, клацніть стрілку в правому кінці смуги та виберіть потрібну зі списку;
- для вибору зі сторінок, які ви відвідали протягом поточної сесії, відкрийте меню *Go* і використайте список в нижній секції меню.
- для вибору зі сторінок, які ви відвідали протягом минулих декількох сесій, відкрийте меню *Go* і виберіть Історію. Ви бачитимете Історію *Sidebar*, що відображає список папок. Клацніть двічі, щоб побачити список папок нижнього рівня або закладок до веб-сторінок. Ви можете клацнути двічі URL поряд з іконою Закладки, щоб перейти на ту сторінку.

### **Використання Sidebar**

*Sidebar* – це область з лівого боку екрану, яку можна використовувати для перегляду закладок або історії.

Щоб ознайомитися з елементами в *Sidebar*, виберіть *View -> Sidebar*. Звідти ви можете вибрати потрібний пункт підменю.

### **Зупинка і перевантаження**

Якщо сторінка завантажується дуже поволі або ви відмовляєтесь від подальшого перегляду, клацніть кнопку *Stop*.

Щоб перезавантажити поточну сторінку або одержати її сучаснішу версію, клацніть кнопку *Reload* або натисніть CtrlCmd+Shift+R.

## **Фіксація переглянутих сторінок**

Коли ви відвідуєте більш ніж одну веб-сторінку за один раз, ви можете використати фіксацію, щоб знайти Web швидше і легше.

Перегляд фіксації дозволяє вам відкрити як таблиці, кожне відображення веб-сторінки, в межах єдиного вікна Firefox. Вам не доведеться відкривати декілька вікон, щоб відвідати різні веб-сторінки. Це звільняє простір на вашому робочому столі. Ви можете відкрити і перенавантажувати веб-сторінки в одному місці без необхідності переключатися в інше вікно.

## **Пошук Web-сторінок**

Пошук веб-сторінок за потрібною вам темою виконується набором декількох слів у смугу пошуку в наступному порядку:

1. Клацніть в Пошуковій Смугі.
2. Наберіть маленьку фразу. Ваш набір замінює будь-який текст у пошуковій смугі.
3. Натисніть *Enter (Return)* для пошуку.

Результати пошуку з'являються у вікні Firefox.

## **Пошук за вибраними словами в різних Веб-сторінках**

Firefox дозволяє шукати визначені слова в різних веб-сторінках:

1. Виберіть (відмітьте) які-небудь слова у веб-сторінці.
2. Утримуючи ліву кнопку *Ctrl*, клацніть клавішу миші, і виберіть пункт меню "*Шукати у Web-сторінці [ваши відібрані слова]*".

Firefox відкриває нове вікно і згідно ваших умов, шукає вказані слова на інших веб-сторінках.

## **Пошук в межах сторінки**

Щоб знайти текст в межах сторінки, яку ви зараз переглядаєте в Firefox:

1. Виберіть *Редагування > Пошук в цій сторінці*. Якщо поточна сторінка містить фрейми, вам, можливо, потрібно клацнути в межах фрейму перед початком пошуку.
2. Внизу вашого навігатора повинна з'явитися нова панель інструментів.
3. Наберіть текст, який ви хочете знайти.
4. Виберіть одну з наступних пошукових альтернатив:
  - пошук в сторінці, що розміщується нижче поточної координати курсору;
  - пошук в сторінці, вище поточної координати курсору.
5. Виділенням програма покаже результат пошуку в поточній сторінці. На випадок знайдення схожого слова, обмежте пошук текстом, який має те ж написання, що і ваш пошуковий рядок.
6. Пошук автоматично починається, коли що-небудь набране в текстовий блок.

Щоб знайти таке ж слово або фразу знову, виберіть *Редагування > Пошук Знову*.

## **Копіювання частини сторінки**

Ви можете вставити скопійований текст в інші програми. Щоб копіювати текст на сторінці:

1. Виділіть текст.
2. Виберіть *Редагування > Копія від Рядка Меню*.

Щоб копіювати посилання (URL) або посилання-зображення на сторінці:

1. Помістіть покажчик над посиланням або зображенням.
2. Утримуючи ліву кнопку *Ctrl*, клацніть на посиланні або зображенні, щоб отримати спливаюче меню.
3. Виберіть *Копіювання посилання* чи *копіювання малюнку*. Якщо малюнок – це також посилання, ви можете вибрати будь-який пункт меню.

Ви можете вставити посилання в інші програми або в смугу адресації Firefox.

### **Збереження частини або цілої сторінки**

Збереження файлу на ваш жорсткий диск дозволяє вам користуватись сторінкою, коли ви не сполучені з Інтернетом.

Щоб зберегти всю сторінку в Firefox:

1. Виберіть *Файл > Сторінка > Зберегти як*.
2. Виберіть диск та папку для сторінки, що зберігається.
3. Виберіть формат для сторінки, яку ви хочете зберегти:
  - Веб-сторінка з малюнками, Збережіть цілу веб-сторінку разом із зображеннями. Цей вибір дозволяє вам переглядати її спочатку із зображеннями, але, можливо, не збереже структуру посилання HTML оригінальної сторінки. Firefox створює нові папки, де зберігається сторінка, а також, малюнки і інші файли, необхідні, щоб показати цілу веб-сторінку.
  - Веб-сторінка, Тільки HTML збереже оригінальну сторінку без малюнків.
  - Текстовий файл: Збереже оригінальну сторінку як текстовий файл. Цей вибір не зберігатиме оригінальну структуру посилання HTML, але дозволить вам бачити текстову версію веб-сторінки в якому-небудь текстовому редакторі.

4. Наберіть ім'я файлу для сторінки і клацніть *Збереження*.

Щоб зберегти фрейм зсередини веб-сторінки:

1. Помістіть покажчик миші в межах фрейма.
2. Натисніть *Ctrl* і клацніть один раз мишкою, для появи спливаючого меню.
3. Виберіть *Цей Фрейм > Фрейм Збереження Як* в підменю нижнього рівня.
4. Відкриється діалогове вікно *Збереження Як*, виберіть розташування для сторінки, що зберігається.
5. Вкажіть формат для сторінки, яку ви хочете зберегти.
6. Наберіть ім'я файлу для сторінки і клацніть *Збереження*.

Щоб зберегти малюнок зі сторінки:

1. Помістіть покажчик миші над зображенням.
2. Утримуючи ліву кнопку *Ctrl*, клацніть на малюнку, щоб з'явилося спливаюче меню.
3. Виберіть *Збереження малюнку Як*.

4. У вікні *Збереження малюнку* вкажіть диск та папку для зображення.
5. Наберіть ім'я файлу для малюнку або скористайтеся запропонованим і клацніть *Збереження*.

Щоб встановити зображення як шпалери робочого столу, клацніть праву кнопку по зображенню і виберіть зі спливаючого меню підпункт *Як Шпалери*.

### **Друк Сторінки**

Щоб надрукувати поточну сторінку:

- Виберіть *Файл > Друк*.

Щоб надрукувати відібраний текст:

- Виберіть текст в поточній сторінці.
- Виберіть *Файл > Друк*.
- В діалоговому вікні друку задати *Діапазон Друку*.

## **5.1.4. Opera**

### **5.1.4.1. Інтерфейс браузера Opera**

Автор програми David M. Gay. Авторські права належать Opera Software ASA.

Інтерфейс основного вікна браузера наведено на рис. 5.4. Позначки на ньому мають наступні значення:

- 1 – Головне меню браузера, яке включає в себе такі позиції як File, Edit, View, Navigation, Bookmarks, Chat, Tools, Windows та Help.
- 2 – Панель інструментів, кнопки якої повторюють часто вживані пункти головного меню.
- 3 – Пошукова панель
- 4 – Панель користувача.
- 5 – Панель навігації, де є вікно із записом URL-адреси веб-сторінки та вікна інших пошукових машин, відмінних від Amazon.
- 6 – Додаткове вікно режимів роботи браузера.
- 7 – Основне вікно, де зображається знайдена веб-сторінка.

### **5.1.4.2. Ключові особливості браузера**

#### **Фіксація блоків, які переглядаються і спливають**

Саме у браузері Opera вперше було застосовано можливість блокування спливаючих вікон. Opera дозволяє вам управляти процесом появи тільки тих об'єктів, які ви бажаєте бачити. Окрім того, вперше було використано прийом так званої „табуляції”, коли ви можете відкрити багато сторінок в межах одного вікна.

#### **Система об'єднаного пошуку**

Знайдіть свої улюблені пошукові машини, наприклад Google, eBay, чи Amazon без необхідності йти до їх веб-сторінок. Використовуйте об'єднане пошукове вікно або шукайте безпосередньо в ярликах використання поля ад-

реси (наприклад, запис "Opera g" в полі адреси знайде Opera, використовуючи Google).

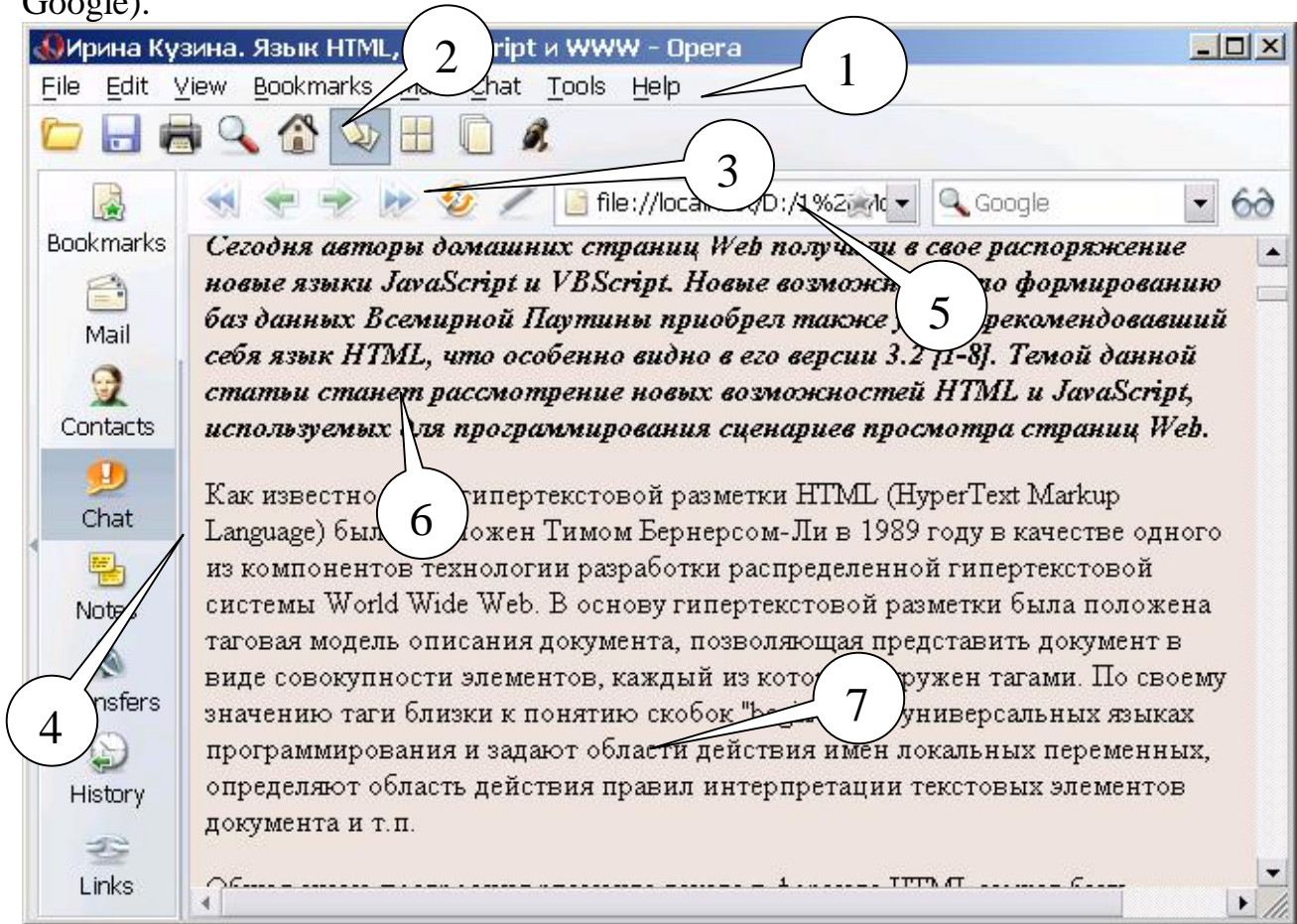


Рис. 5.4. Інтерфейс основного вікна браузера Опера

### **Безпека**

Система безпеки може захистити вас від вірусів та інших шпигунських додатків, які мовчки нападають на ваш комп'ютер, під час навігації в Інтернеті. Наприклад, Opera відображає інформацію безпеки усередині поля адреси, розміщеного поряд з іконкою висячого замка, яка указує рівень наявності безпеки на вузлі. Opera також забезпечує захист проти атак хакерів і автоматично перевіряє на модифікації безпеки.

### **Швидкість**

Opera є найшвидшим браузером з існуючих на даний час. Розробники постійно працюють, щоб ще більш прискорити процес завантаження веб-сторінок від провайдерів на ваш комп'ютер.

### **Менеджер пароля**

Менеджер пароля пам'ятає імена користувача і паролі. Менеджер пароля може запам'ятати більш ніж одне ім'я користувача і комбінацію пароля для кожної сторінки, і дозволить вам вибирати якийсь один для використання, якщо у пам'ять занесено декілька комбінацій.

### **Настройка**

Використовуючи діалог „появи”, існує можливість переглядати сторінки будь-яким шляхом, переміщати кнопки на панелі інструментів, додавати і вида-

ляти панелі інструментів, і так далі. Режим Opera skins може змінити вигляд навігатора, надаючи йому іконки і кнопки за вашим вибором.

### **Регулювання ширини вікна**

Мається на увазі зручніша горизонтальна прокрутка зображення веб-сторінки і поліпшену здатність друкувати повні веб-сторінки. У комбінації з функцією *Zoom*, ви можете міняти розмір веб-сторінки так, щоб розглядати їх без необхідності прокрутки в сторону.

### **Голос**

Opera – перший браузер, готовий до майбутнього веб-вузлів, що пропонують взаємодію голосом при купівлі або замовленні послуг. Існує можливість перегляду мовних команд для використання тим чи іншим Web-вузлом, як наприклад "Send Opera", "Opera back", або "Opera speak". Остання команда забезпечить можливість читання змісту веб-сторінки і повідомлення електронної пошти в голос. Режим Voice зараз пропонується тільки англійською. Щоб використовувати функцію голосового управління браузером, потрібні навушники з мікрофоном.

### **Розробка web-сторінок**

Вбудований редактор веб-сторінок дозволяє створювати їх і одразу переглядати отримані результати.

### **Додаткові особливості**

- Повна підтримка поштової служби пошукового сервера Google – Gmail.
- Підтримка сервера новини RSS/Atom newsfeeds.
- Перший браузер, який підтримує векторні графіки, формату \*.SVG.

## **5.2. Мейлери Інтернету**

Електронна пошта слугує прямому обміну листами поміж користувачами в мережі, замінюючи пошту, телеграф чи телефон. *Електронна пошта*, або *E-mail*, базується на трьох протоколах. SMTP, POP3 та IMAP.

Для того, щоб мати можливість обмінюватися листами по електронній пошті, користувач повинен стати клієнтом однієї з комп'ютерних мереж – підрозділів Internet, доступ до яких здійснюється через постачальників мережних послуг – провайдерів. Для кожного такого користувача на одному з мережних комп'ютерів виділяється область пам'яті – електронна поштова скринька. Обмін повідомленнями в Internet відбувається за допомогою поштових серверів. Електронний лист складається з тексту та заголовку. Обов'язкова частина листа нагадує адресу на конверті і має такі обов'язкові поля:

**TO:** (кому) – електронна адреса одержувача листа, наприклад ekit@ngau.ua.

**FROM:** (від кого) електронна адреса відправника листа, наприклад ekit@nmu.org.ua. Вставляється автоматично мейлером – програмою, що відправляє листи електронною поштою.

**CC:** (carbon copy) електронні адреси тих, кому направляються копії вашого листа. Адреси записуються через кому або через пропуск.

**BC:** (blind copy) електронні адреси тих, кому направляються копії вашого листа, але основний одержувач листа, на відміну від CC не знатиме, що копії відправлені ще комусь.

**DATE:** (дата) – вставляється автоматично.

**SUBJECT:** (або SUBJ) – тема вашого листа. Інколи, щоб не писати в тексті вдруге цю тему, пишуть просто subj. Наприклад, якщо темою листа є купівля комп'ютера, то пишуть : “Куплю subj, не гірше за пентіум-2”.

Мейлер сам автоматично додає ще таку інформацію, як маршрут листа, тип кодування листа 7-ми або 8-ми бітний код та тип мейлера, яким ви користуєтеся.

Текст часто містить спеціальні значки – смайли (від англ., smile – посмішка), вони надають емоційного забарвлення вашому тексту, мають вигляд смішних облич, якщо розглядати їх, повернувши на кут у 90 градусів за годинниковою стрілкою, а також типових скорочень, які дозволяють замість часто вживаних словесних зворотів вживати аббревіатуру. У таблицях 5.1 та 5.2 наведено деякі з цих елементів електронного листування. Типові скорочення частіше всього вживаються в англомовних листах.

Таблиця 5.1

Часто вживані смайли

:-) Ваша основна посмішка.	;-) Посмішка с підморгуванням.
:-( Похмура фізіономія.	:-I Індиферентна фізіономія.
:-7 Крива посмішка.	:-* Похмурий.
:-P або :-b Той, що висуває язика.	:-S Непослідовний
:-/ Скептик.	:-o Ох-ох.
:-] Дурень.	:-[ Не усміхнений дурень.
:-\ Нерішучий.	8-) З сонячними окулярами.
В-) З окулярами.	В:-) Окуляри підняті на голову.
:-{} Вусатий.	g-) З пенсне (моноклем).
8:-) Маленька дівчинка.	:-)-8 Доросла дівчинка.

Таблиця 5.2

Основні скорочення у текстах листів

Скорочення	Англійська фраза	Зміст
AAMOF	As A Matter Of Fact	Як факт,...(Насправді..)
AFAIK	As Far As I Know	Наскільки мені відомо
AS	On Another Subject	З іншого питання (бесіди)
BBS	Bulletin Board System	ББС (BBS) – сервер новин
BTW	By The Way	Між іншим (до речі)
CU	See You	Побачимось
CUL/CUL8R	See You Later	Побачимось пізніше
EMFBI	Excuse me for butting in	Вибачте, що втручаюся

Скорочення	Англійська фраза	Зміст
FYI	For Your Information	До вашого відома
IC	I See	Зрозуміло (Я зрозумів)
IMHO	In My Humble/Honest Opinion	На мою скромну думку
OTOH	On The Other Hand	З іншого боку...
PMJI	Pardon my jumping in	Вибачте, що втручаюся
POV	Point Of View	Точка зору
RTFM	Read The Fine Manual	Читай документацію
SYSOP	System Operator	Системний оператор

### 5.2.1. Програма Outlook Express

Програма Outlook Express створена на основі стандартів, використовуваних в Інтернеті, і призначена для роботи з електронною поштою і новинами.

Програму Outlook Express можна використовувати для читання груп новин чи груп обговорень, таких як Usenet. Робота з групами новин здійснюється через сервери новин NNTP.

Програма Outlook Express містить у собі адресну книгу Windows. Програма надає широкі можливості керування контактними даними, включаючи створення груп контактів і папок для сортування повідомлень і розміщення адрес електронної пошти.

#### 5.2.1.1. Інтерфейс Outlook Express

Outlook Express має стандартний для додатків Windows вигляд (рис. 5.5). Основне меню містить команди, що надають доступ до усіх функцій, необхідних для роботи з Outlook Express. *Панель інструментів* призначена для швидкого доступу до деяких найчастіше використовуваних команд Outlook Express.



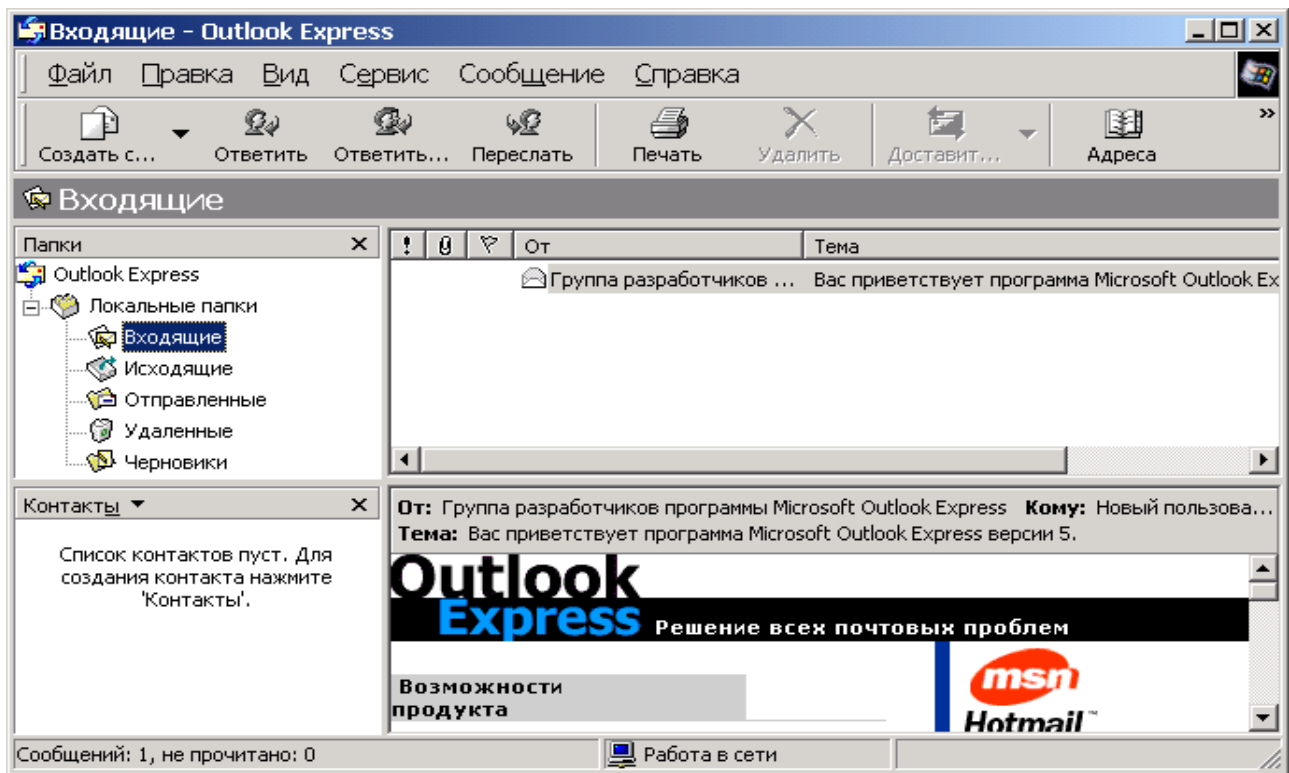


Рис. 5.5. Интерфейс програми Outlook Express

Після завантаження, Outlook Express встановлюється в загальний режим (коли ще не обраний потрібний режим – пошта чи новини). У цьому режимі кнопки панелі інструментів виконують наступні функції:

- *Создать сообщение* відкриває вікно для формування нового листа.
- *Доставить почту* – ви можете швидко і легко підключитися до провайдера, щоб перевірити і доставити вхідну пошту, а також відправити власну.
- *Адреса* – відкриває доступ до адресної книги, в яку для зручності записують E-mail адреси ваших друзів, колег по роботі і т.д.
- *Соединить* – натискання цієї кнопки активізує процес з'єднання з провайдером.
- *Разорвать соединение* – розриває з'єднання з провайдером послуг Internet.

Панель **Локальные Папки** дозволяє вивести на екран списки листів, що зберігаються в одній зі стандартних папок Outlook Express: **Входящие, Исходящие, Отправленные, Удаленные і Черновики**. Outlook Express дозволяє створити нові додаткові папки користувача, і вони також будуть доступні з цього рядка.

*Область перегляду* Outlook Express при роботі з електронною поштою чи новинами розділена на дві частини: у верхній частині ви бачите список повідомлень електронної пошти з поточної папки, а в нижній частині показується зміст виділеного листа. Зміст листа можна подивитися і в окремому вікні, якщо двічі клацнути лівою кнопкою миші на рядку з обраним листом. Область перегляду може бути розділена по горизонталі чи по вертикалі — якщо такий спосіб організації інтерфейсу здається вам більш зручним. За замовчуванням, усі листи в папках розташовуються відповідно до алфавітного порядку імен відправників,

але їх можна відсортувати по іншим характеристикам, наприклад, по даті надходження листа.

*Рядок стану* служить для двох цілей. Звичайно в ньому Outlook Express указує загальне число повідомлень у даній папці й окремо — число непрочитаних повідомлень. У правій частині рядка стану при перевірці надходження нової пошти з'являється напис, що інформує про прибуття чи відсутність нових листів. Крім того, при роботі з Outlook Express у рядку стану з'являються значки, що характеризують режим роботи цього додатка в даний момент часу (наприклад, закреслений значок мережного диска означає, що в цей момент немає з'єднання з Internet).

### **Папки Outlook Express**

**Входящие.** Сюди за замовчуванням надходить уся нова пошта і зберігаються всі повідомлення, що прийшли.

**Исходящие.** Призначена для тимчасового збереження листів, що відправляються. Навіщо це потрібно? Уявіть, що ви готуєте кілька листів один за одним. Щоб не з'єднуватися щоразу з Internet для відправлення чергового листа, їх тимчасово накопичують в цій папці. Потім при натисканні кнопки *Доставить почту* вони разом відправляються на поштовий сервер провайдера і далі – до своїх адресатів.

**Отправленные.** Тут за замовчуванням зберігаються копії відправлених повідомлень, і ви завжди зможете подивитись, що, кому і коли ви відсиляли.

**Удаленные.** Якщо ви вирішите видалити непотрібні повідомлення, то вони тимчасово містяться в цій папці (на випадок, якщо ви передумаєте їх видалити). Якщо ви вирішите видалити повідомлення з цієї папки назавжди, натисніть правою кнопкою миші по значку папки й у контекстovому меню виберіть пункт **Очистить папку**.

**Черновики.** Якщо ви готуєте новий лист, але тимчасово відкладаєте на потім, виберіть у меню *Файл* пункт *Сохранить*. Такий "недописаний" лист зберігається в папці **Черновики**. Щоб згодом продовжити роботу над листом □, просто відкрийте цю папку і подвійним щиглем на чернетці листа розкрийте його. Потім, якщо лист готовий, його можна відправити, і він буде вміщений в папку **Исходящие**. Якщо ж лист, як і раніше, не готовий до відправлення, його знову можна зберегти в папці чернеток.

При першому відкритті Outlook Express, відкривається вікно з попередженням, що ця програма – не поштова програма, яка використовується за замовчуванням, та з пропозицією зробити її такою. Тому необхідно натиснути на кнопку *Да*, а у вікні, що відкрилося, вибрати режим роботи з електронною поштою.

Потім, щоб користуватися поштою, треба створити свій обліковий запис. Для цього за допомогою пункту меню *Сервис/Учетные записи...* відкриваємо вікно, в якому у закладці *Почта* вибираємо кнопку *Добавить*. Далі вибираємо пункт *Почта*. Після цього завантажується майстер підключення до Internet. У першому вікні майстра вказуємо ім'я відправника. В другому – пишемо адресу електронної пошти. В третьому – встановлюємо характеристики поштового сер-

веру. У полі *Сервер входящих сообщений* установіть протокол POP3, у полі *Сервер входящих сообщений (POP3, IMAP, HTTP)* введіть IP-адресу серверу, в полі *Сервер исходящих сообщений (SMTP)* введіть IP-адресу. В четвертому вікні у полі *Имя для входа* введіть своє ім'я та пароль, зніміть прапорець з поля *Запоминать пароль*. Відкриється останнє вікно, в якому натисніть кнопку *Готово*. Відбудеться звернення у вікно *Учетные записи в Internet*. Створений обліковий запис потрапляє до списку записів. Виділіть його і натисніть на кнопку *Используйте по умолчанию*.

**Адресна книга** — це збірник E-mail адрес ваших колег по електронному переписуванню, організований за допомогою зручної програмної оболонки. Для заповнення адресної книги необхідно виконати команду *Сервис/Адресная книга*. У вікні, що відкрилось, виконати команду *Файл/Создать контакт* чи кнопку на панелі інструментів *Создать/Создать контакт*. Вкладку *Имя* треба заповнити обов'язково, а інші параметри вводити необов'язково.

Заповнювати адресну книгу Outlook Express можна в двома способами: поперше, витратити спочатку якийсь час і заздалегідь увести дані про ваших колег, і, по-друге, поповнювати адресну книгу "на льоту", по мірі роботи з поштою, копіюючи адреси листів, що надійшли.

Крім імені й E-mail адреси, в адресній книзі можна зберігати безліч різної інформації – номери телефонів, пейджерів, особистих і службових сторінок Web, звичайну поштову адресу абонента та ін.

Для створення повідомлення треба виконати команду *Сообщение/Создать сообщение*. У вікні, що відкрилось в полі *Кому* введіть ім'я адресата чи виберіть його з адресної книги, що зручніше і швидше, для цього клацніть на кнопці *Кому*. Відкриється вікно *Выбрать получателей*, в якому можна вибрати одержувача зі списку, наприклад, подвійним щигликом, після чого ім'я одержувача автоматично з'являється у вікні *Создать сообщение*.

Аналогічним чином у полі *Копия* можна задавати імена адресатів, яким ще слід надіслати це повідомлення. У полі *Тема* можна написати короткий опис повідомлення. Потім набрати текст повідомлення та відправити його за допомогою команди *Файл/Отправить* або кнопки на панелі інструментів (ПІ) *Отправить*.

В Outlook Express є також можливість відформатувати текст повідомлення за допомогою команди *Формат*. Форматування відбувається такими ж прийомами, як і у Word.

У програмі є можливість створити повідомлення на якомусь бланку, для цього треба натиснути кнопку на ПІ *Создать* і в меню, що відкриється, вибрати будь-який шаблон, наприклад, *Приглашение на праздник*.

Для прийому повідомлень треба натиснути кнопку *Доставить*. У меню, що відкриється, виберіть команду *Доставить почту*. У вікні *Вход – имя пользователя*, введіть ім'я користувача та пароль.

Отримане повідомлення вміщується в папку **Входящие**, число в дужках, що знаходиться поруч з нею, показує скільки є нових повідомлень. Нове пові-

домлення буде виділятися жирним шрифтом, крім того, відображається зміст поля *Тема* та дата надісланого повідомлення.

Для знищення повідомлень треба виділити його назву та натиснути кнопку *Удалить*.

Для того, щоб відповісти на повідомлення, треба виконати команду *Сообщение/Ответить* чи кнопку *Ответить* на ПІ.

Відкриється вікно аналогічне вікну створення повідомлення. В поле *Тема* програма автоматично копіює текст з однойменного поля вихідного повідомлення та надає йому префікс *Re*, що означає відповідь. Набираємо текст відповіді та відправляємо повідомлення за допомогою команди *Файл/Отправить* чи кнопки на ПІ *Отправить*.

В результаті в папці *Отправленные* з'явиться повідомлення з відповіддю.

Для того, щоб приєднати файл до повідомлення треба у вікні *Создать сообщение* виконати команду *Вставка/Вложение из файла* або натиснути на кнопку на ПІ *Вставить*. У вікні, що відкрилось вибрати необхідний файл. Після цього у вікні *Создать сообщение* в полі *Присоединить* з'явиться ім'я файлу.

Згодом, коли в різних папках накопичиться багато листів, є сенс подумати про таку можливість організації збереження вашої пошти, як *стиск листів* (створення архіву), що сприяє економії ресурсів вінчестера.

Для того щоб стиснути повідомлення в конкретній папці, відзначте потрібну папку і виберіть у меню команди *Файл/Папка/ Сжать*.

Крім того, можна задати опцію стиску всіх папок, для чого в зазначеному меню виберіть пункт *Сжать все папки*.

### 5.2.2. Програма **The Bat**

Ще одним цікавим мейлером є програма **The Bat**. Як і Outlook, вона забезпечує листування з іншими абонентами, а також можливість підписатися на сервери новин і конференцій. Отже ця програма повністю працює за схемою мейлера та редактора "off-line"-мереж. Структура вікна (рис. 5.6) такої програми типова.

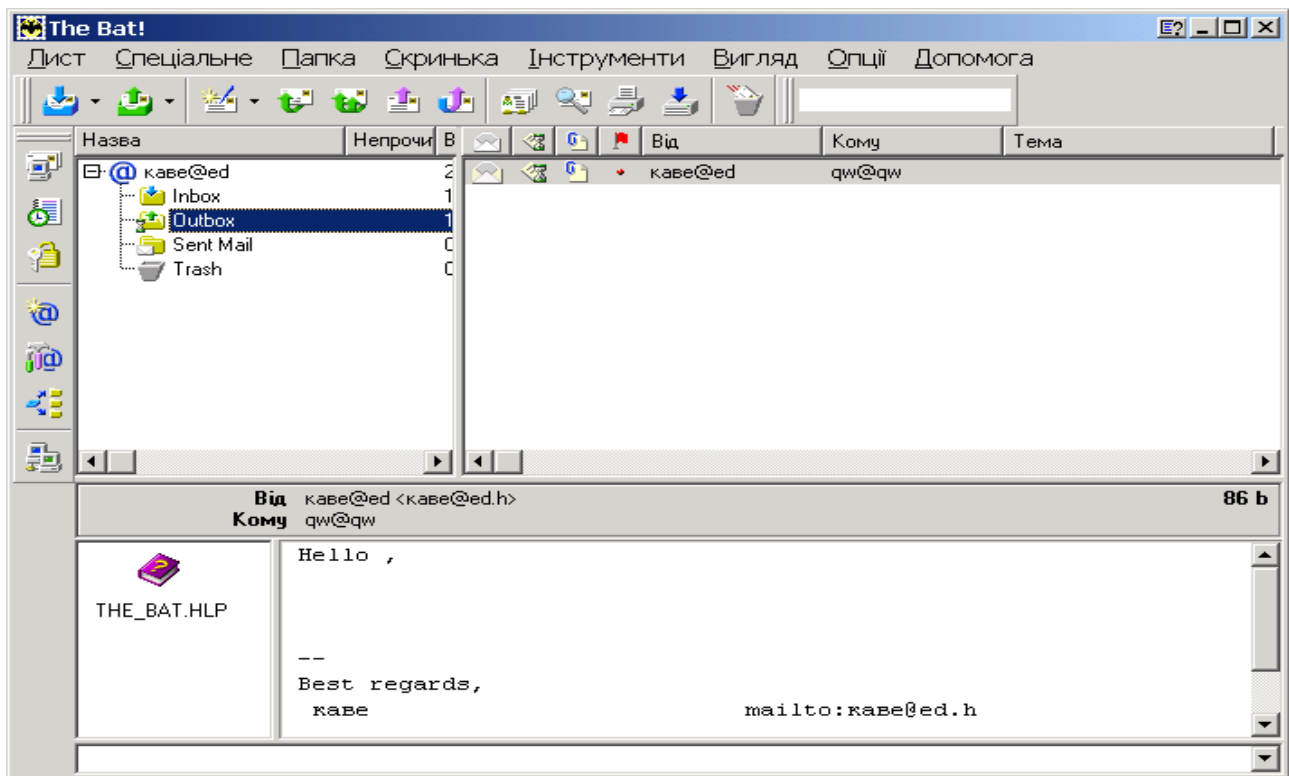


Рис. 5.6. Зовнішній вигляд основного вікна мейлера The Bat.

В лівій частині наведено перелік папок для листів, новин і конференцій, а в правій – перелік листів, які надійшли чи вийшли з тієї чи іншої папки.

Цей перелік можна відсортувати або відфільтрувати. Останнє означає, що частина листів буде невидимою для нас згідно вибраної ознаки. Наприклад, якщо в полі SUBJ вказана тема “Про мейлери” або в полі NAME – “Petroff” і т. ін.

Програма надає стандартні сервіси мейлера по формуванні та відправці повідомлень). Досить повна і структурована адресна книга з можливістю ідентифікації цифрових підписів адресатів.

**The Bat** фактично старійшина серед інших мейлерів, коли він працював, Outlook можливо і в проекті не було. Звичайно, з часом, текстовий інтерфейс змінився на більш зручний і звичний.

Це додаток до браузера, який дозволяє користатися можливостями електронної пошти. Для цього потрібно вибрати пункт меню *Tools -> Mail and Chat Accounts*. При першому ввімкненні програми, з’явиться майстер налаштування електронних адрес. Після заповнення запропонованої анкети, з’явиться можливість користуватися браузером як мейлером, через той же пункт меню.

Коли вмикається майстер налаштувань, вам пропонується вибрати з деякого переліку запитань і надати на них відповіді. Коли обсяг відповідей стає достатнім, кнопка „Next” стає активною. Натиснувши на неї, отримуємо нові запитання. Про закінчення налаштування повідомляється.

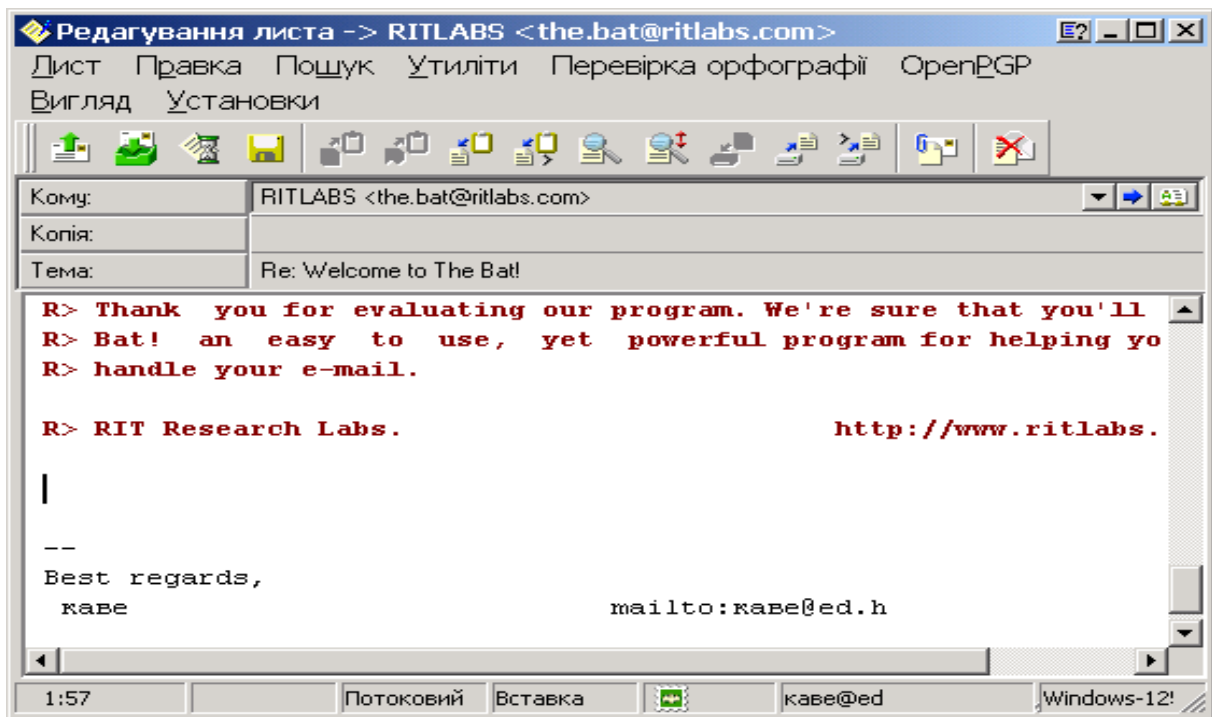


Рис. 5.7. Вікно відповіді на лист з цитованими рядками тексту та стандартним початком і закінченням листа

### 5.2.3. Поштовий агент браузера Опера

Спочатку запитується, які з додаткових функцій програми треба встановити. Потрібно обрати пункт „Import E-Mail”.

Далі потрібно вибрати пункт “Import from Opera”, бо якщо вибрати ІМАР, програма одразу пропонує створити листа. Вказуємо пароль, логін, ІР-адресу провайдера і отримуємо підтвердження, що ваша електронна адреса створена (рис. 5.8) у вигляді вікна, в якому до списку вже існуючих адрес додається і ваша. Тепер можна писати листи.

## 5.3. Можливості Інтернету

### 5.3.1. Віртуальна електронна пошта

Електронна адреса, яку ми отримуємо від провайдера і записуємо в налаштуваннях програми-мейлера, прив’язана до конкретного комп’ютера, на якому вона встановлена. Якщо виникає потреба у постійному переміщенні від одного комп’ютера до іншого, у нагоді стають віртуальні електронні адреси, послуги яких надають все більше і більше сайтів. Безкоштовна адреса електронної пошти, звичайно, не буде цілком дармовою – ви все одно платите провайдеру за доступ у Internet. Сьогодні в Internet існують сотні добре відомих і надійних служб безкоштовної електронної пошти – Hotmail ([www.hotmail.com](http://www.hotmail.com)), AltaVista ([altavista.digital.com](http://altavista.digital.com)), BigFoot ([www.bigfoot.com](http://www.bigfoot.com)), [www.freemail.ukr.net](http://www.freemail.ukr.net), [www.mail.ru](http://www.mail.ru), [www.gmail.google.com](http://www.gmail.google.com) і т.д.

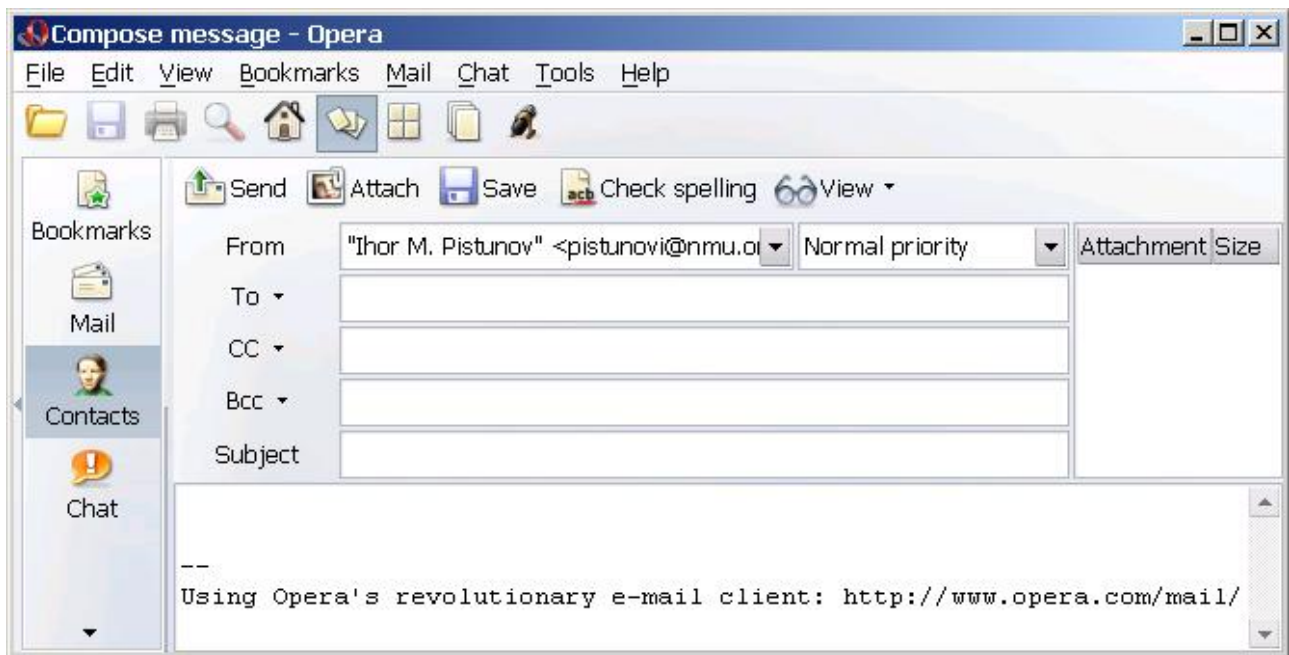


Рис 5.8. Робота браузера Opera в режимі поштового агента

При виборі поштового сервера слід звернути увагу на наступні фактори: розмір дискового простору, що надається під скриньку, загальний інтерфейс і можливості його налаштування, різноманітні сервісні можливості, як то автовідповідач, чорні списки, служба антиспаму та вірусного захисту.

Значним достоїнством електронної пошти безперечно являється швидкість передачі, основним недоліком – спам, тобто різноманітні рекламні розсилки.

Суттєва відмінність цих служб у використанні поштової скриньки, що зберігається на поштовому сервері, який надає такі послуги, а доступ до неї відбувається за допомогою браузера.

### 5.3.1.1. Створення поштової скриньки

Відкривши за допомогою браузера вікно сайту, на якому пропонується створення віртуальної адреси, ми бачимо, що там потрібно ввести логін та пароль. Поруч з ними має бути слово „Зареєструватися”, „Нова адреса” чи щось подібне.

Сторінка, що відкриється містить угоду, що обумовлює умови користування сервісом, права та обов'язки сторін і т.і. Далі користувач сам формує свій обліковий запис: вводить ім'я, пароль і деяку додаткову інформацію. Наприклад, інколи вимагається місце проживання, стать, інтереси, додаткова електронна адреса. Більшість сайтів містить службу нагадування паролів, якщо ви їх забули. Для цього ви маєте увести відповідь на якесь запитання, яке вам буде задано, коли ви, вказавши логін, виберете пункт „Забули пароль?”. Для підтвердження правильності вводу своїх даних та захисту від автоматичної реєстрації, вам буде запропоновано вказати набір символів з малюнку, який з'явиться на екрані. Тому при реєстрації режим показу малюнків треба увімкнути.

Деякі сайти дозволяють настроїти мейлер на віртуальну адресу, яку ви на них отримали. При цьому, замість IP-адреси SMTP та POP3 сервера, потрібно вказувати URL-адресу сайта без символів <http://www>.

### **5.3.2. Пошук інформації в Інтернеті. Пошукові сервери**

Пошук здійснюється за допомогою пошукових серверів (search engines). Типова структура пошукового сервера містить базу даних (БД) і програму для їх обробки.

Будь-яка пошукова система складається з двох частин: інтерфейсу і спеціальної програми, встановленої на пошуковому сервері, яка перебираючись від сторінки до сторінки за посиланнями, індексує всю текстову інформацію в свою базу даних. Зазвичай такій програмі дається сторінка з деякою кількістю посилань на різні каталоги ресурсів. Програма, перебираючи по черзі посилання, набирає все більший обсяг інформації. Коли користувач вводить в цю систему яке-небудь слово для пошуку, воно шукається в базі даних на сервері, а користувачу видається низка посилань на ті сервери, на яких задане слово зустрічається.

Оскільки сервери постійно міняються, програмі доводиться періодично повертатися до вже проіндексованих серверів для оновлення інформації, інакше вже через два-три місяці половина посилань, які пошукова машина видає у відповідь на запит, будуть недоступні. У різних пошукових системах період оновлення може істотно відрізнятись, причому часто чим менший обсяг проіндексованої інформації система містить, тим частіше вона повертається до кожної з сторінок для перевірки на наявність змін.

Для підвищення ефективності пошуку інформації бажано дотримуватись наступних рекомендацій:

1. Використовуйте фрази. Замість одного слова краще набрати у відповідному полі введення цілу фразу, яка більш конкретно описує тему пошуку.
2. Будьте як можна більш точними у визначеннях. Ключові слова повинні бути вкрай інформативними. Наприклад, при пошуку вільно поширюваних архіваторів замість фрази "безкоштовні програми" спробуйте набрати "безкоштовний архіватор, доступний для завантаження".
3. Для побудови запитів активно використовуйте команди "AND" та "OR". Це дозволить вам включати або виключати для пошуку певні слова, присутні на сторінках, що обробляються пошуковою машиною.
4. Сортуйте результати. Більшість систем дають можливість користувачеві здійснювати повторний пошук даних серед віднайдених результатів.

#### **5.3.2.1. Українські пошукові сервери**

**BIGMIR.NET (<http://www.bigmir.net/>)**



У листопаді 2003 року Internet-холдинг Sputnikmedia.net оголосив про завершення розробки власної пошукової системи по українським сайтам і введенні її в експлуатацію.

Новий сервіс доступний всім бажаючим з головної сторінки порталу bigmir.net. На сьогоднішній день проіндексовано близько 7 млн. сторінок на 52 тис. серверах, що відносяться до України.

#### **МЕТА (<http://www.meta.ua>)**

Проект "Мета" досить молодий, однак вже встиг зарекомендувати себе як авторитетний ресурс в українському Internet. На заголовній сторінці є можливість вибору української, англійської чи російської версії сайту. Пошук можна проводити як по всьому документу, так і по тексту, заголовку, ключовим словам, підписам до картинок, посиланням і коментарям. МЕТА розуміє нові формати документів, такі, як .doc, .xls, .rtf, .zip, .rar і т.п.

У доповнення до словникових морфологічних аналізаторів, що використовуються в системі, були розроблені модулі "ймовірностної морфології" для української, російської та англійської мов. Вони дозволяють з дуже високою точністю відмінювати "несловникові" слова.

Система складається з 4-х пошуково-індексуєчих серверів і 1-го веб-сервера, що дозволяє значно прискорити оновлення пошукового індексу і надання нових українських ресурсів, підвищивши також швидкість обробки запитів. На сьогоднішній день система щодня обробляє близько 4-5 GB нових документів, а загальний об'єм документів в індексі перевищує 100 GB (5 млн. 567 тис URL).

#### **TOPPING (<http://www.topping.com.ua>)**

Проект Ping, що розвивається компаніями PACO Links і Skyline Electronics, один з перших українських порталів. На даний момент в системі зареєстровано біля 3000 сайтів і їх кількість постійно зростає.

### **5.3.2.2. Російські пошукові сервери**

#### **RAMBLER (<http://www.rambler.ru>)**

Сервер Rambler компанії Stack (рис. 5.18) є одним з самих великих і найбільш відомих пошукувачів російського Internet. Система містить мільйони документів з більш ніж 15 тис. Web-вузлів, а над підтримкою всього цього проекту трудяться біля сотні фахівців.

Рейтинг Rambler Top 100 дуже часто використовується багатьма компаніями як показник авторитетності тих або інших ресурсів. Навіть широко відомий [www.anekdot.ru](http://www.anekdot.ru) для визначення кількості відвідувань користується послугами лічильника Rambler. Значущість цієї системи пошуку для російськомовного Internet можна, мабуть, порівняти з авторитетом AltaVista в англійській частині мережі.

### **АPORT (<http://www.aport.ru>)**

Пошукова система "Апорт!" належить компанії "Агама" і розробляється при підтримці Intel.

Пошук в "Апорт!" ведеться по базі даних, що містить понад два мільйони документів, оригінали яких розташовані на більш ніж 13 тис. серверів. Система може шукати необхідну інформацію по різних словоформах введених слів, виправляти в них помилки, підтримувати пошук по фразах, в яких легко вказати обмеження у відстані між фрагментами тексту, задане потрібною кількістю слів або фраз, а також допустимий тимчасовий період створення документів.

Серед інших приємних особливостей цього сайту можна назвати індикатор завантаження, рейтинги самих уживаних ключових слів і найбільш популярних серверів.

### **ЯNDEX (<http://www.yandex.ru>)**

Сайт Yandex, що належить компанії CompNet, є одним з проектів, що динамічно розвиваються в РуНеті. У ньому пошук ведеться більш ніж по 30 тис. серверів, проіндексовано біля 2 млн. документів, і ці показники постійно зростають.

Основною перевагою системи Yandex є здатність знаходити задані слова незалежно від форми, в якій вони вживаються в документах. Цікаво, що система здатна "відмінювати" навіть ті слова, яких немає в словнику, причому пошук може вестися як за всіма формами ключових слів, так і за конкретно заданими.

Дана пошукова система дозволяє встановлювати відстань між фрагментами ключової фрази з урахуванням їх порядку, посилювати значущість того або іншого слова, використовувати уточнюючі слова. Пошук можна також вести по заголовках документів, по посиланнях, що зустрічаються на сторінках. Крім того, в Yandex легко знайти документи, схожі по значенню на ті, що зацікавили вас при попередньому запиті. Можна обмежувати пошук серед вже знайдених документів, тобто проводити своєрідне "чищення" результатів запиту.

Yandex має можливість пошуку зображень "Яндекс.Картинки". "Яндекс.Картинки" – найбільша база російськомовного пошуку в своїй категорії (34 млн. картинок в базі, 900 тис. користувачів в тиждень).

"Яндекс" - перший з російських пошуковців почав індексувати документи не-Web-форматів. Yandex має пошук по форматах \*.rtf, \*.pdf та \*.doc.

### **5.3.2.3. Міжнародні пошукові сервери**

#### **YAHOO! (<http://www.yahoo.com>)**

Одним з найвідоміших порталів в мережі був, є, і залишається сайт компанії Yahoo, який з'явився у квітні 1994 р.

Сервер Yahoo! має приблизно 40 млн. відвідувачів на місяць, і це число безперервно зростає. Крім того, багато шанувальників і у розділу Yahoo! Headlines, де зібрані новини від провідних інформаційних агентств: Reuters, AP, MSNBC і інш. Має більше ніж 50 розділів, що стосуються самих різних облас-

тей: Yahoo! Finance для бізнесменів і фінансистів; Yahoo! Travel для мандрівників; Yahoooligans! for Kids для неповнолітніх і безліч інших. Є і вичерпне керівництво для покупців різних товарів.

Відвідувачі сайту можуть отримати безкоштовно адресу електронної пошти, скористатися можливостями мережевого пейджинга (Instant Messaging), взяти участь в on-line дискусіях (Chat) і, що особливо цінується клієнтами, створити індивідуальний портал (My Yahoo!).

Сайт має 14 локалізованих версій, де розташована інформація, що стосується тільки конкретного регіону, наприклад Yahoo! Germany або Yahoo! Asia.

### **ALTAVISTA (<http://www.altavista.com>)**

Навесні 1995 р. троє співробітників дослідницької лабораторії DEC отримали в своє розпорядження, новітній на той час, супершвидкісний сервер Alpha 8400 і розмістили на ньому доступну для пошуку інформацію з дощок оголошень (bulletin boards) за останні десять років. Як говорять співробітники компанії сьогодні: "Все почалося з любові до порядку".

До недавнього часу база даних AltaVista була самою великою в Internet. Реєстрація сайтів в ній відбувається протягом 48 годин. Серед особливостей цієї пошукової системи можна указати такі додаткові механізми, як Photo Finder – інструмент пошуку зображень, технологію онлайн-перекладу документів, розроблену компанією Babelfish. До речі, остання властивість дозволила пошуковій машині індексувати документи на декількох мовах, список яких ви знайдете на її головній сторінці.

### **GOOGLE (<http://www.google.com>)**

Сьогодні Google індексує понад 6 млрд. документів.

Дві основні задачі програмістів Google – це релевантність результатів і зручність пошуку. За останні два роки Google поліпшив Web-інтерфейс пошукової машини (додалися посилання на пошук ілюстрацій, новин, конференцій Usenet і каталогу).

Алгоритми для визначення релевантності знаходяться в стані постійного оновлення – щомісяця Google впроваджує декілька нових механізмів, покликаних поліпшити якість пошуку.

Google є найпотужнішою і найшвидшою пошуковою машиною у світі і об'єднує всі можливості вище вказаних пошукових серверів.

### **5.3.3. Поняття “чату” і форуму в Інтернеті**

“Чатом” в Інтернеті називається проведення конференції в режимі реального часу. При цьому кожен з кореспондентів цієї конференції може бачити листи, що надходять від усіх інших кореспондентів. Можна також об'єднатися з деякими кореспондентами і зробити своє листування невидимим для інших учасників конференції. Подальший розвиток швидкості передачі даних в Інтернеті та нових форматів звукових та відео файлів привів до можливості голосового та відео спілкування в режимі реального часу.

Форум нагадує чат, але в ньому обговоренню підлягає якась, наперед визначена тема. Найчастіше форуми влаштовуються для того, щоб опитати читачів якоїсь статті про їх думку. Інколи форуми призначені для постійно обговорюваної теми, наприклад, ремонт комп'ютерів чи антивіруси. Інша відмінність форуму в тому, що в будь який час ви можете прочитати всі послання, адресовані на цей форум. В чаті ж, все, що написано декілька годин тому, пропадає.

### 5.3.3.1. Чат та форум на сайтах

Щоб отримати доступ до форуму або чату на якомусь сайті, потрібно знайти текст, який би запрошував прийняти участь у обговоренні чогось. Цей текст обов'язково матиме слово „чат” або „форум”.

Для прикладу, розглянемо сайт [www.gorod.dp.ua](http://www.gorod.dp.ua), який дозволяє з певного списку вибрати пункт „Форум” (рис. 5.9).

На сторінці форуму ви бачите повний перелік тем, за якими ведеться дискусія (рис. 5.10). Обравши потрібну (двічі клацнути лівою кнопкою мишки), ви бачите перелік повідомлень, які написали читачі цього форуму. Вибравши потрібне повідомлення, можете його прочитати. Якщо натиснути на кнопку „Отвечить”, ви зможете вступити в дискусію з автором листа. Якщо натиснути кнопку „Новая тема”, то межах цього ж форуму ваше повідомлення потрібно створити з новою темою.

Доступ до чату виглядає як список коротких реплік, які подають учасники чату в режимі реального часу. Кожній репліці передують ім'я, яке ви собі обрали для спілкування. В будь який момент ви можете написати свою репліку, звертаючись до когось із учасників чату. Ваша репліка буде видна одразу всім, хто „знаходиться” в чаті.



Рис. 5.9. Звертання до форуму на сайті [www.gorod.dp.ua](http://www.gorod.dp.ua)

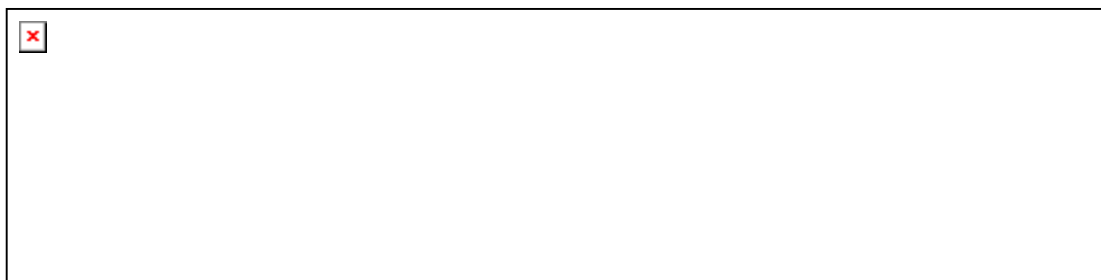


Рис. 5.10. Перелік тем на форумі сайту [www.gorod.dp.ua](http://www.gorod.dp.ua)

### 5.3.3.2. Програмно забезпечений чат

Такий чат забезпечується вже відомим браузером Opera. Достатньо тільки в головному меню програми обрати пункт „Service - Mail&Chat Account - Chat”. Почне роботу майстер налаштування цього режиму роботи програми. Достатньо буде обрати нікнейм (nickname, англ. – псевдонім ) для спілкування в чаті та вказати, через який чатовий сервер буде здійснюватися спілкування: OperaNet Europe, UnderNet Europe, UnderNet North America, IR Chat Asia, IR Chat Europe, EF Net Asia, EF Net Europe, EF Net North America, DAL Net Europe, DAL Net Asia, DAL Net North America, FreeNode North America, FreeNode Europe, FreeNode Asia або FreeNode Oceania.

При роботі програми в цьому режимі, ви можете бачити повідомлення про підключення до обраного чату. Саме вікно чату, мало відрізняється від інших вікон чату, які забезпечують чатові сайти.

Щоб вести розмови через Інтернет, необхідно мати певне устаткування. Комп'ютер повинний бути обладнаний звуковою картою і підключений до мікрофона і динаміків. Звукова карта може бути дуплексною або напівдуплексною. Комп'ютер повинний бути підключений до Інтернету за протоколом TCP/IP. Хоча в зустрічі можуть брати участь кілька людей, але аудіо- і відеоз'єднання може бути встановлено тільки з одним користувачем.

Програма Microsoft Net Meeting дозволяє людям із різних точок земної кулі спілкуватися один з одним через Інтернет, виконувати спільну роботу і користуватися загальними ресурсами.

Net Meeting надає наступні можливості:

- Виклик користувачів, що працюють на інших комп'ютерах, через локальну мережу або Інтернет, за допомогою Net Meeting. Зв'язок буде встановлено, якщо у користувача, що викликається, є Net Meeting або інша стандартна програма мережних конференцій.
- Використовуючи Net Meeting, можна відправляти своє відеозображення або зображення обговорюваного об'єкта. Для цього знадобиться карта відеозахвату і камера або камера, що підтримує відеосигнал для Windows. Відеосигнал можна одержувати навіть без відеоустаткування.
- За допомогою NetMeeting можна відчинити один з додатків (програм) і передати його в загальне користування, щоб інші учасники зустрічі бачили, як ви в ньому працюєте. Ви можете дозволити їм разом із вами пра-

цювати у вашому додатку.

- При спільній роботі в додатку, курсором можна управляти тільки по черзі. Але іноді необхідно, щоб всі учасники зустрічі могли працювати одночасно. Саме таким додатком є «Дошка», яка відкриває додаток Paint, в якому всі учасники можуть одночасно писати і малювати на дошці і бачити її зміст.
- Додаток «Розмова» дозволяє учасникам інтерактивної зустрічі відправляти одне одному повідомлення в режимі реального часу. Якщо під час зустрічі хто-небудь відчинить додаток «Розмова», то вікно розмови відобразиться на екранах всіх учасників. Оскільки тільки двоє учасників можуть користуватися звуковим і відеозв'язком одночасно, вікно розмови особливо зручно для зустрічі кількох людей (рис. 5.26). У списку ярликів ви-кликати перераховані ваші друзі і колеги з указівкою поточного стану підключення до мережі.
- На Web-сторінку можна помістити посилання, за яким вас зможуть ви-кликати користувачі Internet Explorer і Net Meeting.

Під час зустрічі можна передати файл усім її учасникам. Для цього достатньо перетягнути цей файл у список учасників на екрані „Текущий вызов”.

#### 5.3.4. Програма ICQ

ICQ (гра слів від англійського I Seek You – "я тебе шукаю") – це програма, що дозволяє кожному користувачу Інтернету установити в будь-який конкретний момент, хто з його друзів по заданому списку (які також є клієнтами програми) приєднаний до мережі. Причому всі приєднані розбиваються на кілька категорій: ті, хто виражає готовність до діалогу і прийняття повідомлення, ті, хто хоче, щоб йому не заважали, ті, хто не торкався клавіатури вже кілька хвилин (і на його екрані з'явилася картинка, що захищає екран), і т.д. Таким чином, кожен користувач ICQ знає, яке положення його віртуальних друзів у мережі: з ким він може спілкуватися, передати електронну пошту, продемонструвати який-небудь сайт чи просто довідатися, що його друг знаходиться за комп'ютером. Це схоже на можливість задалегідь, перш ніж набираєш номер телефону друга, довідатися, чи знаходиться він у будинку, чи зайнята лінія, а також хоче він говорити з тобою чи ні.

Важлива особливість ICQ – у тому, що ви можете працювати в Word, Excel чи будь-якому іншому додатку, залишаючи при цьому ICQ включеною, і весь цей час продовжувати одержувати чи відправляти повідомлення друзям, – звичайно ж, при умові, що ви підключені до мережі через телефон.

Серед доступних функцій ICQ – чат, повідомлення, електронна пошта, передача URL-адрес і файлів. Програма дозволяє мати онлайновий зв'язок.

ICQ – як факсовий апарат: ніхто не зможе нею користатися, поки партнер не придбає аналогічну програму. Тому, щоб процвітати, потрібно добратися до "критичної маси". "Розмножуються" вони просто: кожний хоче, щоб у його друзів теж була така програма, щоб знати, хто зараз знаходиться в мережі, а кого –

немає. Сама ICQ заохочує користувачів рекомендувати її своїм друзям і повідомляє, коли приятель переніс її у свій комп'ютер.

### **5.3.5. Безперервне отримання інформації (“викачування”)**

Значна кількість сайтів містять на своїх веб-сторінках посилання на файли, що містять потрібну вам інформацію. Це можуть бути нові програми чи нові настройки (оновлення) для програм, які у вас вже є: музика, кінофільми, енциклопедії, інструкції по роботі, ремонту, книжки і т. ін.

Деякі з цих файлів мають значний розмір: десятки, а то і сотні мегабайт. Для того, щоб скористатися ними, їх треба переписати з того сервера, де вони знаходяться, на свій комп'ютер. При відносно невеликій швидкості з'єднання, яку мають більшість користувачів Інтернету (4-7 Кб/с) – час отримання такого файлу розтягується на термін від десятків хвилин до декількох годин. За цей час зв'язок з вашим провайдером може розірватися. При цьому, той файл, який ви переписуєте (або як кажуть “викачуєте”) з сервера, буде втрачено і вам знадобиться повторно почати цей процес.

Такі складності приводять до збільшення часу користування Інтернетом, а значить, і збільшенню оплати за нього. А для надвеликих файлів їх отримання може стати просто неможливим, оскільки “коннект” (connect, англ. – з'єднання) з провайдером може бути меншим, аніж час, потрібний для їх викачування.

Для вирішення цієї проблеми були створені програми, які виконують процес отримання файлів з якогось сервера з фіксацією обсягу вже отриманої частини. Якщо станеться розрив зв'язку, ці програми, після його поновлення, починають “викачування” з того байта файлу, на якому цей розрив стався.

Іноколи у вас виникає потреба переписати увесь сайт на свій комп'ютер, щоб потім, вимкнувши Інтернет, в локальному режимі (не приєднуючись до провайдера), спокійно розібратися в змісті цього сайту. Сайт частіше всього складається зі значної кількості веб-сторінок, пов'язаних з першою гіперпосиланнями.

Описані нижче програми дозволяють отримати весь сайт з усіма веб-сторінками, оскільки вони автоматично переходять по гіперпосиланням і отримують весь пакет інформації, а не тільки т□ сторінку, яку ви розглядаєте.

#### **5.3.5.1. Програма Net Vampire**

Net Vampire – універсальний менеджер, за допомогою якого ви можете знайти список і завантажити файли з серверів Інтернету. Тільки перемістіть адресу сайту від вашого вікна перегляду до малого плаваючого вікна, і програма переписує цей файл для вас.

Net Vampire знайде місцезнаходження файлу в Internet, перевірить місце перебування для доступності і швидкості, установить зв'язок дискового набору, коли складено розклад, і почне завантаження. Вам немає необхідності хвилюватися про зайняті сервери і невдачі зв'язку – програма викличе сервер, поки зв'язок установлюється і в резюме покаже вам, скільки ще залишилося.

На завершення, Net Wampire сповістить вас текстовим повідомленням або звуком, відкриє отриманий файл або почне його перевірку вашим вірусним інформатором, роз'єднає модем або навіть припинить роботу вашої системи – у залежності від результатів цієї перевірки та вашої обачності.

Net Wampire має простий, інтуїтивний інтерфейс (рис. 5.11).

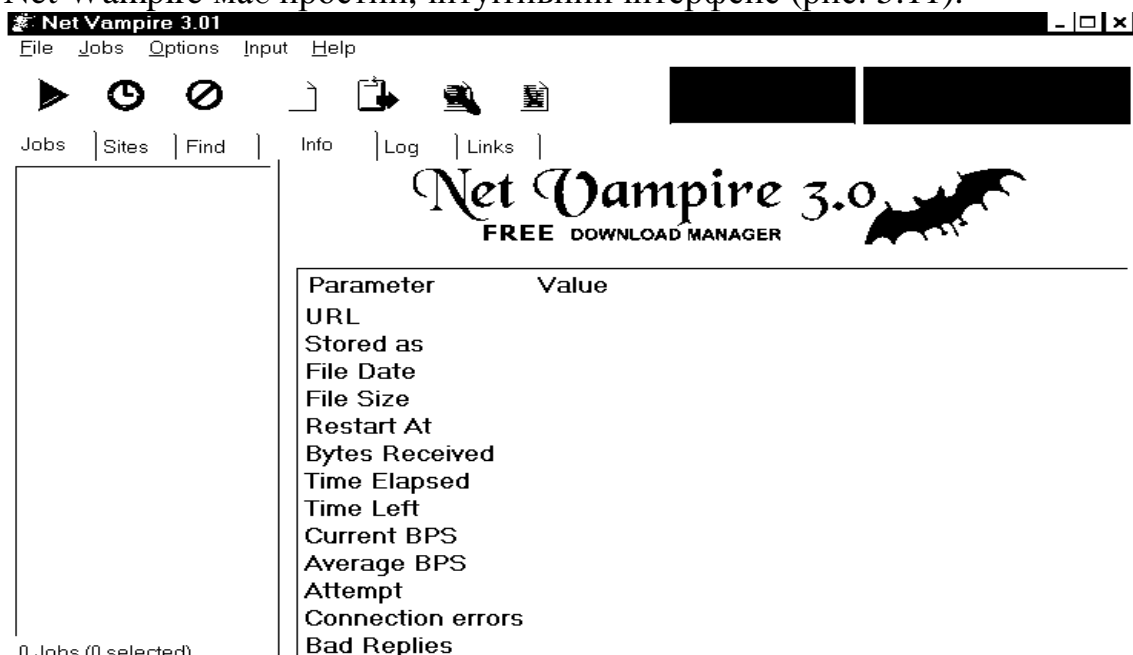


Рис. 5.11. Основне вікно програми Net Wampire

Виберіть **“Варианти/Сети”** в головному меню і надрукуйте вашу адресу електронної пошти в полі **“Адреса электронной почты”**. Це використовуватиметься, як пароль протягом анонімного початку сеансу для серверів FTP. У більшості випадків це тільки елемент ввічливості, щоб показати вашу дійсну адресу електронної пошти серверу, до того ж, деякі сервери не дозволятимуть вам почати сеансу якщо адреса електронної пошти хибна.

Якщо ви використовуєте повноваження, щоб приєднатися до Internet, клацніть на закладці **“Полномочия”** і додайте всі повноваження, які ви плануєте використати до **“Список полномочий”**. Виберіть повноваження по умовчання для протоколів FTP чи HTTP.

Якщо ви приєднуєтеся до Internet, використовуючи зв'язок дискового набору, клацніть на закладці дискового набору.

Щоб дати можливість об'єднання з вашим вікном перегляду, виберіть пункт перегляду **“Вход / Монитор”** в основному меню.

Якщо у вас є час, щоб завантажити що-небудь, відкрийте веб-сторінку або довідник FTP у вашому вікні перегляду, потім перейдіть на зв'язок з **Сетевой Корзиной** Net Wampire.

Мережний діалог відкриється полем **“Расположение”**, яке заповнюється з URL. Клацніть стартову кнопку, щоб додати до списку робіт ще одну і виконати її.

Утримуйте **Alt**, клацаючи на зв'язку у вашому вікні перегляду, щоб послати URL до Net Wampire. Якщо Net Wampire не працює, він почне роботу автомати-



чно. Вікно перегляду працює з Internet Explorer 4.0 та з його наступними версіями, і з усіма версіями Netscape.

Щоб дати можливість контролю буфера обміну, виберіть **“Буфер обміна Вхід/ Монитор”** в головному меню.

Якщо ви маєте файл, який отримано тільки частково іншою програмою, ви можете легко імпортувати через Net Wampire кінець завантаження. Виберіть **“Файл Імпорта Робот”** в головному меню, виберіть файл на відкритому вікні діалогів і клацніть **ОК**.

З Net Wampire 3.0 ви можете додати значення URL-ів до довідників FTP, так як і до індивідуальних файлів, список файлів в довіднику буде завантажений і збережений в форматі HTML. В списку зазначення URL-ів ці файли відображаються на закладці зв'язків в правій панелі. Ви можете легко сортувати зв'язки, вибирати деякі або всі, вручну чи з використанням фільтра, додавати вибрані зв'язки для завантаження, використовуючи контекстне меню.

Завантажені документи HTML і весь отриманий URL аналізуються, включаючи і розроблені за допомогою Java applets, ActiveX, JavaScript. Коли нова робота додана до списку, всі установки встановлюються до значень по умовчання.

Щоб редагувати установки для специфічної роботи, відкрийте діалог **“Установки”** за допомогою подвійного клацання лівої кнопки миші на вибраному файлі в списку або за допомогою вибору **“Установки Работы”** в головному меню.

Коли ліміт до ряду діючих спільно зв'язків не дозволяє всім активним задачам починатися в один і той же час, Net Wampire вибирає початок наступної Роботи випадково. Роботи з вищим пріоритетом матимуть більші шанси, щоб початися. Найвища пріоритетна установка має номер 9.

Деякі сервери НТТР розпізнають тільки ті замовлення, які посилаються специфічним вікном перегляду. Для цього Net Wampire удає, що це Internet Explorer або Netscape.

Ви можете задати значення “таймаут” окремо для встановлення зв'язку, відповідей команди і даних. Міняйте ці установки, щоб досягнути максимальної дієвості для кожного випадку.

### 5.3.5.2. Програма REGET

Одна із самих зручних програм, призначених для викачування файлів з FTP і НТТР серверів. Самі розробники назвали її "менеджер викачувань". На тлі конкурентів, ReGet дуже вигідно виглядає за рахунок абсолютної готовності вести техніко-рятивні бесіди російською мовою.

Настроїти ReGet рекомендується вже при першому запуску (звісно пізніше Ви зможете внести будь-як зміни в конфігурацію). Нічого складного, задача з чотирьох кроків:

- сказати Next (Далі);

- визначити місце збереження файлів, що завантажуються, (за замовчуванням, C:\My Downloads) і сказати Next (Далі);
- задати адреси FTP і HTTP Proxy (наприклад, proxy.ukr.net:8080), якщо забули – запитайте у Вашого провайдера, якщо не бажаєте використовувати – зніміть прапорці, і знову Next (Далі);
- визначте рівень функціональних можливостей програми: Simple mode – для новачків, Advanced mode – для досвідчених, і жміть на Finish (Готово).

Якщо на останньому етапі Ви не забрали прапорець в рядку View Documentation, то при першому старті програми відкриється вікно короткого опису.

Основна особливість ReGet полягає в тому, що ви створюєте чергу викачування. Нове викачування можна додати в чергу декількома способами:

- Вибрати пункт **"Download using ReGet"** в контекстному меню Internet Explorer (щоб викликати його, треба клацнути на посиланні правою клавішею миші).
- Перетягнути посилання з браузера у вікно ReGet і відпустити.
- Встановити стеження за буфером обміну на вкладці **"Інтеграція"** меню **"Option"** в ReGet, потім натиснути правою клавішею миші на посиланні і вибрати **"Copy Shortcut"**.
- Скопіювати адресу в буфер обміну, а потім вручну додати в чергу, натиснувши **Ctrl+V** або вибравши **"Insert"** в меню **"Edit"**.
- Скопіювати адресу в буфер обміну, потім клацнути правою клавішею миші на іконці ReGet в System Tray і вибрати **"Insert URL"** з меню.
- Вибрати пункт **"New Download"** в меню **"Download"** і ввести URL з клавіатури.

ReGet може бути запущений з англійським інтерфейсом. Щоб змінити мову інтерфейсу, виберіть в меню **View/Language**, потім виберіть з представлених потрібну мову.

### 5.3.4.2. Програма Opera

Режим докачки вмикається у цій програмі автоматично, як тільки ви натиснули слово „викачування” на сайті, де пропонується отримати файли (рис. 5.12). Інтерфейс одразу показує процес отримання файлів та місце, куди вони будуть покладені.

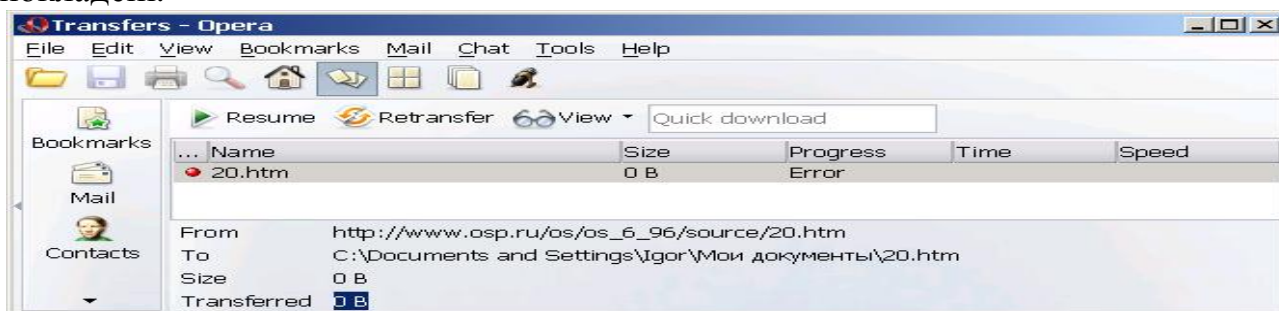


Рис. 5.12. Браузер Opera в режимі докачки

### 5.4.Індивідуальне завдання № 4.

## Робота з браузером Opera та Internet Explorer

Кожен етап роботи треба ілюструвати документом у форматі Word, шляхом збереження зображення через PrintScreen і наступним перенесенням цього зображення у Word гарячими клавішами “Ctrl+V”.

Кожна дія має бути повторена у обох браузерах, і в тексті документа визначено власне відношення до зручності роботи з кожним браузером.

За цю роботу викладач виставляє дві оцінки – за кожний із браузерів.

1. Відкрити сторінку одного з пошукових чи інформаційних серверів Google, Yahoo, bigFoot, Meta, Vrama, Rambler чи якогось іншого.
2. Переходити з однієї сторінки на іншу за гіперпосиланнями.
3. Знайти Web-сторінки з економічними новинами, з показниками курсів валют, з курсами акцій різних компаній та фірм.
4. Додати в папку “Вибране” свою папку та гіперпосилання на будь-яку Web-сторінку.
5. Копіювати у звіт таблиці, малюнки, текст. Зберегти Web-сторінку через “Файл-Зберегти як...” в різних режимах збереження «Тільки текст», «Повна сторінка» і т.ін.
6. Вставити у звіт гіперпосилання на будь-яку Web-сторінку.
7. Налаштувати кнопку “Home” на будь-яку сторінку.

### 5.5. Індивідуальне завдання №5. Робота з електронною поштою

1. Налаштування Outlook Express.

Завантаження Outlook Express: Пуск>Програми>Outlook Express.

Для початку роботи з електронною поштою, в першу чергу, необхідно налаштувати обліковий запис пошти.

До нього входить адреса поштового сервера, e-mail адреса і ім'я користувача.

Сервіс→Облікові записи→вкладка «Пошта».

Кнопка «Додати» →Пошта...

Далі дотримуйтесь вказівок майстра. Ім'я користувача: clientN, де N – номер комп'ютера: client02, client04...і т.д. Поштова адреса: client@ekit20.ekit, пароль збігається з ім'ям користувача.

Порада: не варто при налаштуванні вводити і зберігати пароль в системі (знімайте прапорець «Запам'ятати пароль») – це різко знижує рівень безпеки, особливо при роботі за стороннім комп'ютером. Вводити пароль краще щоразу при вході в Outlook Express чи одержанні пошти.

Адреса поштового сервера: 172.16.3.20. (Без крапки наприкінці).

Після закінчення роботи майстра зайдіть у властивості створеного облікового запису і змініть його назву на своє прізвище, потім зробіть його використовуваним за замовченням.

2. Робота з Outlook Express: відправлення пошти, читання пошти.

Відправте просте текстове повідомлення (лист) про успішне налаштування Outlook Express колегам по ланці. Прочитайте вхідну пошту, використовуйте

можливість автоматичної відповіді на вхідний лист (Контекстне меню вхідного листа > Відповісти відправнику).

### 3. Використання спеціальних можливостей e-mail.

Відправте колегам по ланці листа з форматуванням (розмір шрифту, колір, вирівнювання і т.д.), уставленими малюнками і гіперпосиланнями на зведений документ по Л.Р№3, що знаходиться на сервері.

Відправте тим же адресатам листа з прикріпленим звітом по індивідуальному завданню №1 і своєю частиною зведеного документа (лист Excel).

4. Відкрити програму Opera.
5. Створити в поштовому клієнті свій обліковий запис (відповідно до настанов у поясненнях по завданню).
6. Обмінятися електронними листами за допомогою цієї програми.

## 5.6. Індивідуальні завдання №6.

### Робота з допоміжними програмами для роботи в Internet'і: ReGet та ICQ і порівняння їх можливостей з особливостями роботи браузерів.

Кожен етап роботи ілюструвати документом у форматі Word, шляхом збереження зображення через PrintScreen і наступним перенесенням цього зображення у Word гарячими клавішами "Ctrl+V". В тексті документа має бути визначено власне відношення до зручності роботи з кожною з програм.

За цю роботу викладач виставляє три оцінки – за кожною з програм.

1. Увімкнути програму ReGet.
2. Вибрати HTML-сторінку іншої групи і спробувати переписати великий файл, на який там дано посилання. Перервати роботу програми через "Ctrl+Alt+Del", а потім повторити операцію. Відмітити: з якого байта починається копіювання файлу.
3. Повторити операцію по п.2 з будь-яким браузером.
4. Виконати настройку програми ICQ по власним параметрам. Отримати власний ICQ-код і спробувати знайти за цією програмою свого одногрупника.

### Контрольні питання

1. Що таке браузер?
2. Що таке гіперпосилання?
3. Як увімкнути режим пошуку за ключовим словом?
4. Що таке електронна пошта? Для чого вона призначена?
5. Що входить в обліковий запис пошти?
6. Як прикріпити до листа файл? Які файли можна прикріплювати?
7. Чим програма ICQ зручніша за інші мейлери?
8. Для яких цілей створена програма ReGet?

*У розділі подано прийоми та поняття застосування мейлерів та браузерів для збереження, пошуку і розповсюдження економічної інформації.*

# 6. БЕЗПЕКА ТА ЗАХИСТ КОМП'ЮТЕРНИХ МЕРЕЖ

*Вивчивши матеріали цього розділу, студент має опанувати принципи та прийоми захисту комп'ютерів від вірусів.*

Internet і інформаційна безпека несумісні по самій природі Internet. Вона народилася як чисто корпоративна мережа, однак, у даний час за допомогою єдиного стека протоколів TCP/IP і єдиного адресного простору поєднує не тільки корпоративні і відомчі мережі (освітні, державні, комерційні, військові і т.д.), що є, по визначенню, мережами з обмеженим доступом, але і рядових користувачів, що мають можливість одержати прямий доступ у Internet зі своїх домашніх комп'ютерів за допомогою модемів і телефонної мережі загального користування.

Як відомо, чим простіше доступ у Мережу, тим гірше її інформаційна безпека, тому з повною підставою можна сказати, що споконвічна простота доступу в Internet - гірше злодійства, тому що користувач може навіть і не довідатися, що в нього були скопійовані - файли і програми, не говорячи вже про можливість їхнього псування і коректування.

*Безпека мережі (Network security)* – заходи, які захищають інформаційну мережу від несанкціонованого доступу, випадкового або навмисного втручання в нормальні дії або намагань руйнування її компонентів. Безпека інформаційної мережі включає захист обладнання, програмного забезпечення, даних і персоналу.

Безпека даних є однією з головних проблем у Internet. З'являються всі нові і нові страшні історії про те, як комп'ютерні зломщики, що використовують усе більш витончені прийоми, проникають у чужі бази даних. У банківській сфері проблема безпеки інформації ускладнюється двома факторами: по-перше, майже всі цінності, з якими має справу банк (крім готівки і ще дечого), існують лише у виді тієї чи іншої інформації. По-друге, банк не може існувати без зв'язків із зовнішнім світом: без клієнтів, кореспондентів і т.п. При цьому по зовнішніх зв'язках обов'язково передається та сама інформація, що виражає собою цінності, з якими працює банк (або зведення про ці цінності і їхній рух, що іноді коштують дорожче самих цінностей). Ззовні приходять документи, по яких банк переказує гроші з одного рахунка на інший. Зовні банк передає розпорядження про рух засобів по кореспондентських рахунках, так що відкритість банку задана а наперед.

Сьогодні всі шкідливі програми називають вірусами, хоча це не зовсім коректно. Це скоріше певна „торгова марка”. Лабораторією Касперського створена досить велика вірусна енциклопедія, яку можна проглянути на сторінці

<http://www.viruslist.com/index.html>.

Фінансові збитки від вірусів в світі (тільки витрати на відновлення інфраструктури) в 2005 році склали приблизно 14 млрд. доларів, а за оцінкою ФБР сукупні збитки в цьому ж році склали 63 млрд. доларів.

Оскільки в сферу уваги інформаційної безпеки всі частіше попадають більш серйозні речі, ніж звичайні віруси, фахівці з комп'ютерного захисту ввели термін malware – "зловмисне програмне забезпечення", яким позначають будь-який код, що наносить шкоду комп'ютеру чи його власнику. Для опору посяганням на приватну інформацію користувача, необхідна глибока і добре продумана система оборони, що складає з декількох бар'єрів між зловмисним програмним забезпеченням і системою.

Захисні технології по рівню розвитку не обганяють інформаційні технології, а йдуть слідом за ними. Доки немає загрози, ніхто не буде витратити зусилля та кошти для захисту.

Основні загрози безпеки: відкриття конфіденційних даних, втрата чи пошкодження даних, зміна даних, відмова в обслуговуванні, помилки програмного забезпечення та відмова від зобов'язань.

При вирішенні проблем безпеки потрібно оцінити важливість того, що ви намагаєтесь захистити для вас і для потенційних крадіїв. Очевидно, що цінність інформації на комп'ютері звичайного користувача, сервері банку чи Центральної виборчої комісії різна.

Фахівці з інформаційної безпеки відмічають зміну напрямку у розвитку шкідливих програм. Спочатку віруси писались для задоволення, а основна мета сьогодні – отримання прибутку.

## **6.1. Огляд шкідливих програм**

В залежності від середовища існування розрізняють наступні типи вірусів: файлові, завантажувальні, файлово-завантажувальні, макровіруси та мережні віруси. Умовно і найбільш грубо віруси можна розділити на програмні і скриптові. Перші представляють собою окремі, автономні саморозмножуючі програми, часто з деструктивною дією. „Скриптові” – не являються окремими програмами, а є набором інструкції для якоїсь з популярних програм (Internet Explorer, Microsoft Office і т.п.). Загальна риса всіх скрипт-вірусів – прив'язка до однієї з вбудованих мов програмування.

За способом розповсюдження програми можна розділити на: комп'ютерні віруси, мережні хробаки та троянські програми. За 2005р. приблизно вдвічі збільшилась кількість троянських програм і приблизно на стільки ж збільшився розмір антивірусних баз. Тільки 5% всіх шкідливих програм було написано для задоволення, 75% – для отримання грошей, 20% – для обох цілей.

### **6.1.1. Комп'ютерні віруси**

*Комп'ютерний вірус* – це невелика за розмірами програма, яка може самостійно розмножуватися, “приписуючи” себе до інших програм, тим самим заражаючи їх, а також виконувати різні небажані дії на комп'ютері.

Програму, в якій знаходиться вірус називається “*зараженою*”. Коли така програма починає працювати, то спочатку управління отримує вірус. Вірус знаходить і “заражує” інші програми, а також виконує інші шкідливі дії (наприклад, пошкоджує файли, чи таблицю розміщення файлів на диску, “засмічує” оперативну пам'ять і т.д.). Доки на комп'ютері заражені відносно мало програм, наявність вірусу може бути практично непомітною. Але, через деякий час на комп'ютері появляються небажані події. Наприклад: деякі програми перестають працювати, чи починають працювати неправильно; на екран виводяться сторонні повідомлення, символи і т.д.; робота на комп'ютері суттєво сповільнюється; деякі файли виявляються пошкодженими і т.п.

Ці типи вірусів фактично втрачають свої позиції (в 2005р. займали приблизно 5%) і віддають першість більш фінансово-орієнтованим.

### **6.1.2. Троянські програми**

Троянські програми, чи просто троянці – шкідливі програми, які самі не розповсюджуються, а маскуючись під популярну програму, спонукають користувачів переписати та встановити шкідника на власний комп'ютер самостійно.

По виконуваним діям троянські програми можна умовно розділити на

- утиліти несанкціонованого віддаленого адміністрування – дозволяють зловмиснику віддалено управляти зараженим комп'ютером;
- утиліти для проведення DDoS-атак (Distributed Denial of Service – атаки типу „відмова в обслуговуванні) – вибирають інформаційні ресурси жертви, в результаті чого система перестає виконувати свої функції і стає недоступною;
- шпигунські програми – таємно наглядають за діями користувача і записують в свій журнал цікаві для зловмисника дані;
- рекламні програми – дозволяють вбудувати рекламні об'яви в часто використовувані додатки;
- програми подзвону – намагаються за допомогою модему і телефонної лінії додзвонитися до платного серверу і заставити користувача заплатити послуги;
- сервери розсилки спаму (непотрібної реклами) – дають можливість перетворити чужий ПК в сервер розсилки спаму;
- багатокomпонентні троянці-завантажувачі – переписують з Інтернет і встановлюють в систему інші шкідливі коди чи додаткові компоненти

Розрізняють троянські програми, які постійно забезпечують доступ до зараженого комп'ютера, тримаючи на ньому відкритий порт транспортного протоколу та програми, які не тримають відкритих портів. Останні пересилають зловмиснику певну інформацію, наприклад паролі або копії текстів, що набираються з клавіатури.



Ріст кількості щомісячно знайдених нових модифікацій троянських програм за перші шість місяців 2006 року склав 9%.

### **6.1.3. Мережні хробаки**

Мережні хробаки не змінюють файли на дисках, а розповсюджуються в комп'ютерній мережі, впроваджуються в операційну систему комп'ютера, знаходять адреси інших комп'ютерів і розсилають по цим адресам свої копії, використовуючи різне середовище розповсюдження.

Віруси-хробаки мають манери справжніх хакерів, находячи діри в обороні підключених до мережі комп'ютерів і проникаючи в беззахисні порти.

Інтернет-хробаки розповсюджуються по Інтернет, LAN-хробаки – по локальній мережі, IRC-хробаки – через чати.

Сьогодні жертвами хробаків є десятки мільйонів комп'ютерів. А такі монстри, як Blaster, Sasser чи Sober мають десятки видів модифікацій і паралізують роботи цілих мереж. Як правило, вони відносно нешкідливі для даних (не рахуючи постійні перезавантаження комп'ютера, які викликає Blaster), але їх „переміщення” по мережі генерує масу зайвого трафіку, займаючи канали зв'язку.

## **6.2. Технології інформаційної безпеки**

Методи захисту від комп'ютерних вірусів та втрати інформації:

1. Загальні засоби захисту інформації, які потрібні не тільки для захисту від вірусів, а як страховка від фізичного пошкодження дисків, неправильно працюючих програм чи помилкових дій користувача. Є два різновиди таких засобів:
  - 1.1.копіювання інформації – створення копій файлів і системних областей дисків;
  - 1.2.розподіл доступу – запобігає несанкціонованому використанню інформації.
2. Криптографічний захист конфіденційних даних.
3. Профілактичні заходи, що дозволяють зменшити вірогідність зараження.
4. Спеціалізоване програмне та апаратне забезпечення.

### **6.2.1. Захист операційних систем**

Фірма Microsoft обладнала програми Microsoft Office захистом від запуску макросів. При відкритті будь-якого документа, що містить вбудовані макроси, на екрані з'являється попередження (рис. 6.1.) і запит на відкриття файлу з макросами. До речі, якщо в параметрах програми вказаний високий рівень захисту, то незважаючи на згоду відкриття, макроси все одно блокуються.

Часто виникає питання, яким же чином вірус може потрапити в комп'ютер? Наприклад, вірус Blaster може перевантажити комп'ютер, через мережу, навіть не проникаючи в нього.

Протокол TCP, який є основним для сучасної мережі, передбачає використання 65536 ( $2^{16}$ ) портів. Крім того, можуть використовуватись інші протоколи, наприклад UDP, в асортименті якого знаходиться така ж кількість. Перші 100 портів (0-99) відносяться до категорії стандартних – їх використання жорстко регламентується і назавжди закріплені за певним службами, наступні 100-1024 називають „широко використовуваними” і пропонують трохи менше повноважень своїм клієнтам. Інші, приблизно 60000 – віддаються стороннім програмам, і не завжди надійно захищені.

В ідеальному варіанті відкритими для використання можуть бути лише ті порти, які зарезервовані за певними відомими системі програмами, інші „двері” повинні бути надійно заблоковані. Але на практиці, в системі захисту Windows (як інших операційних систем) є багато „дірок”, які дозволяють зловмиснику отримати доступ до інформації через „недозволений” канал. Наприклад, Blaster, атакує комп’ютер через порт 135 чи 139.

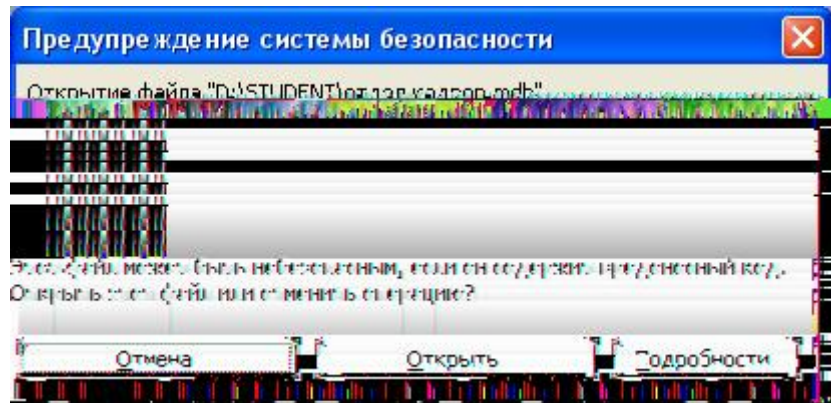


Рис. 6.1. Попередження про наявність макрокоманд в файлах.

Крім того, існують спеціальні програми для сканування портів, за допомогою яких можна знайти вхід практично в будь-який комп’ютер.

Розробники систем „латають” ці „дірки” спеціальними програмами, які часто просто називають „латками”, а взагалі це оновлення для використовуваної версії операційної системи. Для цього досить зайти на сайт Windows Update (<http://windowsupdate.microsoft.com>), чи клацнути по значку Windows Update в меню Сервіс програми Internet Explorer.

Крім того, практично всі сучасні системи дозволяють створювати різні види користувачів та надавати їм певні права доступу до програмного та апаратного забезпечення системи. Причому, деякі функції регулюються додатковим програмним забезпеченням. Наприклад, Адміністратор безпеки (рис. 6.2)

### 6.2.2. Налаштування параметрів безпеки в браузері Internet Explorer

Налаштування системи мережної безпеки складається із трьох пунктів, дістатися до яких можна таким шляхом: меню **Сервіс-> Свойства обозревателя**. У цьому діалоговому вікні нашу увагу зосередимо на закладках **Безопасность**, **Конфиденциально** та **Дополнительно**.

1. Вибираємо закладку **Безопасность**. Вказуємо на верхній панелі зону **Интернет** і на нижній панелі натискаємо кнопку **Другой**. Ці маніпуляції надають змогу дістатися до параметрів (рис. 6.2-6.5), які керують виконанням активних

кодів, вбудованих у html-сторінки. По кожному з параметрів є три можливих варіанти: Запретить, Предлагать та Разрешить. Перший варіант – відключить всі запропоновані елементи, другий – потребує постійних уточнень користувача для практично всіх дій, третій – для ризикових людей.

2. Вкладка **Конфиденциальность** використовується для настройки рівня мережної анонімності при прийомі *cookies* (мітки, які залишають на комп'ютері користувача деякі сайти Інтернет, призначені для ідентифікації користувача при повторному вході) – від повного їх блокування до прийому від будь-якого Web-сайта.

На вкладці **Дополнительно** (рис. 6.4) слід звернути увагу на дві групи параметрів – **Обзор** та **Безопасность**.

Група **Обзор** може стати в нагоді при встановленні параметрів відображення елементів на сторінці, тобто при пришвидшенні завантаження, можна відмовитись від малюнків, анімаційних блоків, звуку і т.п. Але, треба відмітити, що на банерну рекламу, як правило, ці налаштування не діють.

Натисніть кнопку двічі **ОК**, і вважатимемо налагодження безпеки у браузері закінченим.

### 6.2.3. Антивіруси

Однією з перших захисних технологій, до сих популярних на ринку, є антивірусний захист, що з'явився в середині 80-х років. Для захисту від вірусів розробляються спеціальні антивірусні програми, що дозволяють виявляти віруси, лікувати заражені файли і диски, запобігати підозрілим діям. Всі сучасні антивіруси оснащені механізмом автоматичного оновлення антивірусних баз даних через Інтернет. По даним лабораторії Касперського середній час виходу оновлення для антивірусної програми зараз складає десь 1 годину 22 хвилини. Більш того, якщо епідемія небезпечна, то urgent-оновлення випускається приблизно за 30 хвилин. Найпопулярнішими в Україні і країнах СНД є такі антивіруси: Антивірус Касперського; Антивірус Doctor Web; Panda Antivirus; Norton Antivirus; McAfee VirusScan.

У кожної з цих програм є свої переваги і недоліки. Кожна з них заслуговує на увагу споживачів. І, що важливіше, кожен з перелічених вище антивірусів може забезпечити ефективний захист вашого ПК.

**Пакет AVP (AntiViral Toolkit Pro).**

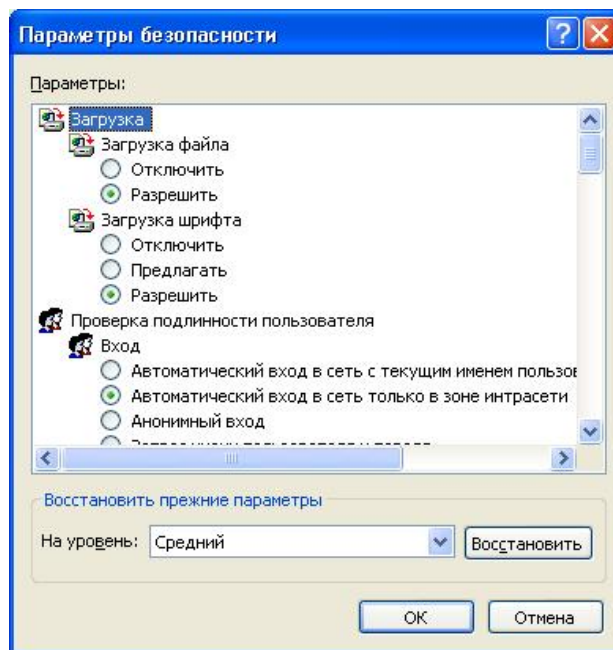


Рис. 6.2. Налаштування параметрів завантаження

Мабуть самий популярний і потужний пакет, створений в Росії в лабораторії Є. Касперського. Антивірус AVP (AntiVirus Program) відноситься до поліфагів, у процесі роботи перевіряє оперативну пам'ять, файли, в тому числі архівні, на гнучких, локальних, мережних і CD-ROM дисках, а також системні структури даних, такі як завантажувальний сектор, таблицю розділів і т.д. Програма має евристичний аналізатор, котрий, за твердженнями розробників антивірусу здатний знаходити майже 80% усіх вірусів. Здійснює пошук і вилучення найрізноманітніших вірусів, у тому числі: поліморфних, або вірусів, що самошифруються; стелс-вірусів, або вірусів-невидимок; нових вірусів для Windows; макровірусів, що заражають документи Word і таблиці Excel.

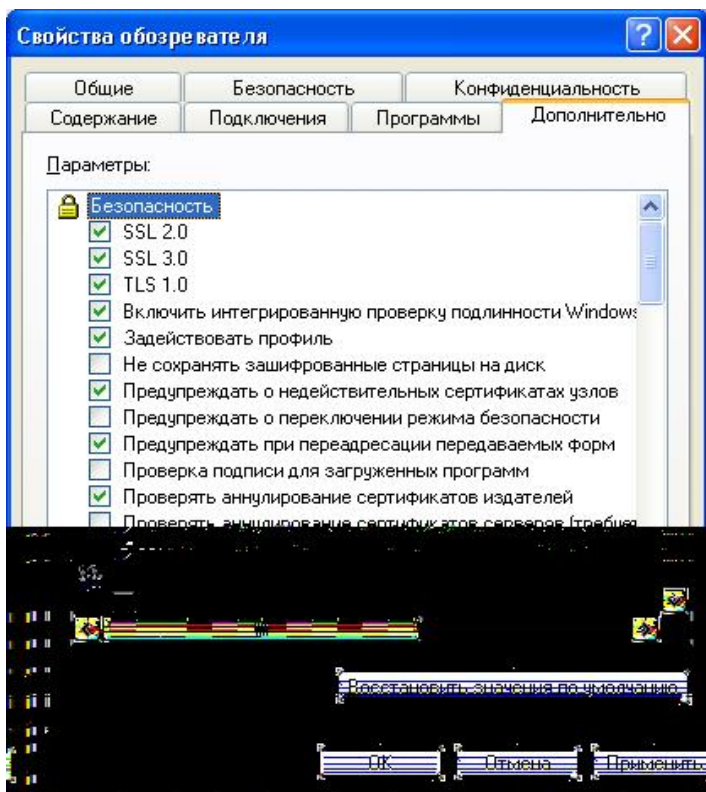


Рис. 6.3. Настроювання додаткових параметрів безпеки

вірусу. Якщо в об'єкті буде виявлений вірус, монітор запропонує вилікувати заражений об'єкт, видалити об'єкт або заблокувати доступ до нього. Тобто AVP монітор дозволяє виявити і виділити вірус до моменту його реального проникнення до системи.

Остання версія системи має досить цікавий інтерфейс (рис. 6.4), чотири основних компонента захисту, незалежні один від одного, які можуть бути відключені або видалені. Перший відповідає за безпеку файлів на комп'ютері, другий – за відправлення і приймання поштових повідомлень, третій – за роботу в Інтернеті, четвертий компонент забезпечує, так званий проактивний захист.

Компонентами активного захисту є перевірка VBA-макросів, контроль за змінами в системному реєстрі, перевірка цілісності додатків і аналіз їх активності.

Крім того, програма AVP здійснює контроль файлових операцій у системі у фоновому режимі, виявляє вірус до моменту реального зараження системи, а також визначає невідомі віруси за допомогою евристичного модуля.

Для перевірки наявності вірусів диску чи дискети потрібно:

Завантажити програму *Пуск* ® *Програми* ® *AntiViral Toolkit Pro* ® *AVP Сканер*.

До складу пакету входить також модуль *AVP Монітор*, який завантажується при включенні комп'ютера, постійно міститься в пам'яті комп'ютера і контролює звернення до файлів та завантажувальних секторів. Перед тим як дозволити доступ до об'єкта, модуль перевіряє його на наявність

Департамент спеціальних телекомунікаційних систем та захисту інформації (ДСТСЗІ) СБ України 6.08.01 підтвердив відповідність "Антивіруса Касперського" нормативним документам, що регламентують вимоги до засобів технічного захисту інформації. Крім того, експерти ДСТСЗІ дозволили використовувати цю програму при побудові систем антивірусного захисту у складі комплексних систем захисту інформації

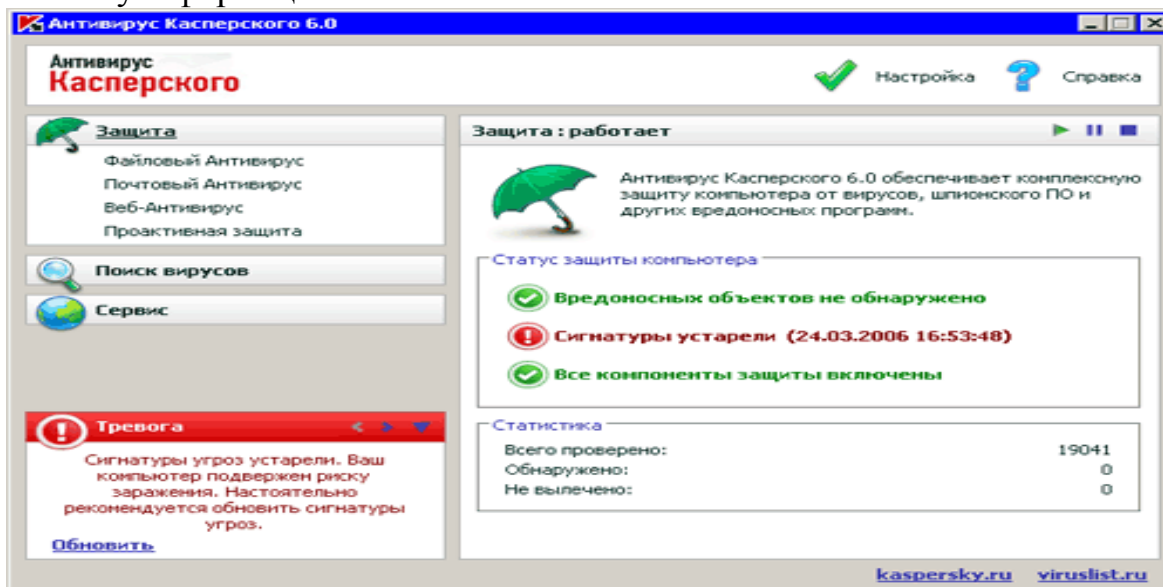


Рис. 6.4. Новий антивірус Касперського

До недоліків системи можна мабуть віднести серйозні вимоги до ресурсів комп'ютера.

### **DRWEB**

Один з кращих антивірусів із сильним алгоритмом знаходження вірусів. Поліфаг, здатний перевіряти файли в архівах, документи Word і робочі книги Excel, виявляє поліморфні віруси, котрі в останній час, отримують все більше поширення. Достатньо сказати, що епідемію дуже небезпечного вірусу OneHalf зупинив саме DrWeb. Евристичний аналізатор DrWeb, досліджуючи програми на наявність фрагментів коду, характерних для вірусів, дозволяє знайти майже 90% невідомих вірусів. При завантаженні програми в першу чергу DrWeb перевіряє самого себе на цілісність, після чого тестує оперативну пам'ять. Програма може працювати у діалоговому режимі, має дуже зручний інтерфейс користувача, який можна налаштувати.

### **Сканер Avast.**

Є порівняно новим антивірусним сканером. При його роботі, в правому є відповідний індикатор. Не великого розміру.

Захист доступу Avast базується на так званих резидентних провайдерах. Це спеціальні модулі, що використовуються для захисту специфічних підсистем комп'ютера, таких як електронна пошта, файлова система і т.п.

## 6.2.4. Міжмережні екрани

Міжмережний екран - це програмний або апаратно-програмний комплекс, що дозволяє розділити мережу на дві чи більше частин і реалізувати набір правил, що будуть визначати умови проходження пакетів з однієї частини в іншу.

Власник комп'ютера, що має вихід в Internet, установлює міжмережний екран (рис. 6.5), щоб запобігти одержанню сторонніми конфіденційних даних, котрі зберігаються на комп'ютері, а також для контролю за зовнішніми ресурсами, до яких мають доступ інші користувачі даної комп'ютерної системи.

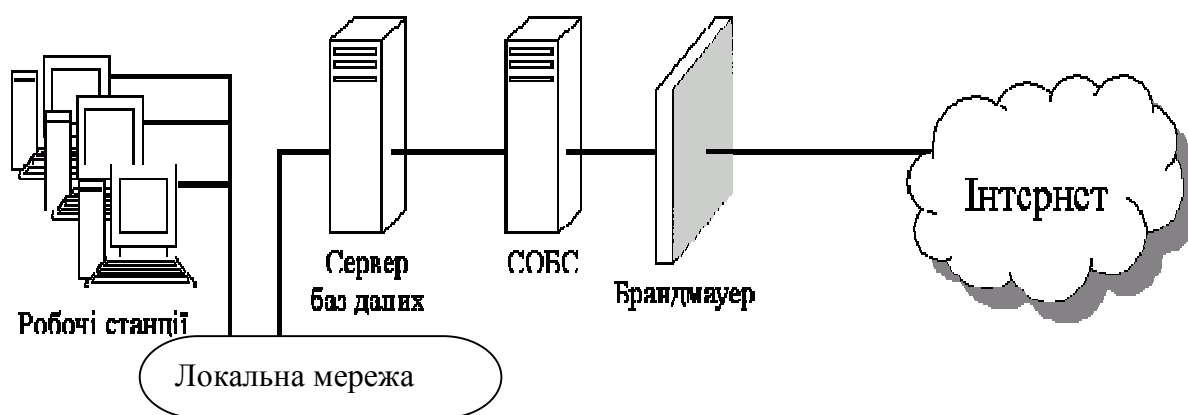


Рис. 6.5. Схема розташування брандмауера

Міжмережні екрани здатні вирішувати ряд завдань стосовно захисту від найбільш імовірних атак для внутрішніх мереж. У вітчизняній літературі частіше зустрічаються терміни іноземного походження: брандмауер (німецького походження) і firewall (англійське). Поза комп'ютерною сферою брандмауером (чи firewall) називають протипожежну стіну, зроблену з вогнестійких матеріалів, щоб перешкодити поширенню пожежі. У сфері використання комп'ютерних технологій міжмережний екран становить собою бар'єр, що захищає від умовної пожежі – спроб зловмисників несанкціоновано вторгнутися у внутрішню мережу для вчинення протиправних дій. Міжмережний екран покликаний створити безпечний доступ до зовнішньої мережі та обмежити доступ зовнішніх користувачів до внутрішньої мережі.

Мірою ефективності firewall є зовсім не його здатність відмовити в наданні сервісів, а його властивість надавати сервіси користувачам у ефективному, структурованому й надійному середовищі. Системи firewall мають аналізувати вхідний та вихідний мережний трафік і правильно визначати, які з транзакцій є санкціонованими, а які - ні.

Найбільш розповсюджені наступні системи: Outpost, McAfee, Norton, Sygate и ZoneAlarm.

### **Outpost Firevall Pro v3.51.**

Потужний брандмауер (рис. 6.6) для безпечної роботи в мережі. Має безліч функцій: контроль додатків, що користаються Інтернетом, контроль компонен-

тів, прихованих процесів, п'ять режимів роботи, набір стандартних правил для визначених додатків, детектор атак, контроль умісту, й ін.

За замовчуванням використовує модулі:

- DNS, що блокує й оповіщає про неправильні DNS запити;
- Детектор атак, що запобігає атаки на ваш комп'ютер (portscan, nuke, winnuke і ін.);
- Інтерактивні елементи, де ви можете визначити, що робити з різними елементами веб-сторінки (Active, Java, Java script, Gif, Flash, Referers, сховані фрейми, зовнішні об'єкти, cookies і спливаючі вікна), причому ви має можливість задати особливі параметри для будь-якого окремо узятото сайта (наприклад, у вас встановлено блокування Flash, а ви зайшли на сайт із Flash іграми, те просто внесіть адресу сайта в список і виберіть "дозволити" для flash);
- Реклама, де блокуються банери (88x31 і ін.) і реклама (певні слова), причому знову ж ви можете додати і видалити слова і розміри банерів (наприклад, бувають баннери 90x35 чи 90x50; додайте їх у список і вони не будуть завантажуватися);
- Журнал усіх подій, де ви можна переглянути всі події і дії програми;
- Вміст, де ви можете заблокувати сайти з визначеним змістом чи адресою, і фільтрація поштових вкладень, де визначаються типи файлів, що прикріплюються, і дії ("повідомити" і "перейменувати").
- В описуваній версії також є антишпигунський сканер.

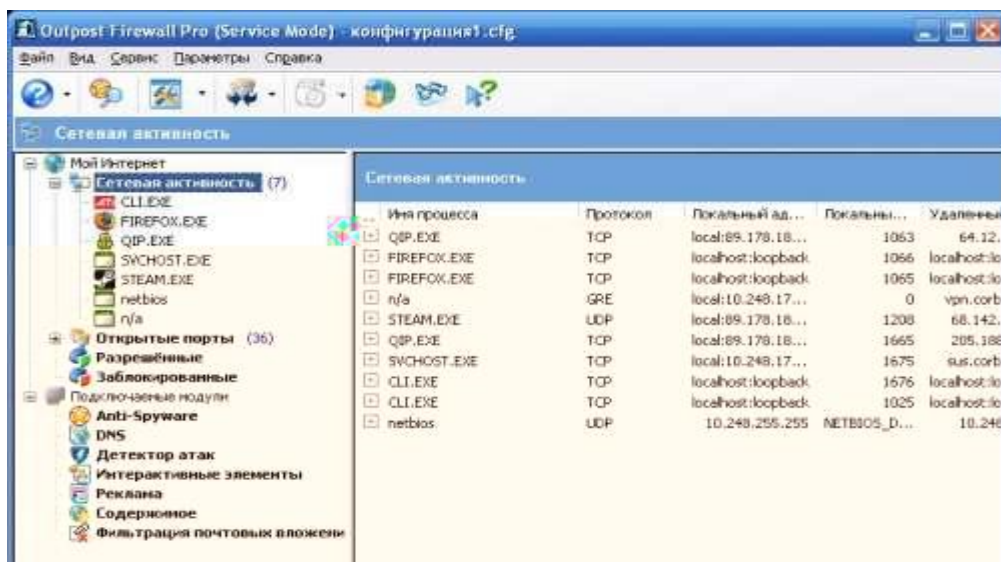


Рис. 6.6. Діалогове вікно брандмауера Outpost

**Zone alarm pro 6** – один із самих потужних і надійних брандмауерів (рис. 6.9) для безпечної роботи в Мережі. Має дружній і легко зрозумілий інтерфейс, гнучке налаштування параметрів захисту, блокування різноманітних Pop-up, Cookies, Java-апплетів, Java-скриптів, ActiveX, Gif-анімації, контроль програм і компонентів, антишпигунський сканер та інше.

## 6.2.5. Системи контролю змісту та антиспаму

Спам – небажані поштові повідомлення. Ця проблема стосується кожного, хто використовує електронну пошту. Відомо, що навіть у самих надійних провайдерів може бути "витік" баз поштових адрес.

Оскільки ця проблема виникла досить давно, існує дві групи програм, призначених для боротьби з спамом. Перші містять базу "чорних" адрес спамерів, яка постійно оновлюється, і при надходженні від них листа видаляють його. Інші, навпаки, пропонують користувачу створити "білий" список адресатів, а інші повідомлення блокують.

**Anti-Spammer** (рис. 6.7) – працює як самостійний додаток, не вбудовуючись в поштовий клієнт, але в той же час може працювати разом з поштовою програмою.

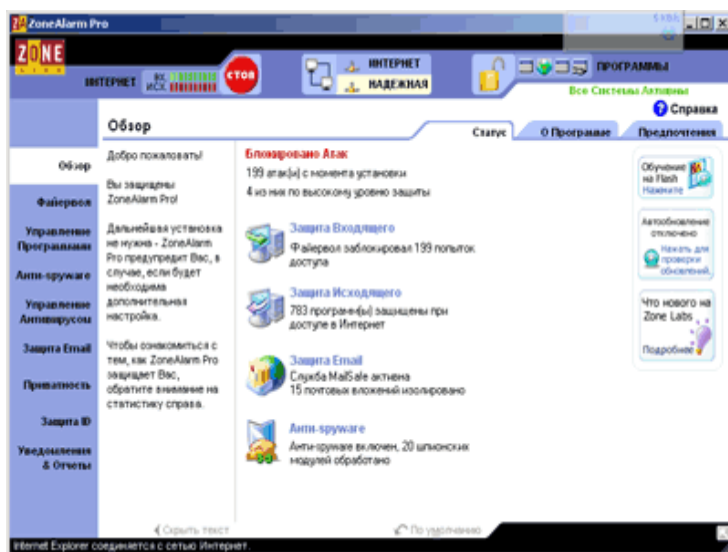


Рис. 6.7. Вікно Zone alarm pro 6

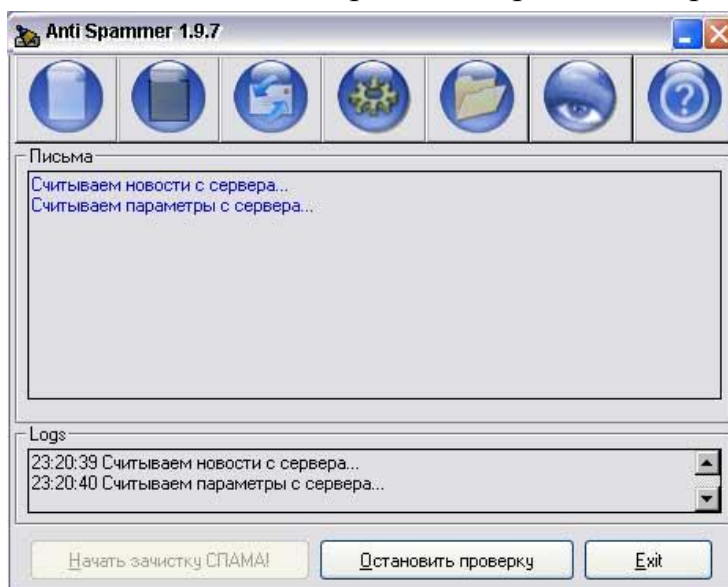
Обробляє небажані повідомлення за принципом "білого" списку (хоча "чорний" список теж використовується). Вам необхідно додати в програму облікові записи пошти (їх кількість не обмежена), а після цього занести в базу утиліти адреси ваших кореспондентів і вказати тему, листи з якою будуть про-

пускатися.

Програма зчитує адресу відправника і тему кожного з листів, після чого перевірить адреси на відповідність "білому" списку. Другий етап - порівняння теми листа. Якщо адреса чи тема листа збігаються з занесеними в список, лист пропускається, якщо ж ні, адреса відправника заноситься в "чорний" список. Далі програмою адресату відправляється повідомлення про те, що лист не був отриманий і що для його одержання повідомлення потрібно відправити ще раз, указавши задану тему листа. Сам лист видаляється чи скачується і зберігається в програмі.

Після обробки Anti-Spammer вхідної пошти, можна запускати поштову програму.

**Spam Washer** є посередником між сервером і поштовим клієнтом, однак може служити і заміною поштовику,





дозволяє фільтрувати вхідну пошту, використовуючи при цьому як убудовані правила, так і створювані користувачем.

### 6.2.6. Автентифікація користувача та шифрування даних

Першим етапом на шляху захисту ресурсів інформаційної системи є організація перевірки, чи є користувач, який входить в систему, тим, за кого він себе видає. Сама процедура перевірки носить назву *автентифікації* користувача.

Автентифікацію користувача можна вважати основою програмно-технічних засобів безпеки, оскільки багато інших сервісів розраховані на обслуговування саме іменованих суб'єктів. Як правило, процедура перевірки складається з двох етапів: ідентифікації та верифікації.

Під *ідентифікацією* розуміють процедуру представлення користувача системі. Звичайно це відбувається шляхом уведення імені, під яким користувач зареєстрований в системі.

*Верифікація* – це процедура, яку система виконує для того, щоб переконатися, що користувач, який входить в систему є саме тим, чиє ім'я він ввів при ідентифікації. Для цього користувачу пропонується ввести пароль, який буде порівняний з паролем в записі обліку цього користувача.

Використання для автентифікації клієнта двох елементів утруднює нелегальне проникнення в систему, оскільки для успіху потрібно подолати два незалежні бар'єри.

У мережному середовищі, коли сторони територіально рознесені, у розглянутого сервісу є два основних аспекти:

1. що служить верифікатором (тобто використовується для підтвердження сутності суб'єкта);
2. як організований (і захищений) обмін даними.

Суб'єкт може підтвердити себе, використавши:

- щось, що він знає (пароль, особистий ідентифікаційний номер, криптографічний ключ і т.п.);
- щось, чим він володіє (особисту картку чи інший пристрій аналогічного призначення);
- щось, що є частину його самого (голос, відбитки пальців і т.п., тобто свої біометричні характеристики).

Унеможливити читання даних сторонніми особами дозволяють процедури шифрування.

*Шифрування (ciphering, encryption)* – це перетворення даних у форму, що не дає можливості безпосереднього сприйняття зашифрованої інформації.

Шифрування здійснюється з використанням криптографічного ключа (cryptographic key, cryptokey або просто key), з використанням ключа здійснюється і зворотна процедура дешифрування, тобто приведення зашифрованої інформації до первинного варіанта.

Прикладом застосування криптографії є банківські послуги і платежі в діалоговому режимі. Електронний бізнес може існувати тільки за умови конфіден-

ційності інформації, одним із найкращих методів забезпечення якої є криптографія.

### **6.3. Індивідуальні завдання №7.**

#### **Налагодження безпечних параметрів доступу до Internet в браузері. Використання антивірусних програм та міжмережних екранів.**

1. Встановити безпечні параметри доступу до Інтернет в браузері Internet Explorer.
2. Визначити, чи мають подібні можливості налаштування інші використовувані Вами браузери.
3. Перевірити, який додатковий захист встановлено на вашому комп'ютері: антивіруси, брандмауери, антиспамери і т.п. По-можливості, перевірте які вони мають функції.
4. Визначити, коли в останнє оновлювалась антивірусна база даних.
5. Перевірити дискету та папку Student на наявність вірусів.
6. Перелічити в яких випадках, при виконанні лабораторних робіт Ви стикались с процедурою авторизації. Скопіювати одне з вікон у звіт.

#### **Контрольні запитання**

1. Дайте визначення вірусу. Як вони себе поведуть?
2. Що таке антивіруси. Для чого вони призначені?
3. Які заходи потрібно застосовувати для зниження вірогідності попадання вірусу в комп'ютер?
4. Чим відрізняється брандмауер від антивірусної програми?
5. Особливості дії троянських програм.
6. В чому небезпека мережних хробаків?
7. Як Ви вважаєте, які програми найбільш небезпечні для корпоративних мереж та звичайних користувачів.
8. Які додаткові профілактичні засоби безпеки, на вашу думку, потрібно використовувати?
9. Як перевірити комп'ютер на вірус?
10. Чи обов'язкове спільне використання процедур ідентифікації та верифікації? Якщо ні, привести приклади.

*В розділі подано опис антивірусних програм та основні прийоми їх використання.*

# 7. WEB-ДИЗАЙН

*У розділі подано зміст основних тегів, що створюють веб-сторінку та порядок їх розташування.*

Усі Web-сторінки створюються на основі мови HTML (*HyperText Markup Language* – мова розмітки гіпертексту).

Вона була створена вченими Європейського центру ядерних досліджень (CERN, м. Женева). Наприкінці 80-х років у CERN зайнялись проблемою збереження і відображення даних, які отримували колеги-фізики. Складність полягала у тому, що кожний фахівець, який приїздив до Центру, застосовував власні методи відображення інформації, і потрібно було терміново створити універсальну систему, яка б не залежала від комп'ютерної платформи і була б досить простою.

Ідея розв'язання проблеми обміну документами між різними комп'ютерами полягала у тому, що документи мали бути розмічені за допомогою визначеного коду – HTML. Такі документи могли б читатися на будь-якому комп'ютері, на якому встановлена одна програма перегляду – браузер.

HTML являє собою набір команд, що описують структуру документа. Така мова розмітки дозволяє виділити в документі окремі логічні частини (заголовки, абзаци, списки і т.д.), але не задає конкретні атрибути форматування. Конкретний вид форматування задає використовуваний при читанні документа браузер, що забезпечує відображення інформації на моніторі.

Конструкції цієї мови – *теги* – дозволяють керувати шрифтом, кольором тексту і фону, визначати посилання, вставляти графіку, аудіо і відео. Теги ніколи не відображаються при перегляді сторінки – вони служать для управління оформленням. Якщо ж при відкритті сторінки ви бачите теги, це означає, що при написанні коду були допущені помилки.

*Гіпертекст* – текст, у якому містяться посилання на інші текстові документи. Це дає можливість при читанні тексту легко і швидко переходити до іншої зв'язаної з ним текстової інформації. Зв'язаний з посиланням текст може бути фрагментом того ж самого документа, чи іншим текстовим документом, що зберігається на будь-якому іншому ПК в глобальній мережі.

Документи, підготовлені мовою HTML, називають HTML-документами (Web-сторінка, HTML-сторінка, Internet-сторінка). Для того, щоб переглянути HTML-код сторінки, потрібно після завантаження її в браузері вибрати команду *Вид – Просмотр HTML-кода*. Код поточної сторінки відобразиться у вікні Блокнота. Віднедавна, деякі автори сайтів, почали захищати свою власність і блокувати копіювання змісту і відповідно перегляд коду.

*Web-сайт* – сукупність Web-сторінок, об'єднаних однією загальною темою, поміщених, як правило, на одному вузловому комп'ютері і контролюються однією людиною чи групою людей.

Посилання в гіперпертекстових документах називаються *гіперпосиланнями* (*лінк, гіперлінк*). В якості гіперпосилання може бути використаний будь-який текст, символ чи рисунок.

Однією з цілей проекту World Wide Web була розробка стандартного способу вказівки посилань на доступні Internet ресурси. Для рішення цієї задачі було введено поняття URL.

*URL (Uniform Resource Locator - універсальний покажчик ресурсу)* являє собою адресу ресурсу в Internet.

Приклад URL: <http://www.mysite.ua/my-page.htm>

## 7.1. Проектування структури сайту

Приступаючи до розробки сайту потрібно чітко визначити його призначення та аудиторію на яку він розрахований. Єдиної класифікації сайтів не існує, але є багато різних підходів до визначення цього питання. Поклавши в основу класифікації призначення сайтів, можна умовно розбити їх на наступні групи:

1) **Комерційні сайти.** Основна мета будь-якого комерційного сайту полягає в обслуговуванні користувачів таким способом, що приносить компанії пряму або побічну вигоду. Комерціалізація Інтернет з кожним роком стає все інтенсивнішою. З'явився вид бізнесу, який називають електронним, тобто ділова активність, що використовує можливості глобальних інформаційних мереж для отримання прибутку. Можна також виділити певні підгрупи сайтів:

- а. Web-представництво фірми – призначені для підтримки бізнесу якої-небудь фірми чи організації, та використовуються для поширення інформації про продукти і послуги, що надаються фірмою, і способах зв'язку з фірмою;
- б. Електронна крамниця – продаж товарів через мережу;
- в. Електронний аукціон – аналог звичайного аукціону;
- г. Електронна біржа – торги великими партіями для юридичних осіб;
- д. Платне надання інформації та програмних продуктів (реферати, аналітична інформація, оновлення програм і т.п.).

2) **Інформаційні сайти.** Ставлять перед собою задачу інформування користувачів у певних сферах громадського життя. Як правило, інформаційні сайти відрізняються великим обсягом розміщених на них матеріалів (текстових, таблиць і графічних), їх можна умовно розбити на:

- |             |                                      |
|-------------|--------------------------------------|
| а. урядові; | г. некомерційних фірм і організацій; |
| б. освітні; | д. релігійних груп;                  |
| в. новин;   | е. суспільні.                        |

3) **Розважальні сайти.** Ставлять метою розважити своїх відвідувачів. Як правило, запропонованим товаром саме і є розвага. Розважальні сайти, в більшості, насичені графікою, анімацією і спеціальними ефектами.

4)**Навігаційні сайти.** Допомагають користувачам у пошуку потрібної інформації через Інтернет, їх часто називають Інтернет-порталами. До них можна віднести пошукові системи, каталоги, рейтинги і деякі довідкові системи.

5)**Художні сайти.** Є самовираженням автора, найчастіше, Web-дизайнера. Вплив користувача на художній сайт зводиться до того, що він може подумки схвалити його або ні. Досить часто сайти виконані в авангардному дизайні.

6)**Персональний сайт.** Часто називається персональною домашньою сторінкою. На таких сайтах розповідається про автора, коло його захоплень та розміщується інформація, якою він бажає поділитись з аудиторією. Практично всі відомі люди, на даний момент мають персональні сторінки (наприклад – Мороз Олександр Олександрович [www.moroz.com.ua](http://www.moroz.com.ua)). Як правило, сайти цього типу не мають на меті отримання прибутку.

7)**Комбіновані сайти.** Поєднують у собі два або більш типів сайтів і їх переваги (комерційний Інтернет-портал, інформаційно-комерційний сайт і т.п.).

Якщо розглядати сайти для бізнесу, то доцільність створення того чи іншого типу сайту визначається наступними основними критеріями:

1) *Коло розв'язуваних сайтом задач.* Необхідно вибрати основну мету створення сайту і другорядні цілі. Від цього багато в чому будуть залежати загальна ідеологія сайту, його структура, компонування, система навігації і т.д.

Можливі цілі створення сайту:

- повна і оперативна інформація про діяльність компанії, доступна в будь-який час доби з будь-якої точки світу;
- формування у споживачів, партнерів та інвесторів сприятливого іміджу сучасної компанії;
- інформаційна підтримка дилерів і партнерів компанії;
- продаж товарів/послуг;
- проведення тендерів на постачання сировини, матеріалів і устаткування;
- вихід на ринки СНД і Далекого Зарубіжжя, що неможливо належною мірою охопити стандартними методами реклами;
- підтримка теле/радіо/друкованої рекламної кампанії. Організація рекламної кампанії з посиленням на сайт, як на джерело найбільш повної і оперативної інформації про предмет реклами і т.д.

2) *Бюджет сайту.* У залежності від нього буде залежати вибір форми документа, на основі якого буде розроблятися сайт. Від бюджету прямо залежить також і загальний обсяг сайту, рівень його технічної складності, строки створення, число майбутніх відвідувачів, а отже, той прибуток, що зможе принести сайт його власникові. Фактично в багатьох випадках від бюджету залежить якість даного сайту і те наскільки серйозно його будуть сприймати відвідувачі. При плануванні бюджету потрібно враховувати як витрати на розробку сайту, так і витрати на його наступний супровід (своєчасне оновлення та поповнення)

3) *Інформаційне наповнення сайту.* Підбір матеріалу надзвичайно важливий фактор. Потрібно оцінити якість матеріалу, його цікавість для відвіду-

вачів. Відібраний матеріал групують по темам, які і будуть визначати розділи майбутнього сайту. Якщо матеріалу по вибраній темі набирається досить багато, слід відсортувати його по ступені важливості.

Матеріали, відібрані для сайту, потрібно організувати у визначену структуру, яка має наступні типи.

Схема	Опис
<p data-bbox="199 427 481 461">Лінійна структура</p> 	<p data-bbox="560 427 1447 591">Сторінки сайту розташовуються строго одна за одною. Зручна для створення невеликого по кількості сторінок сайту з малою кількістю гіперпосилань і послідовним викладом матеріалів.</p>
<p data-bbox="172 602 513 636">Структура у виді ґрат</p> 	<p data-bbox="560 602 1447 898">Ґрунтується на побудові системи навігації сайту, коли між вертикальними і горизонтальними елементами (сторінками) є взаємний зв'язок і можливість швидкого переходу з однієї сторінки на іншу, без необхідності відвідування проміжних сторінок. Подібна структура приводить до зайвого збільшення гіперпосилань і застосування її обмежене для об'ємних сайтів.</p>
<p data-bbox="164 911 521 987">Оптимальна структура (деревоподібна)</p> 	<p data-bbox="560 911 1447 1252">На верхньому рівні знаходиться початкова сторінка сайту, з якої відвідувач за допомогою меню чи посилань може переходити на сторінки наступного рівня. Ця структура може складатися з декількох рівнів підпорядкування. Дуже важливо визначити оптимальне співвідношення між кількістю рівнів (глибиною коренів чи висотою дерева) і кількістю варіантів вибору на кожному рівні (шириною дерева).</p>
<p data-bbox="164 1263 521 1346">Різновид деревоподібної структури</p> 	<p data-bbox="560 1263 1447 1559">Використовується в особливо складних сайтах, що мають декілька сот окремих html-сторінок. Принципово відрізняється тим, що має велику глибину сайту. При трирівневій системі навігації можна розмістити приблизно 15 000 тисяч сторінок. Обсяг інформації такого сайту може скласти кілька десятків тисяч сторінок.</p>

Відвідувач може скласти уявлення про структуру сайту за допомогою засобів навігації: списків-посилань, меню і т.д. Для кращої орієнтації відвідувачів до складу сайту може бути включена окрема сторінка, так звана **карта** сайту.

## 7.2. HTML-коди

Перш ніж створювати свої власні HTML-сторінки, слід розібратися в їх структурі та принципах функціонування, а також знати код HTML-документа. Мова HTML дозволяє формувати різну гіпертекстову інформацію на основі структурованих документів, а браузер визначає сформовані посилання і, через протокол передачі гіпертексту HTTP, відкриває доступ до документа іншим користувачам Internet.

Для створення Web-сторінок необхідний текстовий редактор, що дозволяє зберігати файли в ASCII кодї, а це означає, що HTML-документ не містить яких-небудь знаків форматування тексту. Він може містити тільки літери, цифри, розділові знаки і деякі інші друкарські символи. Найпростішим редактором, що дозволяє створювати Web-сторінки, є Блокнот, який вбудований в операційну систему Windows. Написавши чергову сторінку в HTML-тегах, потрібно зберегти її з розширенням \*.html. Використання спеціальних програм дозволить прискорити час створення Web-сторінки, але не позбавить від необхідності знання HTML-коду.

HTML-документ, по суті, є звичайним текстовим файлом. Редагувати Web-сторінки, опубліковані в мережі Internet, може лише той, хто їх створив, а не будь-який користувач, оскільки кожна сторінка має свою унікальну адресу (URL) та захищена паролем. Існування двох різних сторінок з однаковою адресою виключене. HTML-сторінка містить як звичайний текст, так і спеціальні команди розмітки (tags або теги), вкладені в кутові дужки (< і >).

*Теги мови HTML задають правила, за якими яких браузер відображає документ на екрані: розміщення тексту у вікні, представлення графічних об'єктів (малюнків), а також виведення звукових файлів, відеокліпів і т.д.*

Кожен тег має однаковий принцип написання:

<ІМ'Я ТЕГА> Вміст тега </ ІМ'Я ТЕГА >

Знаки «більше», «менше» вжито для розділення тегів один від одного і повідомлення браузеру, що це команда, а не звичайний текст.

Комбінація з відкриваючого і закриваючого тега називається контейнером тегів. „Вмістом тега” може бути текст чи інші теги.

Теги не чутливі до регістра. Це означає, що, наприклад, HTML-тег <body> буде сприйнятий браузером так само, як тег <BODY> або <bOdY>. Теги не чутливі до їх розташування на сторінці. Їх можна розмістити в один рядок, але для зручності їх перегляду при написанні тексту web-сторінки, рекомендується розташовувати теги ступінчасто, щоби теги старшого рівня були лівіше від тегів нижчого рівня.

Будь-який текстовий документ складається з окремих об'єктів: заголовків, абзаців, малюнків, таблиць, рисунків. Ці об'єкти, переведені у формат HTML, зображаються у вигляді елементів HTML. Наприклад елементом є заголовок сторінки: <TITLE> Приклад </TITLE>.

Всі елементи, передбачені в HTML, можна умовно поділити на декілька категорій:

- структурні – елементи, обов'язкові для документа, що відповідає стандарту HTML (наприклад, HTML, HEAD, BODY і TITLE);
- блокові – елементи, призначені для форматування цілих текстових блоків (наприклад, DIV, H1, H2, P), часто відокремлюються від іншої частини документа пропуском рядка;
- текстові – елементи, що задають розмітку шрифту (I, B, U, BIG, SMALL і ін.), розмітку тексту (STRONG, VAR, CITE);

- спеціальні – елементи порожнього рядка (BR, HR), впроваджені елементи (IMG, MAP, OBJECT), якірний елемент (A), елементи таблиці (TABLE), елементи фреймів тощо.

### Структура HTML-документа

HTML-документ можна умовно розбити на три частини:

<!DOCTYPE >	Службова інформація для браузера
<HTML>	
<HEAD> <TITLE> Заголовок вікна браузера </TITLE> </HEAD>	Заголовна частина документа, до якої вміщена назва документа, а також службова інформація для серверів, описи невеликих програм-сценаріїв.
<BODY> Вміст сторінки </BODY>	Тіло документа
</HTML>	

<HTML> – тег використовується для відкриття HTML-документа, з нього починається кожна Web-сторінка і закінчується закриваючим тегом </HTML>.

<HEAD> – будь-який HTML-документ складається як мінімум з двох частин: заголовка і власне документа. Даний тег визначає заголовок Web-сторінки і повинен мати обов'язковий закриваючий тег </HEAD>.

Між тегами <TITLE> і </TITLE> поміщається назва документа. По ній браузері можуть знайти інформацію, тому місце для назви завжди визначене –вгорі і окремо від вмісту HTML-документа. Відображається назва в заголовку вікна браузера, тому на нього накладається обмеження: не більше 40 символів.

<BODY>. Даний тег містить в собі безпосередньо документ, який відображається браузером. Також необхідний закриваючий тег </BODY>.

На рис. 7.1 наведено приклад застосування цих тегів.

Теги можуть бути: парними – мають початковий і кінцевий тег (наприклад, <HEAD> і </HEAD>, <BODY> і </BODY>) і одиночними (наприклад, <BR>, <!DOCTYPE>).

Часто теги, крім оформлення містять додаткові елементи, які називаються *атрибутами*. Наприклад, якщо в тег тіла документа <BODY> ввести додатковий елемент <BODY bgcolor="yellow">, то це означатиме, що документ має відображатися на жовтому тлі. Слово bgcolor є атрибутом, а yellow – значенням атрибуту.

Після написання початкового коду HTML-документа і перед розміщенням його в мережі Internet, необхідно оцінити результат роботи з точки зору користувача. Саме на цій стадії деколи виникають труднощі, оскільки готовий HTML-документ, тобто Web-сторінка, що відображається в браузері, не знаходиться під безпосереднім контролем автора, як це звично буває у випадку з надрукованими документами.



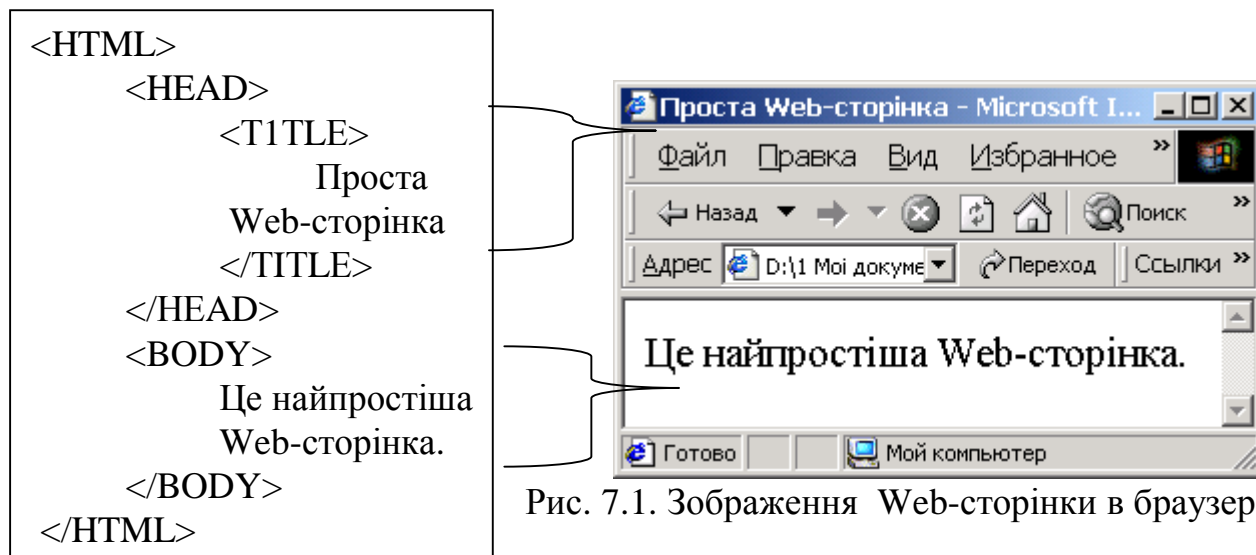


Рис. 7.1. Зображення Web-сторінки в браузері

Відображення залежить від того, який браузер використовує користувач і як він набудований – одна і та ж сторінка може в різних браузерах відображатися по-різному. Також представлення документа залежить від настройок монітора користувача, тобто від встановленої кількості точок на екрані і колірної схеми. Тому для переглядання створеного документа рекомендується використовувати декілька браузерів (як мінімум два). На сьогоднішній момент найпоширенішими браузерами для сімейства операційних систем Windows, як уже відмічалось, є Internet Explorer корпорації Microsoft, Netscape Navigator компанії Netscape і Opera фірми Opera Software.

Якщо отримане зображення у браузері вимагає коригування, потрібно внести зміни в тексті сторінки, зберегти її і натиснути кнопку „Обновити” на браузері для перевірки. Коригування виконують доки зображення не буде задовольняти повністю.

### 7.2.1. Основні теги

Всі теги перераховані не по порядку їх розміщення в документі і важливості, а згруповані по призначенню в алфавітному порядку.

#### Теги форматування сторінки

- <!-- текст коментарю --> – коментар, який інколи вставляється для пояснень певних блоків програмного коду, читачу в браузері не виводиться.
- <ADDRESS>...</ADDRESS> – відомості про автора сторінки, дата створення і оновлення. Виділяється курсивом.
- <BASE> – указує повну базову адресу, де зберігаються HTML-документи. Атрибути: href = "..." – URL-адреса, та target = "..." – ім'я кадру.
- <BODY>...</BODY> – контейнер, в який поміщується текст (тіло) Web-сторінки. Є обов'язковим елементом HTML-документа.
- <HEAD>...</HEAD> – заголовок HTML-документа.
- <HTML>...</HTML> – контейнер, в який вміщується весь HTML-документ.
- <META> – містить метайнформацію про властивості документа. Атрибути:
  - http-equiv = "..." – визначає тип властивості;

- value="..." – визначає властивість, значення: document –документ;
- content= "..." – задає значення властивості;
- url="..." – задає URL-адресу програми для властивості;
- name="..." – додатковий опис властивості. Якщо відсутній, то співпадає з http-equiv. Можливі значення: author –автор програми, author-corporate – фірма-розробник програми, keywords – ключ для пошуку (перелік слів), description – короткий опис Web-сторінки, призначений для відображення пошуковими системами.

Наприклад:

http-equiv = "content-Type" content="text/html; charset=windows-1251" – програма містить текст і коди HTML в кодуванні windows-1251.

http-equiv="content-language" content="en" – програма на англійській мові;

http-equiv="refresh" content="30" url="..." – сторінку з вказаною адресою завантажити через 30 секунд;

http-equiv="refresh" contents="30" – поточну сторінку перезавантажувати через кожні 30 секунд;

name="Keywords" content="список ключових слів" – список для пошуку по ключу даного документа пошуковими системами;

name="Description" content="короткий опис" – анотація – список для індексації документа пошуковими системами, приблизно, 200-300 символів. Звичайно в пошукових системах виводиться під назвою документа або спливає, якщо затримати покажчик миші на назві.

HTTP-EQUIV="Page або Site –Enter або Exit " – встановлює ефекти при відкритті або закритті сторінки чи сайту. До цього йде параметр CONTENT="revealTrans(Duration=4.0,Transition=12) "> (ефект при відході з сайту), де Duration задає час в секундах, а Transition - номер ефекту, який має наступні варіанти: 0 - Box in (у полі); 1 - Box out (з поля); 2 - Circle in (у коло); 3 - Circle out (з кола); 4 - Wipe up (стирання вгору); 5 - Wipe down (вниз); 6 - Wipe right (праворуч); 7 - Wipe left (вліво); 8 - Vertical blinds (вертикальні жалюзі); 9 - Horizontal blinds (горизонтальні жалюзі); 10 - Checkboard across (у шаховому порядку поперечний); 11 - Checkboard down (вниз); 12 - Random dissolve (випадковий наплив); 13 - Split vertical in; 14 - Split vertical out; 15 - Split horizontal in; 16 - Split horizontal out; 17 - Strips left down; 18 - Strips left up; 19 - Strips right down; 20 - Strips right up; 21 - Random bars horizontal (випадкові горизонтальні лінії); 22 - Random bars vertical; 23 - Random (випадковий ефект).

<SCRIPT>...</SCRIPT> – опис скрипта в заголовку. Атрибут language="..." – визначає тип скрипта.

<STYLE>...</STYLE> – контейнер опису стилю. Має необов'язковий атрибут type="text/css", який визначає тип стилю.

<TITLE>...</TITLE> – контейнер призначений для визначення назви Web-сторінки. Є обов'язковим елементом заголовка. Текст, укладений в контейнер, відображається в заголовку вікна браузера.

## Теги форматування символів

<BASEFONT> – описує стандартний шрифт для даного Web-документа. Діє до наступного тега цього типу. Має наступні атрибути:

- face="..." – визначає гарнітуру шрифту або декілька можливих шрифтів;
- size="..." – задає розмір шрифту;
- color="..." – задає колір шрифту.

<B>...</B> – виділити текст напівжирним шрифтом.

<BIG>...</BIG> – збільшує розмір шрифту щодо базового.

<CITE>...</CITE> – цитата, найчастіший текст, укладений в цей контейнер, виділяється курсивом.

<CODE>...</CODE> – машинописний текст, відображається шрифтом фіксованої ширини.

<DFN>...</DFN> – визначення, виділяється курсивом.

<EM>...</EM> – текст виділяється шрифтом із зрушенням і курсивом.

<Hn>...</Hn> – заголовок стилю n-го рівня (n = 1,2,..., 6). У стилі 1 – найкрупніший шрифт браузера, а в стилі 6 – найдрібніший шрифт.

<FONT>...</FONT> – тег опису шрифту тексту, поміщеного в контейнер. Має наступні атрибути:

- face="..." – задає гарнітуру шрифту або декілька можливих шрифтів (Arial, Times New Roman,...);
- size="..." – встановлює розмір шрифту. Визначені сім основних розмірів шрифту, що вимірюються не в пунктах, а в деяких умовних одиницях – від 1 до 7. Як правило, звичайний шрифт має розмір "3".
- color="..." – задає колір шрифту;
- style="..." – визначає опис стилю.

<PRE>...</PRE> – контейнер для тексту, відформатованого в іншому редакторі. Для його відображення застосовується тільки шрифт фіксованої ширини. Атрибут: width="..." – максимальне число символів в рядку.

<SMALL>...</SMALL> – текст, поміщений в даний контейнер, має розмір на одиницю менший, ніж основний текст.

<STRIKE>...</STRIKE> – текст, укладений в даний контейнер, перекреслюється горизонтальною лінією.

<STRONG>...</STRONG> – в даний контейнер поміщують важливий фрагмент документа. Зазвичай виділяється браузером напівжирним шрифтом.

<SUB>...</SUB> – контейнер призначений для виведення нижніх індексів.

<SUP>...</SUP> – контейнер призначений для виведення верхніх індексів.

<TT>...</TT> – текст, укладений в контейнер, відображається шрифтом фіксованої ширини.

<U>...</U> – підкреслений текст.

<VAR>...</VAR> – виділити змінні. Звичайно відображається шрифтом фіксованої ширини.

## Теги форматування абзаців

<BLOCKQUOTE>...</BLOCKQUOTE> – контейнер для цитати. Зручно використовувати для створення полів в документі, тому що цитата зсувається

вправо.

<BR> – розрив рядку, текст, розташований після нього, переноситься на новий рядок без початку абзацу. Два оператори підряд дають пропуск рядка.

<CENTER>...</CENTER> – текст, укладений в цей контейнер, виводиться по центру документа.

<DIV>...</DIV> – контейнер абзацу. Атрибут align="..." задає вирівнювання тексту абзацу в рядку: center – вирівнювання по центру вікна браузера, left – по лівому краю, right – по правому краю, justify – по ширині.

<KBD>...</KBD> – текст вирівняний по ширині з обох боків.

<NOBR>...</NOBR> – вміщений текст не розбивається на рядки.

<P>...</P> – контейнер абзацу. Має ті ж атрибути, що і <DIV>

<SAMP>...</SAMP> – моделює текст, виведений на друк. Звичайно відображається шрифтом фіксованої ширини. Призначений для виведення прикладів програм, сценаріїв і т.д.

### Теги створення списків

<DIR>...</DIR>, <UL>...</UL> або <MENU>...</MENU> – використовуються для формування маркірованого (нумерованого) списку. Має атрибут type="...", який визначає тип маркера. Вид маркерів в кожному браузері свій для різних значень: disc (диск), square (квадрат), circle (круг). В якості маркерів, можуть бути використані графічні зображення. Кожен елемент списку визначається тегом <LI>.

<DL>...</DL> – список визначень. Теги елементів визначення:

<DT> – пояснюваний термін.

<DD> – визначення терміну. Обидва тега визначення можуть також використовуватися окремо для створення нумерованих і не маркірованих списків.

<OL>...</OL> – контейнер нумерованого списку. Кожен елемент списку, як і для нумерованого, визначається тегом <LI>. Атрибути:

- start="..." – задає початковий номер в списку;
- type="..." – встановлює тип маркера. Має наступні значення: A – маркер у вигляді прописних літер, a – малих літер, I – великих римських цифр, i – малих римських цифр, 1 – арабських цифр.

### Теги розміщення гіперпосилань

<A>...</A> – контейнер для опису покажчика гіперпосилання. На покажчику потрібно клацнути мишею, щоб виконати перехід по посиланню. Може мати наступні атрибути:

- href="..." – URL-адреса документу (фрагменту, сторінки чи серверу), на який встановлюється посилання;
- name="..." – використовується для надання імені-ідентифікатора (аналог закладок редактора Word) фрагменту документа, на який встановлюється внутрішнє посилання;
- target="..." – встановлює параметри відкриття документа. Має наступні значення: \_self завантажує сторінку в поточному вікні браузера; \_blank – в новому вікні; \_parent – у батьківський фрейм; \_top – відмінляє всі фрейми і

завантажує сторінку в повному вікні браузера.

### Теги вставки графічних елементів

<HR> – задає горизонтальну лінію. Можливі атрибути перераховані нижче:

- width="..." – довжина лінії в пікселях чи у відсотках від ширини вікна;
- size="..." – ширина лінії;
- color="..." – колір лінії;
- noshade – відмінняє рельєфність лінії;
- align="..." – положення лінії на рядку екрану. Значення: left – притиснути лінію до лівого краю, center – розташувати по центру, right – притиснути до правого краю.

<IMG> – вставляє зображення на Web-сторінку. Атрибути:

- src="..." – URL-адреса файлу зображення;
- alt="..." – альтернативний текст, який виводиться замість малюнка, якщо малюнок не завантажився;
- height="..." і width="..." – встановлюють висоту і відповідно ширину зображення;
- border="..." – задає ширину рамки навколо зображення;
- vspace="..." і hspace="..." – визначають відстань по вертикалі і по горизонталі від зображення до краю тексту;
- align="..." – задає положення тексту щодо зображення.

<MARQUEE>...</MARQUEE> – контейнер рядка, що біжить. Текст, укладений в нього, переміщується по сторінці відповідно до наступних атрибутів:

- width="..." і height="..." – ширина і висота рядка, що біжить. Задається в пікселях, або у відсотках від ширини і висоти вікна;
- align="..." – задає положення зображення даного рядка по вертикалі;
- bgcolor="..." – встановлює фоновий колір рядка;
- behavior="..." – задає тип руху. Значення: scroll, slide, alternate;
- direction="..." – задає напрям руху. Значення: left (справа наліво), right (зліва направо);
- scrollamount="..." і scrolldelay="..." – задають швидкість переміщення: число пікселів, на яке повинне переміститися зображення за задане число мілісекунд;
- loop="..." – задає число повторів анімації. За умовчанням встановлений режим безперервного повтору.

### Теги створення табличних документів

<CAPTION> – контейнер заголовка таблиці. Має наступні атрибути: align="..." – задає місце заголовка, height="..." та width="..." – задає висоту та ширину блоку, в якому розташовується заголовок таблиці.

<TABLE>...</TABLE> – контейнер, в якому міститься таблиця. Має наступні атрибути:

- width="..." і height="..." – ширина і висота таблиці в пікселях, або у відсотках від ширини і висоти вікна;
- border="..." – ширина рамки. Якщо значення рівне нулю, то рамка не відображається;

- `cellpadding="..."` і `cellspacing="..."` – додають вільний простір між даними клітини і її межами, а також між елементами таблиці. Якщо рамка відсутня, то відступ від межі буде рівний сумі цих значень;
  - `align="..."` – задає положення даних в таблиці по горизонталі.
- `<TD>...</TD>` – контейнер елемента таблиці. Задані атрибути клітини пригнічують атрибути рядка і таблиці. Атрибути:
- `width="..."` і `height="..."` – ширина і висота в пік селях, або у відсотках від ширини і висоти вікна;
  - `align="..."` – встановлює положення даних по горизонталі;
  - `valign="..."` – встановлює положення даних по вертикалі, значення: `bottom` – вирівняти по нижньому краю, `middle` – розташувати по центру; `top` – притиснути до верхнього краю клітини;
  - `bgcolor="..."` – задає колір фону;
  - `bordercolor="..."` – визначає колір рамки клітини;
  - `colspan="..."` – задає число клітин, що об'єднуються в одну по рядку;
  - `rowspan="..."` – встановлює число клітин, що об'єднуються в одну по стовпцю;
  - `nowrap` – відміна перенесення слів на інший рядок усередині клітини.
- `<TH>...</TH>` – контейнер визначення заголовка стовпця чи рядка. За умовчанням текст показується центрованим жирним шрифтом. Контейнер має такі ж атрибути, як і `<TD>` (див. вище).
- `<TR>...</TR>` – контейнер рядка таблиці. Якщо контейнер порожній, то рядок залишається порожнім. Задані атрибути рядка мають більший пріоритет, ніж атрибути таблиці. Має такі ж атрибути, як і `<TD>`

### Теги розробки фреймів

- `<FRAME>...</FRAME>` – контейнер опису окремого фрейма. Число таких тегів повинне бути рівне числу зарезервованих кадрів в контейнері (за винятком вкладених контейнерів). Розташовуються кадри послідовно – зліва направо і зверху вниз. Тег має наступні атрибути:
- `src="..."` – URL-адреса вмісту кадру. Звично це файл HTML-документа з того ж каталогу, що і сам контейнер, але може бути і абсолютна адреса файлу з будь-якого комп'ютера. Якщо атрибут не заданий, кадр буде порожнім;
  - `marginwidth="..."` і `marginheight="..."` – ширина від межі кадру до рамки по вертикалі і горизонталі;
  - `scrolling="..."` – визначає смуги прокрутки. За умовчанням смуги прокрутки додаються в кадр, якщо вміст кадру більше його розмірів. Значення: `yes` – смуги прокрутки будуть обов'язково, `no` – смуги прокрутки заборонені, `auto` – смуги прокрутки додаються за умовчанням;
  - `noresize` – забороняє користувачам змінювати розміри кадру. За умовчанням, якщо захопити рамку кадру мишею, з'являється двонаправлена стрілка, за допомогою якої можна перемістити рамку;
  - `frameborder="..."` – задає тип рамки кадру. Значення: `yes` – рамка має об'ємну форму, `no` – рамка пласка;

- bordercolor="..." – задає колір рамки кадру;
- name="..." – ім'я кадру, показує, в який кадр слід виконувати завантаження при посиланні.

<FRAMESET>...</FRAMESET> – контейнер фреймів, тіло Web-сторінки у вигляді кадрів. Має наступні основні атрибути:

- rows="...,..." – список значень висоти рядків зверху вниз;
- cols="...,..." – список значень ширини стовпців зліва направо;
- frameborder="..." – задає тип рамок між кадрами контейнера. Значення: no – рамки невидимі, тобто мають колір фону вікна браузера; yes – рамки кольорові;
- bordercolor="..." – задає колір рамок між кадрами. За умовчанням співпадає з кольором рамки вікна;
- border="..." – встановлює ширину рамок між кадрами контейнера. Якщо значення відсутнє, то кадри відділяються один від одного тонкими лініями заданого кольору;
- framespacing="..." – відстань між кадрами.

<NOFRAMES>...</NOFRAMES> – вкладений текст відобразатиметься тільки в тому випадку, якщо браузер не підтримує механізм фреймів. Браузери, що підтримують кадри, ігнорують цей контейнер.

### 7.2.2. Спеціальні символи і колір

Окрім тегів у мові HTML використовується спеціальний управляючий символ & – амперсant. Він застосовується для виведення спеціальних символів і символів з розширеної кодової таблиці, які не можна ввести з клавіатури. Наприклад, виведення самого символу амперсant (&) здійснюється за допомогою послідовності символів &amp;, для виведення кутових дужок використовують послідовності &lt; (для "<") і &gt; (для ">"). У свою чергу, символ із заданим номером з кодової таблиці (наприклад 182) може бути заданий послідовністю &#182; .

Найбільш поширені знаки, та їх коди приведені в наступній таблиці.

Символ	Код	Приклад
Знак авторських прав	&copy;	Copyright ©
Зареєстрована торгова марка	&red;	МММ®
Торгова марка	&#8482;	Company™
Менше	&lt;	<
Більше	&gt;;	>
Амперсant	&amp;	&
Нерозривний пропуск	&nbsp;	
Тире	&#8212;	—
Лапка	&quot;	„

Слід відмітити, що специфікація HTML 4.0 визначає код &trade; для знаку торгової марки, проте, він ще не настільки поширений, як &#8482;.

Вибір кольору можна проводити двома способами: завданням імені чи ви-

значенням номера кольору по колірній схемі RGB (Red, Green, Blue).

Мова HTML підтримує наступні імена кольорів: AQUA, BLACK, BLUE, FUCHSIA, GRAY, GREEN, LIME, MAROON, NAVY, OLIVE, PURPLE, RED, SILVER, TEAL, WHITE, YELLOW.

Номер кольору RGB задається трьома двозначними шістнадцятирічними числами, причому кожне число належить інтервалу від 00 до FF і визначає інтенсивність відповідного кольору. Наприклад, номер кольору #FF0000 відповідає червоному кольору, оскільки має максимальну інтенсивність для червоного кольору, а зелений і блакитний мають значення, рівні нулю. Відповідно, номер #00FF00 кодує зелений колір, а номер #0000FF – синій. Білий колір утворюється в результаті насичення всіх основних кольорів, тобто його кодом буде число #FFFFFF.

У разі повної відсутності всіх трьох кольорів можна одержати абсолютно чорний колір (код #000000). Очевидно, що, використовуючи таку комбінацію шістнадцятирічних чисел, можна одержати дуже велику палітру.

Для зручності, основні шістнадцять кольорів, їх коди та назви приведені в таблиці.

Колір	Код RGB	Ім'я	Колір	Код RGB	Ім'я
чорний	#000000	black	срібний	#C0C0C0	silver
темно-бордовий	#800000	maroon	червоний	#FF0000	red
зелений	#008000	green	лимонний	#00FF00	lime
оливковий	#808000	olive	жовтий	#FFFF00	yellow
темно-синій	#000080	navy	синій	#0000FF	blue
фіолетовий	#800080	purple	фуксія	#FF00FF	fuchsia
чирок	#008080	teal	блакитний	#00FFFF	aqua
сірий	#808080	gray	білий	#FFFFFF	white

### 7.2.3. Приклади Web-сторінок

Наведемо декілька прикладів, які пояснюють роботу різних тегів по представленню тексту файлів у HTML-кодах та їх вигляд у браузері.

Перший приклад (рис. 7.2) стосується заголовків різних розмірів від H1 до H7. Можуть використовуватися для структурування тексту документу через багаторівневі заголовки. Далі показано формування маркірованих (тег UL) та нумерованих (тег OL) списків. Кожен елемент списку починається тегом LI. Тег P переводить текст на наступний рядок, B – робить шрифт жирним, I – курсивом, U – підкреслює текст, який в ньому знаходиться. Можна всі ці теги розташувати один в одному, але потрібно уважно ставитись до порядку їх закриття. Для виділення абзаців можна використовувати дві основних теги – P чи DIV. Перший, використовує американський метод форматування, відділяючи абзаци пустим рядком. DIV – більше орієнтований на звичне українське форматування.

Тег PRE дозволяє вставляти текст, який було відформатовано раніше яки-



мось текстовим редактором. Це буває корисним, наприклад, при цитуванні віршів або статей, розташування тексту в яких має принципове значення.

```
<HTML>
<HEAD>
<TITLE> Приклад заголовків, спи-
сків, параграфів </TITLE>
</HEAD>
<BODY>
<H1> Заголовок1 </H1>
<H2> Заголовок2 </H2>
<H3> Заголовок3 </H3>
<H4> Заголовок4 </H4>
<H5> Заголовок5 </H5>
<H6> Заголовок6 </H6>
<UL>
<LI> Рядок 1 списку
<LI> Рядок 2 списку
<LI> Рядок 3 списку
</UL>
<OL>
<LI> Рядок 1 списку
<LI> Рядок 2 списку
<LI> Рядок 3 списку
</OL>
<P> Текст <B>Жирний</B>
<I>Курсив</I>
<U>Підкреслений</U></P>
<PRE>
Тек ст нефор
матова ний
</PRE>
</BODY>
</HTML>
```

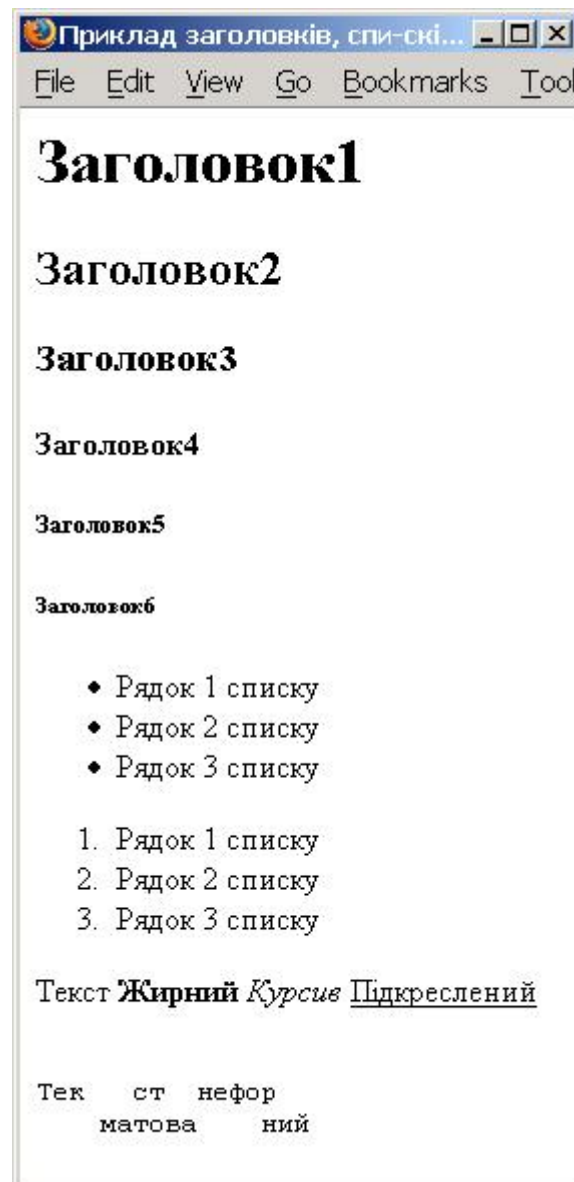


Рис. 7.2. Приклад, заголовків, списків та параграфів

Одним з основних елементів web-сторінок є таблиці. Наведемо приклад складної таблиці з вирівнюванням елементів різними стилями.

Тег TH дає заголовок таблиці, теги TR і TD формують рядки і колонки таблиці. Режим тега TH – COLSPAN – забезпечує використання одразу двох колонок таблиці під заголовком. Режим тега TD – ROWSPAN – забезпечує використання одразу двох рядків під клітинку таблиці.

Цей код дає наступне зображення (рис. 7.3).

```

<HTML>
<HEAD>
<TITLE>      Приклад      таблиці
</TITLE>
</HEAD>
<BODY>
<TABLE      BORDER      =”1”
CELLPADDING= “3” BGCOLOR=
“white” >
<TH> Мова програмування </TH>
<TH ALIGN =”right” COLSPAN
=”2”> Особливості </TH>
<TR>
<TD> C++ </TD>
<TD ROWSPAN = “2”>
Використовує </TD>
<TD> Класи </TD>
<TR>
<TD> Visual BASIC </TD>
<TD> Об`єкти </TD>
</TABLE>
</BODY>
</HTML>

```

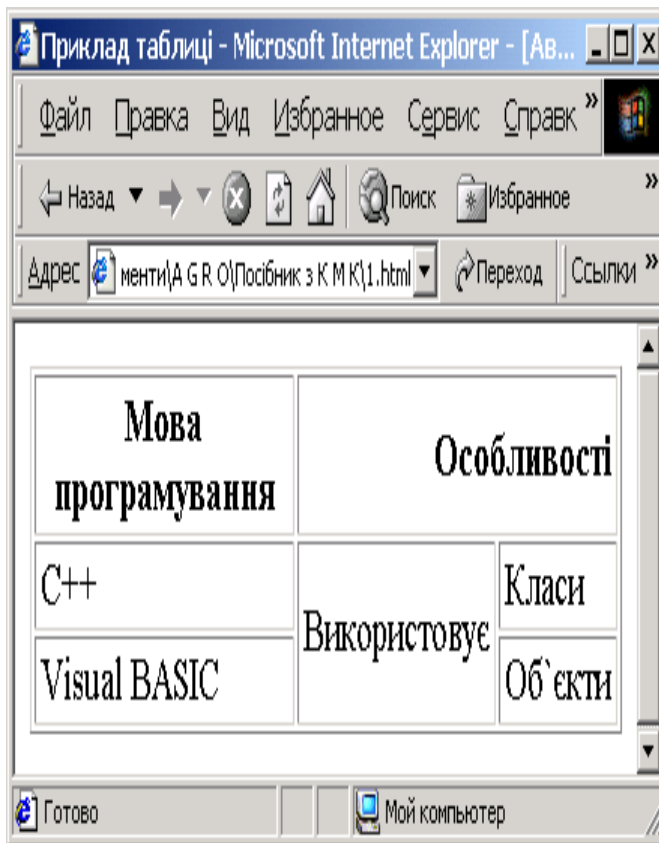


Рис. 7.3. Приклад зображення таблиці

Фрейми (frames – кадри) розбивають Web-сторінку на незалежні вікна, що містять власну інформацію. В кожне з цих вікон завантажуються окремі HTML-документи.

Для створення фрейму, як уже зазначалося, використовується елемент <FRAMESET>, який заміняє елемент <BODY> в документі і використовується для розділу екрану на області-фрейми.

Всередині елемента знаходяться елементи <FRAME>, які посилаються на HTML-документ, призначений для завантаження в область фрейма.

Об`єднаємо попередні приклади в загальну структуру (рис. 7.5).

Стартова сторінка буде мати називатися index.html. Зліва на сторінці, розміщується меню, тобто в нашому випадку це назви трьох прикладів, справа – виводиться вміст вибраного документа.

### 7.3. Таблиці стилів CSS

CSS (Cascading Style Sheets – каскадні листи стилів) є інструментом впливу на зовнішній вигляд Web-сторінок.

Стилі – це набір властивостей, які можна привласнити багатьом елементам (таким як таблиці, текст посилання і т.п.), і які впливають на відображення

об'єкта у вікні браузера.

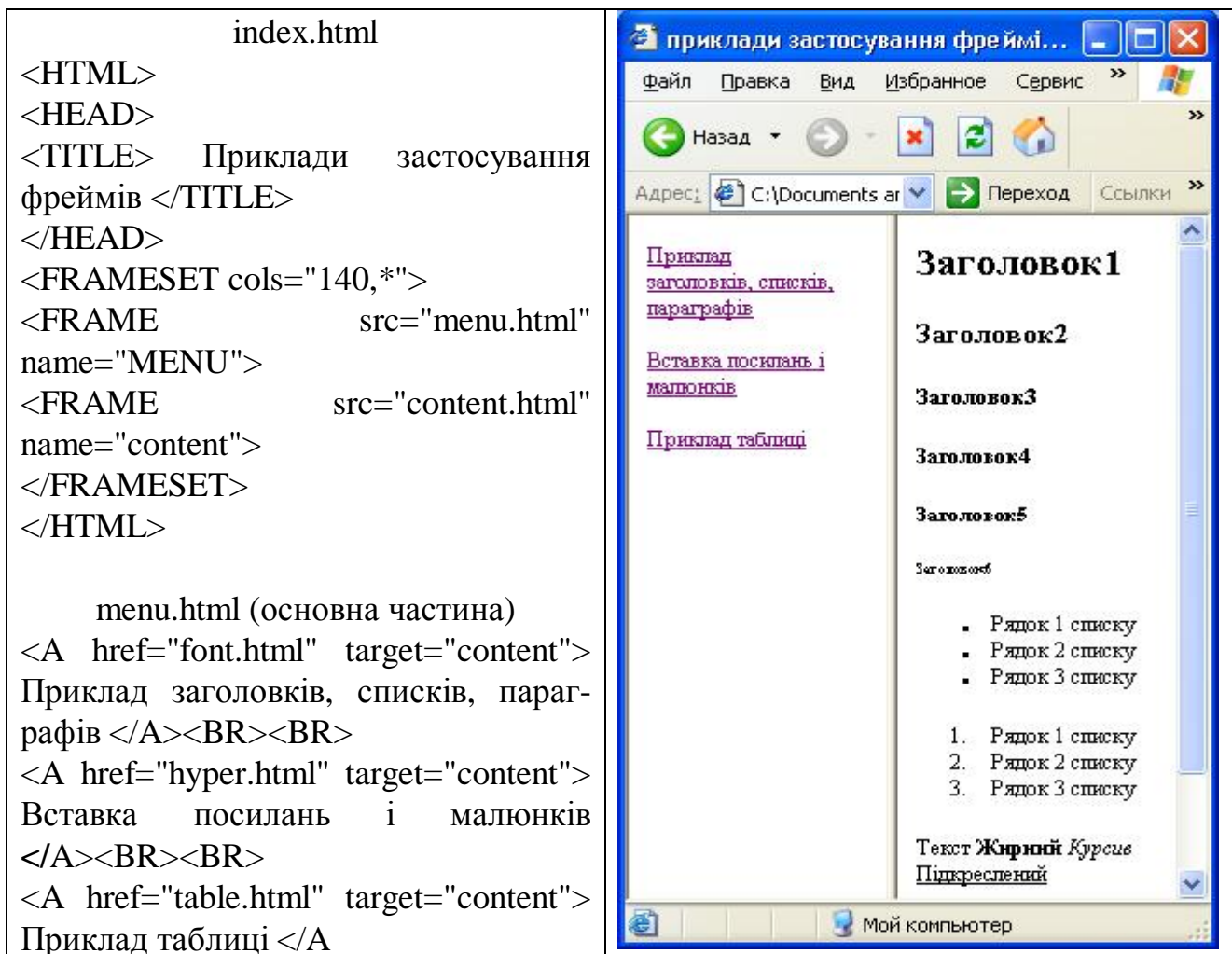


Рис 7.4. Приклад застосування фреймів

### 7.3.1. Базові поняття. Специфікація CSS

Розробник повинен мати базові знання по HTML і по поліграфічній термінології. Синтаксис правила стиля CSS в найбільш загальному вигляді має наступну форму:

селектор { визначення }

Визначенням стилю називається пара значень „властивість: значення”.

Наприклад, для визначення кольору елементів 'H1' як синій, достатньо написати:

H1 { color: blue }

Селектор є зв'язком між HTML документом і таблицею стилів, повний набір типів елементів містить всі можливі селектори. Типи елементів визначені в специфікації HTML.

Авторам Web-сторінок необхідно визначати свою таблицю стилів тільки в тому випадку, якщо вони хочуть запропонувати специфічний стиль для своїх документів.

Текстові коментарі в таблицях стилів оформляються так:

```
EM { color: red } /* червоне, справді червоне!! */
```

Коментарі не можуть вкладатися один в одного. Для обробника CSS коментар еквівалентний пропуску.

### 7.3.2. Включення в HTML

Для того, щоб таблиця стилів впливала на вигляд документа, браузер повинен знати про її існування. Специфікація HTML визначає способи включення таблиць стилів в HTML. Тому даний розділ є інформативним, а не нормативним:

```
<HTML>
  <HEAD>
    <TITLE>title</TITLE>
    <LINK REL=STYLESHEET TYPE="text/css"
      HREF="http://style.com/cool" TITLE="Cool">
    <STYLE TYPE="text/css">
      @import url(http://style.com/basic);
      H1 { color: blue }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>Headline is blue</H1>
    <P STYLE="color: green">While the paragraph is green.
  </BODY> </HTML>
```

Даний приклад демонструє чотири способи об'єднання стилю з HTML:

- використовуючи елемент LINK для зв'язку із зовнішньою таблицею стилів, посилається на альтернативну таблицю стилів, яку може вибрати читач, тоді як таблиці стилів, що імпортуються, автоматично об'єднуються з рештою таблиць стилів;
- використовуючи елемент STYLE усередині елементу HEAD;
- імпортуючи таблицю стилів за допомогою нотації CSS @import;
- використовуючи атрибут STYLE в елементі усередині секції BODY. Цей спосіб змішує стиль з вмістом і тому втрачає відповідні достоїнства традиційних таблиць стилів.

Традиційно браузери ігнорували невідомі теги. Тому старі ігноруватимуть елемент STYLE, але його вміст вважатиметься, частиною тіла документа і відображатиметься в документі. Протягом перехідного періоду, вміст елементу STYLE можна "ховати", використовуючи коментарі SGML:

```
<STYLE TYPE="text/css"><!-- H1 { color: green } --></STYLE>
```

### 7.3.3. Групування

Для зменшення розміру таблиці стилів, селектори можна групувати в списки, розділені комами:

```
H1, H2, H3 { font-family: helvetica }
```

Точно також можна групувати визначення:

```
H1 { font-weight: bold;
      font-size: 12pt;
      line-height: 14pt;
      font-family: helvetica;
      font-variant: normal;
      font-style: normal; }
```

Зазначимо, що деякі властивості мають власний синтаксис групування:

```
H1 { font: bold 12pt/14pt helvetica }
```

що еквівалентне попередньому прикладу.

#### 7.3.4. Успадкування

Уявіть, що є елемент H1 має всередині елементом <EM>:

```
<H1>The headline <EM>is</EM> important!</H1>
```

Якщо для елемента EM не було визначено ніякого кольору, то параметр "is" успадкує колір батьківського елемента, тобто буде відображений синім кольором. Інші властивості стилю також спадкуються, наприклад 'font-family' і 'font-size'.

Для визначення властивості стилю для документа за умовчанням, можна задати цю властивість елементу, від якого успадковується решта видимих елементів. В HTML-документах цю функцію виконує елемент BODY:

```
BODY { color: black;
        background: url(texture.gif) white; }
```

Цей приклад спрацює, навіть якщо автор опустив тег BODY (що є допустимим) тому, що браузер самостійно вставить пропущений тег. Попередній приклад встановлює чорний колір тексту, а фон – містить картинку. Якщо рисунок буде недоступний, фон залишиться білим.

Інколи значення властивості вказується у відсотках від іншої властивості:

```
P { font-size: 10pt }
P { line-height: 120% } /* relative to 'font-size', i.e. 12pt */
```

Для будь-якої властивості, значення якої можуть вказуватися у відсотках, визначено властивість на яку вона посилається. Дочірні елементи P успадкують обчислене, а не процентне значення 'line-height' (12pt).

#### 7.3.5. Клас як селектор

Для збільшення гнучкості контролю над елементами, в HTML доданий новий атрибут CLASS. Всі теги усередині елемента BODY можуть бути класифіковані, а на клас можна послатися через таблицю стилів:

```
<HTML> <HEAD>
      <TITLE>Title</TITLE>
      <STYLE TYPE="text/css"> H1.pastoral { color: #00FF00 }
      </STYLE>
```

```

</HEAD>
<BODY> <H1 CLASS=pastoral>Way too green</H1>
</BODY> </HTML>

```

До класифікованих елементів застосовуються звичайні правила спадкоємства, вони успадковують значення своїх батьків в структурі документа.

Можна адресувати всі елементи одного класу опустивши ім'я тега в селекторі:

```
.pastoral { color: green } /* all елементи with CLASS pastoral */
```

Для кожного селектора можна визначити тільки один клас. Тому `P.pastoral.marine` неправильний селектор в CSS. Контекстові селектори можуть визначати один клас для одного простого селектора.

### 7.3.6. ID як селектор

HTML визначає атрибут ID, який має унікальне значення в документі. Він важливий, як селектор таблиці стилів, і може адресуватися за допомогою '#':

```

#z98y { letter-spacing: 0.3em }
H1#z98y { letter-spacing: 0.5em }
<P ID=z98y>Wide text</P>

```

У приведеному прикладі перший селектор відповідає елементу P завдяки ID-атрибуту. Другий селектор визначає як тип елементу (H1), так і значення ID, і, тому, не відповідає елементу P.

Використовуючи ID як селектор, можна встановлювати властивості по-елементно. Тоді як таблиці стилів були розроблені для візуалізації структури документа, ця властивість дозволяє авторам створювати документи, які коректно відображаються без використання структурних елементів HTML. Але таке застосування таблиць стилів не вітається.

### 7.3.7. Основні параметри таблиці стилів

ВЛАСТИВОСТІ ШРИФТУ	
font-family	Використовується для вказівки шрифту або шрифтового сімейства, яким відобразатиметься елемент. P { font-family: Times New Roman, sans-serif; }
font-weight	Визначає ступінь жирності шрифту за допомогою трьох параметрів: lighter, bold, bolder B { font-weight: bolder; }
font-size	Встановлює розмір шрифту. Параметр може указуватися як у відносній (відсотки), так і абсолютній величині (пункти, піксели, сантиметри) H1 { font-size: 200%; } H2 { font-size: 150px; } H3 { font-size: 400pt; }
font-size	Встановлює розмір шрифту. Параметр може указуватися як у відносній (відсотки), так і абсолютній величині (пункти, піксели, сантимет-

	ри) H1 {font-size:200%;} H2 {font-size:150px;} H3 {font-size: 400pt;}
КОЛІР ЕЛЕМЕНТУ І КОЛІР ФОНУ	
color	Визначає колір елемента I {color: yellow;}
background-color	Встановлює колір фону для елемента – саме для елемента, а не для сторінки. Зверніть увагу, що браузері відображають цю властивість по-різному: Microsoft IE відводить під фон елемента всю доступну ширину сторінки, а Netscape Navigator – лише ширину, займану цим елементом.
ВЛАСТИВОСТІ ТЕКСТУ	
text-decoration	Встановлює ефекти оформлення шрифту, такі, як підкреслення або закреслюваний текст H4 {text-decoration: underline;} A {text-decoration: none;} .wrong {text-decoration: line-through;}
text-align	Визначає вирівнювання елемента. P {text-align: justify} H1 {text-align: center}
text-indent	Встановлює відступ першого рядка тексту. Найчастіше використовується для створення параграфів з табульованим першим рядком. P {text-indent: 50pt;}
line-height	Управляє інтервалами між рядками тексту. P {line-height: 50 % }
ВЛАСТИВОСТІ МЕЖ	
margin-left	Встановлюють значення відступів навколо елемента. IMG { margin-right: 20pt } P { margin-left: 2cm }
margin-right margin-right margin-top	Встановлюють значення відступів навколо елемента. IMG { margin-right: 20pt } P { margin-left: 2cm }
ОДИНИЦІ ВИМІРЮВАННЯ	
px	Пікселі
cm	Сантиметри
mm	Міліметри
pt	Пункти (друкар.)
%	Відсотки

## 7.4. JavaScript

Розроблена в 1995 р. фірмою Netscape для версії 2.0 свого браузера мова JavaScript дотепер залишається допоміжною, але в той же час є абсолютно незамінним інструментом, що дозволяє завантаженій в браузер сторінці динамічно управляти своїм вмістом, а разом і власне браузером. По своєму набору функцій ця мова близька до макромов, які з давніх пір вбудовуються в будь-яку достатньо складну програму або систему програм.

Як правило, для скриптів (script – сценарій) використовується дві основні мови програмування – JavaScript та VBScript.

JavaScript-сценарії вільно переплітаються і взаємодіють з HTML-розміткою сторінки. Вони дозволяють відкривати і закривати вікна браузера, завантажувати в них документи, управляти фреймами і взаємодіяти з полями форм (наприклад, перевіряючи правильність введених в них значень).

### 7.4.1. Розміщення скриптів

Сценарій, вбудований в документ за допомогою тега SCRIPT, може вставляти блоки HTML-коду в те місце документа, в якому розташований сам. Стала можливою плавна зміна кольору фону при завантаженні сторінки або «живі» меню, кожен пункт яких змінюється, коли над ним проводиш мишею.

Скрипти можуть розміщуватися в заголовку чи тілі HTML-документа. Їх місцезнаходження часто не впливає на роботу програми. Але ті, які повинні виконуватись в першу чергу, розміщують в заголовку.

Тег SCRIPT вставляється всередину тега HEAD у такому порядку:

```
<HEAD>
  <SCRIPT language="JavaScript">
    Команди мови JavaScript
  </SCRIPT>
<TITLE> Вміст заголовка </TITLE>
</HEAD>
```

При цьому, дія команд JavaScript розповсюджується на всю web-сторінку. Зручно підключати зовнішні сценарії, написані на JavaScript.

```
<SCRIPT type="text/javascript" src="http://server.com/progs/script.js">
</SCRIPT>
```

В цьому випадку немає необхідності добавляти в контейнер <SCRIPT> вміст – на сторінку завантажиться сценарій script.js з вказаного місця. Ця можливість дуже зручна при використанні стандартних скриптів та цілих бібліотек.

Основні способи виконання скриптів – через виклик функції чи при настанні певної події (при завантаженні сторінки, натисканні кнопки і т.п.

### 7.4.2. Приклади застосування JavaScript

JavaScript має велику кількість команд і свій синтаксис. Наведемо декілька



прикладів застосування цієї мови.

### **Функції затримки часу**

Деколи в програмі потрібно створити певну затримку часу. Стандартна функція JavaScript `setTimeout()`; іноді не підходить, оскільки "на її фоні" можуть виконуватися інші функції, що може привести до небажаних накладок. Коректніший результат дає наступна функція:

```
function pause (mSec) { clock = new Date();
    justMinute = clock.getTime();
    while (true) { just = new Date();
        if (just.getTime() - justMinute > mSec) break; } }
```

### **Функції відкриття вікон**

`window.open("URL", "windowName", ["windowFeatures,..."])` де `windowFeatures` - це:

- `copyhistory [=yes/no] / [=1/0]` – збереження історії завантаження документів у дане вікно;
- `directories [=yes/no] / [=1/0]` – наявність в даному вікні кнопок груп новин;
- `height =pixelheight` висота вікна в пікселях;
- `location [=yes/no] / [=1/0]` наявність поля;
- `locationmenubar [=yes/no] / [=1/0]` наявність меню;
- `resizable [=yes/no] / [=1/0]` наявність рамки вікна, яка дозволяє змінювати його розміри;
- `scrollbars [=yes/no] / [=1/0]` наявність лінійок прокрутки;
- `status [=yes/no] / [=1/0]` наявність рядка стану;
- `toolbar [=yes/no] / [=1/0]` наявність панелі інструментів;
- `width =pixelwidth` ширина вікна в пікселях.

Використовувати пропуски в рядку `windowFeatures` не допускається.

## **7.5. Програми для створення html-сторінок**


Як вже зазначалося, документ, складений мовою розмітки HTML представляє собою текстовий файл, який можна створювати і редагувати в найпростіших текстових додатках наприклад в Блокноті чи редакторі WordPad. Однак, нині існують більш розвинені і зручні програмні засоби. Їх умовно можна розділити на візуальні HTML-редактори і професійні редактори HTML-коду.

### **7.5.1. Візуальні редактори html-сторінок**

У Web-редакторах типу WYSIWYG (What You See Is What You Get – „що бачите, те і отримаєте”) користувач має справу не з кодом документа, а з графічними образами елементів HTML. Тобто створює не код, а оформлення сторінки, після чого програма автоматично підбирає для неї код. В результаті на сторінці часто виявляється багато зайвого коду. Він може залишатися, наприклад, від скасованих проб або коментарів, які програма вставляє сама, що безумовно сповільняє

завантаження сторінки. До редакторів цього типу можна віднести: Microsoft FrontPage, Macromedia Dreamweaver, Netscape Composer, Star Office та ін.

Найпростішим способом створення web-сторінки є застосування текстового редактора Word. Спочатку засобами Word потрібно створити текст, включаючи, за необхідності малюнки, таблицю та різноманітні прийоми форматування. При цьому в головному меню по пункту „Вид” потрібно вибрати „Веб-документ”. Тоді ваша сторінка виглядатиме так, як у браузері.

Якщо потрібно створити посилання на якийсь файл чи адресу, потрібно відмітити фрагмент тексту чи малюнок і вибрати пункт меню „Вставка–Гіперссылка” або натиснути кнопку  на панелі інструментів. Вікно, що відкріється (рис. 7.5) забезпечить вас можливістю вказати файл або адресу, на яку потрібно зробити перехід.

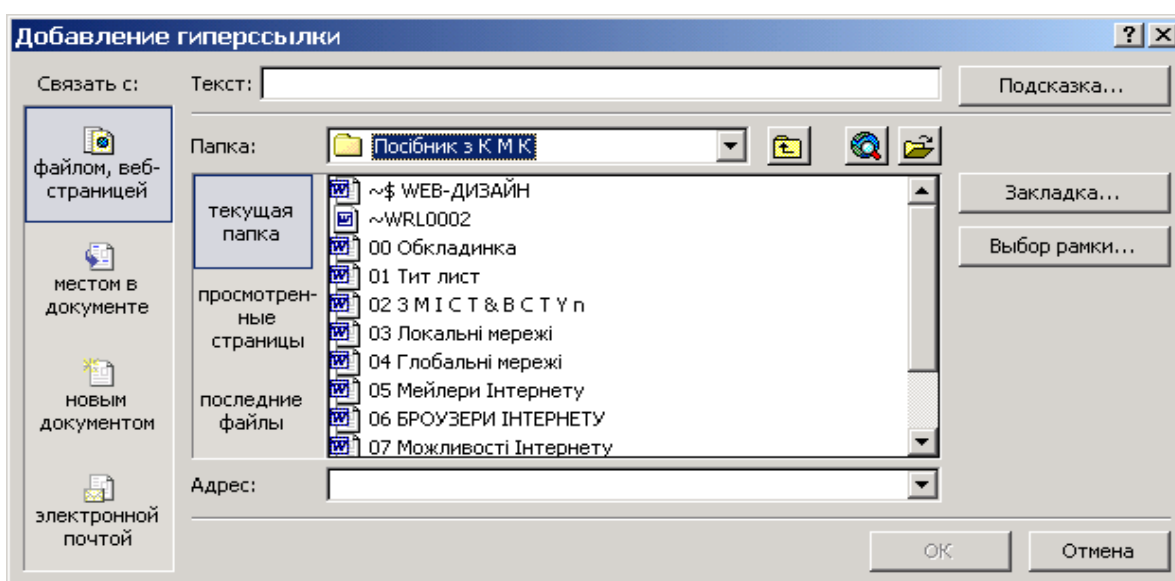


Рис. 7.5. Вікно додавання гіперпосилання у текстовому редакторі Word

Після закінчення редагування потрібно обрати пункт меню „Сохранить как...” і вікні файл-менеджера, що відкріється обрати тип файлу – „Веб-сторінка”. При збереженні, Word самостійно згенерує HTML-код, згідно вказаного Вами форматування. За допомогою Майстра Веб-сторінок можна створити навіть сайт з декількох сторінок.

Файли, створені таким чином часто мають надмірно великий обсяг і не завжди працюють так, як планувалося. Такий спосіб створення можна використовувати хіба що при створенні найпростіших персональних сторінок.

Зручніше скористатися можливостями браузера Opera. Якщо вибрати пункт меню “View-Source”, то зображення web-сторінки зміниться на її текст, який можна редагувати так само, як Блокноті (рис. 7.6).

Повернення до зображення здійснюється через повторне відкриття того ж файлу, але перед цим треба зберегти зміни у тексті сторінки.

Розглянемо створення веб-сторінок за допомогою програми Page Composer, яка є елементом NetScape Communicator`а. За структурою він дуже схожий на Word.

Меню **Files/Publish** призначене для відправки створеної вами html-сторінки на той сервер, який надає послуги по підтримці вашої сторінки в Інтернеті.

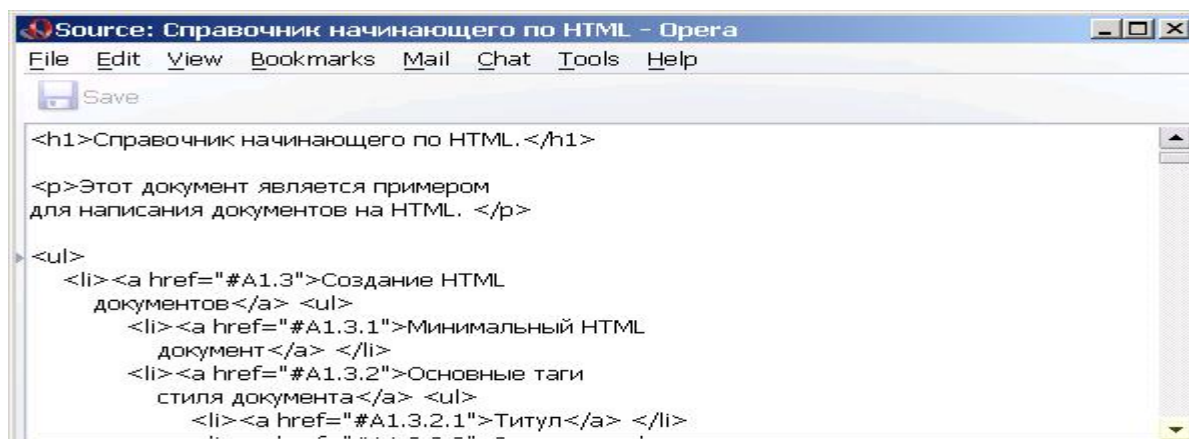


Рис. 7.6. Приклад редактора тегів броузера Opera

**Insert/Link** дозволяє вставити гіперпосилання, так само, як і у Word`і на якусь Інтернет-адресу. **Insert/Target** дозволяє встановити зв'язок на інший елемент тієї самої сторінки. **Insert/Image**, **Horisontal Line**, **HTML Tag** – це вставка відповідно малюнка, горизонтальної лінії та якоїсь HTML-тегу. **Insert/New Line Break**, **Break Below Image** забезпечують вставку лінії розриву та обтікання текстом малюнка.

Починаючи створювати нову HTML-сторінку, треба вибрати її тип. Це буде чиста сторінка, сторінка з готових шаблонів, сторінка з майстра підказок чи сторінка з якогось файлу.

Далі, обов'язково треба вказати, де знаходитиметься файл з нашою сторінкою. Цю інформацію дає провайдер. Бажано доступ до сторінки закривати паролем, щоб ніхто її без вас не зміг змінити. Обов'язковим є вказування всіх файлів, які були приєднані до вашої сторінки (часто це малюнки), інакше в Інтернеті їх не буде видно.

Є ще один нескладний редактор веб-сторінок Microsoft FrontPage (рис. 7.9). Розберемо його детальніше.

Редактор містить багато шаблонів графічного оформлення Web-сторінок, різноманітні приклади готових сайтів, колекцію розроблених інтерактивних елементів (лічильники, кнопки і т.п.), має засоби роботи з базами даних і інструменти для створення програмних анімацій

Так само, як у документах Word, у Web-сторінках можна створювати маркування списків. Основна відмінність полягає в тому, що крім маркерів у списках можна використовувати графічні зображення.

Набір таких зображень знаходиться в діалоговому вікні **Список** (меню **Формат**). Крім графічних зображень, у діалоговому вікні пропонуються текстові маркери, підтримувані HTML для Web-сторінок. Для перегляду всього списку можливих графічних зображень натисніть кнопку **Другие**. Виберіть придатне зображення і натисніть кнопку **ОК**, щоб повернутися в документ.

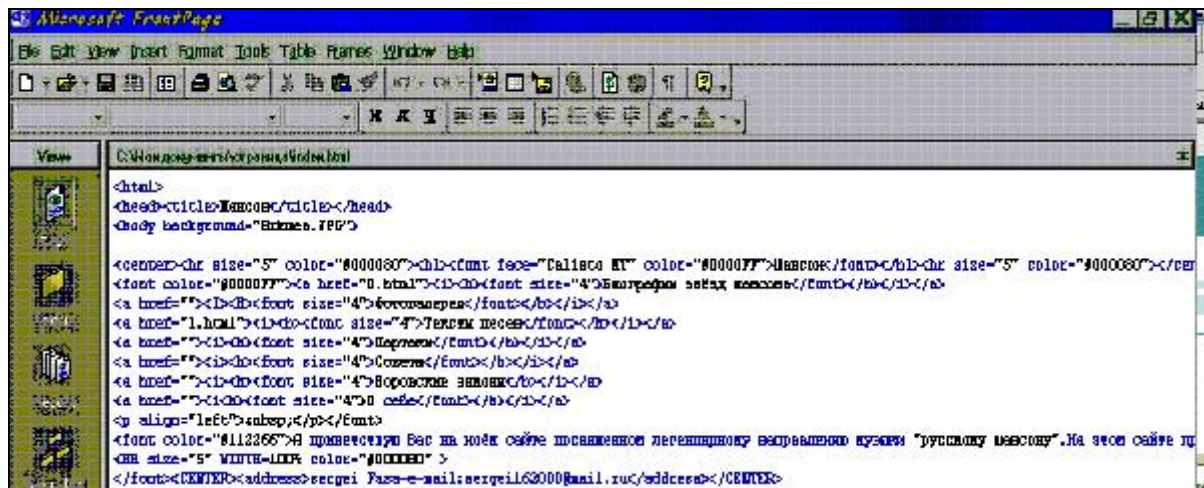


Рис. 7.7. Інтерфейс програми Microsoft Front Page

Нумерація списків на Web-сторінках майже не відрізняється від нумерації в документах Word. Відмінність полягає в тому, що для Web-сторінок неможлива автоматична нумерація структурних списків і заголовків. Однак щоб створити список з декількома рівнями, можна застосувати кілька стилів нумерації і відступів.

Для поділу тексту на частини на Web-сторінках часто використовуються горизонтальні лінії. Для їх вставки потрібно виконати наступні дії:

1. Клацніть те місце, де передбачається розмістити лінію.
2. Виберіть команду **Горизонтальна лінія** в меню **Вставка**.
3. Зі списку **Вид** виберіть потрібну лінію або натисніть кнопку **Другие** і виберіть іншу лінію.

У Web-сторінку можна також уставити рядок, що біжить.

1. Виберіть команду **Бегущая строка**, у меню **Вставка**.
2. Уведіть текст рядка, що біжить, у поле **Ввести текст бегущей строки**.
3. Виберіть інші параметри.

### 7.5.2. Редактори html-кодів

Серед редакторів html-текстів найбільш відомі Hometown та HotDog.

Використання цих програм передбачає знання користувачем мови HTML (а також, за бажанням, JavaScript і ін.). Вони автоматизують введення коду, перевіряють помилки і мають безліч доступних функцій, зручних для користувача. Тому їх можна розглядати не тільки як могутній інструмент розробки Web-публікацій професійної якості, але і як засіб навчання технології Web-дизайну.

До основних функціональних можливостей Hometown 5 можна також віднести: швидкий перегляд сторінок у всіх браузерах, установлених на комп'ютері користувача; навігація по сторінках сайту, а також перегляд і навігація по ієрархії тегів усередині сторінки; перевірка коректності коду сторінки, а також контроль правопису текстового вмісту сторінок; автоматична перевірка коректності посилань у межах створюваного сайту; копіювання файлів проекту на віддалений сервер; підтримка технологій створення динамічних і інтерактивних сторінок.

нок; створення макросів і нових елементів інтерфейсу користувача засобами вбудованої мови програмування та ін.

Головне вікно HomeSite (рис. 7.8) надає користувачу, по суті, стандартний Windows-інтерфейс, що значно спрощує роботу з ним.

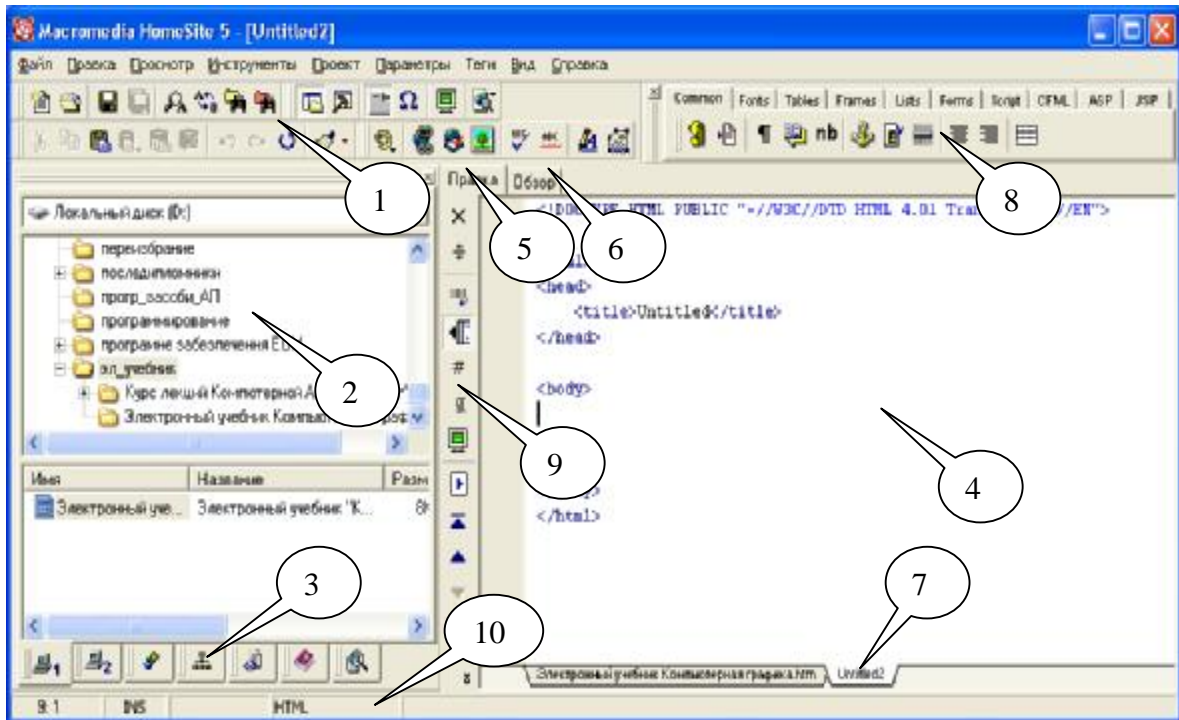


Рис. 7.8. Основні елементи інтерфейсу редактора HomeSite

Цифрами на рисунку показано наступні основні елементи:

1 – панелі інструментів, на них винесені найбільше часто застосовувані команди з усіх розділів меню, кожна кнопка має спливаючу підказку, управляти панелями можна за допомогою меню Вид;

2 – вікно ресурсів, назва досить умовна, оскільки в ньому може бути представлена найрізноманітніша інформація, наприклад, файлова структура проєкту, перелік розділів довідкової системи, результати пошуку і т.д., при цьому кожен вид даних відображається на окремому "аркуші", переключення між якими здійснюється за допомогою ярликів ресурсів;

3 – ярлики ресурсів;

4 – вікно документа – у цій позиції екрана може бути представлено або вікно редактора коду (вкладка Правка – 5), вікно браузера (Обзор – 6, використовується браузер, зазначений у параметрах настроювання HomeSite), чи вікно довідника (Help);

7 – ярлик документа – повідомляється ім'я файлу відкритого документа, кількість одночасно редагованих документів не обмежена;

8 – панель швидкої вставки тегів містить кнопки, що відповідають основним тегам мови HTML і елементам деяких інших мов, підтримуваних HomeSite, у залежності від типу зв'язаного з кнопкою тегу щиклик на ній приводить або до вставки тегу в текст документа, або до відкриття відповідного діалогового вікна

редактора тегів, за допомогою якого здійснюється введення необхідних параметрів; для зручності роботи кнопки панелі згруповані по призначенню, переключення між групами виконується за допомогою вкладок груп;

9 – панель текстового редактора, дозволяє налаштовувати вид робочого вікна, нумерацію рядків, перевірку коректності тегів і т.п.;


10 – рядок стану виводить додаткову інформацію про виконувані операції.

Як і в інших Windows-додатках, перелік доступних елементів керування (зокрема, кнопок панелей інструментів) визначається поточним станом редактора і властивостями активного об'єкта. Наприклад, вкладка Help з'являється на екрані тільки після першого звертання до довідкової системи.

У залежності від типу файлу і виду виконуваної операції HomeSite надає користувачу різні засоби для роботи з ним: на основі команд і/чи на основі прямого маніпулювання (drag-and-drop -і "перетягни і залиши"). Наприклад, щоб зберегти на диску текстовий файл, що редагується, варто вибрати в меню File команду Save, а щоб створити посилання на графічний файл, можна перетягнути мишею його значок з вікна ресурсів у вікно документа.

Можливості HomeSite виходять за рамки "звичайного" текстового редактора. Оскільки сучасну Web-публікацію складно уявити собі без графіки (а також інших мультимедійних елементів), без підтримки інтерактивності й інших сучасних Web-технологій, то в HomeSite передбачена можливість роботи з даними, представленими в близько 50 різних форматах.

Автоматична підстановка тегів, тобто, допомога пакету при введенні тегу і підборі його атрибутів, дозволяє досить швидко виконувати розмітку та уникати помилок при наборі.

Одним з найважливіших достоїнств HomeSite є те, що в ньому реалізований дуже зручний і ефективний механізм роботи з каскадними таблицями стилів. Цей механізм базується на використанні самостійного додатка TopStyle Lite (виробник - фірма Bradbure Software, LLC). Він встановлюється разом з HomeSite при згоді користувача. Виклик цього додатка виконується безпосередньо з текстового редактора натисканням кнопки  (редактор стилів) або вибором аналогічного пункту в меню **Інструменти**.

## 7.6. Публікація Web-вузла в Internet

Після закінчення створення HTML-документа, потрібно розмістити його в мережі Internet для забезпечення доступу до нього цільової групи користувачів. Попередньо потрібно виконати ще деяку підготовчу роботу:

1. Скопіювати всі файли в окрему папку і назвати її, наприклад, Website. У цій папці має бути файл index.html, що подає початкову сторінку сайту. Інші файли папки містять HTML-код інших сторінок (файли з розширенням \*.html), також в даній папці розміщують файли каскадних таблиць стилів (з розширенням \*.css). Імена усіх файлів повинні бути набрані малими латинськими буквами. У папці сайту бажано передбачити папку для розміщення графічних елементів у форматі GIF чи JPEG (наприклад, IMG).

2. Перевірити роботу сайту, за необхідності внести зміни в HTML-код.

3. Вибрати Web-сервер, на якому цей сайт буде розміщено. Найбільш поширеною програмою Web-серверу є Apache Server, яка працює швидко і встановлюється безкоштовно. Часто використовуються також програми TomCat, Microsoft IIS, NCSA та ін. За допомогою них свій Web-сервер може створити будь-яка організація і навіть приватна особа. Однак, це не завжди виправдано, оскільки повноцінний Web-сервер повинен мати швидке з'єднання з Internet'ом (наприклад, через виділену лінію) і працювати цілодобово. Тому, багато організацій та приватних власників вирішують проблему Web-серверу за допомогою сторонніх організацій – провайдерів чи спеціалізованих фірм, які займаються хостингом.

Для того, щоб розмістити сайт в Інтернеті, необхідно отримати доступ на Web-сервер. Послуги по наданню місця для розміщення сайту, його обслуговуванню і технічному супроводу називаються хостингом (від англ. слова hosting спільне розміщення). Ресурси серверу і лінії зв'язку використовуються спільно безліччю клієнтів.

При виборі стороннього Web-серверу керуються двома основними критеріями: ціною і продуктивністю. Хостинг буває платний (від 85ум. од.) і безкоштовний (рис. 7.9).

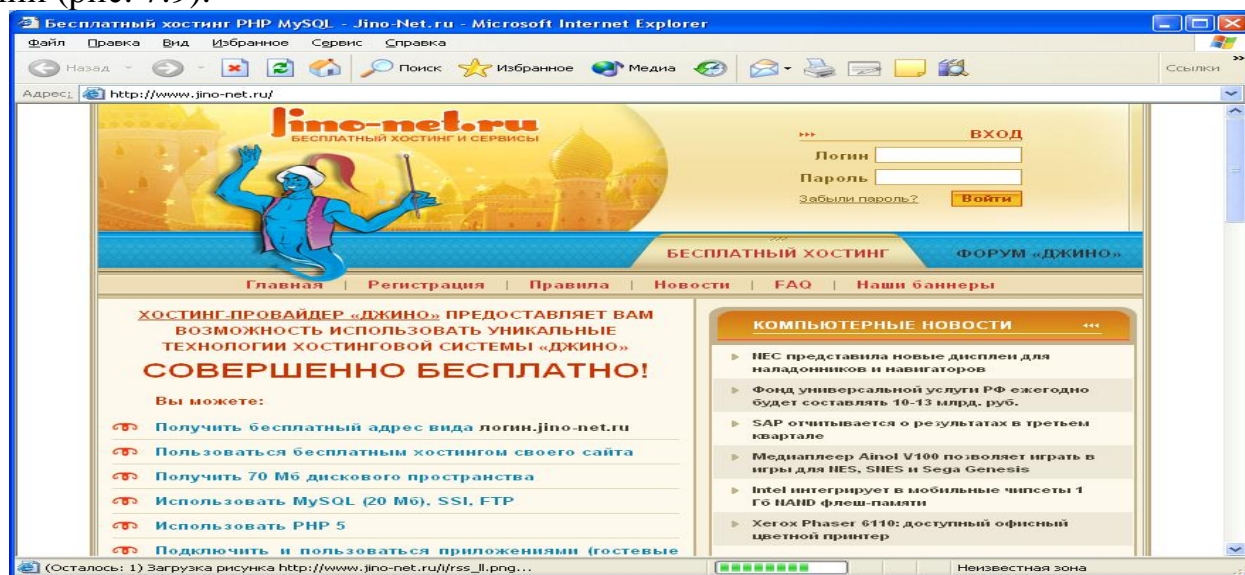


Рис. 7.9. Сервер безкоштовного хостинга Джино

Кількість безкоштовних серверів постійно росте і міняється, деякі час від часу перестають надавати такі послуги, наприклад [www.narod.ru](http://www.narod.ru). До них безкоштовних можна віднести російськомовні сервери [www.narod.ru](http://www.narod.ru), [www.boom.ru](http://www.boom.ru), [www.by.ru](http://www.by.ru), [www.webservis.ru](http://www.webservis.ru), [www.hotbox.ru](http://www.hotbox.ru). Такі сервери не вимагають оплати за підтримку Web-сайтів і існують за рахунок рекламодавців. Зареєстрованому клієнту надається обмежений, але достатній простір на диску серверу в середньому від 10 до 50 Мбайт. На ньому клієнт може розмістити як невеликий, так і солідний сайт. Єдине зобов'язання, що бере на себе клієнт безкоштовного Web-серверу, це розміщення на своєму сайті банерів – маленьких помітних зображень (часто анімованих), призначених для реклами.

При передачі файлів на безкоштовний сервер використовується протокол HTTP, тобто протокол, за допомогою якого здійснюється звичайний обмін файлами в WWW. Послідовність дій при розміщенні Web-сайту:

- підключитися до Internet і завантажити сторінку сервера (наприклад, [www.boom.ru](http://www.boom.ru)).

- Клацнути по посиланню *Регистрируйтесь и начинайте создание сайта* (чи подібне повідомлення). На наступній сторінці натиснути кнопку *Начать регистрацию*. після чого завантажиться сторінка для введення даних.

- Ввести ім'я свого сайту, наприклад, portal22 (адреса сайту матиме вигляд: [www.portal22.boom.ru](http://www.portal22.boom.ru)). Набрати пароль і підтвердити повторним набором. В поле E-mail ввести адресу електронної пошти. На цю адресу буде надісланий лист із підтвердженням реєстрації. В розділі *Дополнительная Информация о Пользователе* набрати своє прізвище, ім'я, нік. В нижній частині вікна набрати код для захисту від автоматичних реєстрацій. Після заповнення обов'язкових полів натиснути кнопку *Отправить*.

- Введені дані будуть відправлені на сервер. Ім'я сайту перевіряється з базою даних існуючих сайтів. Якщо воно унікальне, з'явиться новий діалог, у якому потрібно буде ще раз ввести ім'я сайту, пароль, E-mail та натиснути *Далее*. Якщо сайт з введеним іменем вже існує, потрібно буде повторити процедуру з більшою фантазією.

Для перевірки роботи створеного вузла, потрібно набрати його адресу у вікні браузера, переглянути його сторінки і роботу гіперпосилань.

Окремо слід зазначити, що на деяких хостингових серверах встановлені програмні засоби для розробки web-сторінок. Іншими словами, вам не потрібно мати спеціальне програмне забезпечення для створення html-документів. Звичайно, цією послугою в основному користуються для створення персональних сторінок.

## 7.7. Індивідуальні завдання №8

Всі ці завдання студенти виконують індивідуально. Оскільки, незважаючи на виконання однакових задач, всі елементи сайту мають нести індивідуальні риси його розробника.

### 7.7.1. Створення HTML-сторінок за допомогою Netscape Composer, Front Page, Word порівняння їх можливостей з особливостями роботи браузерів.

1. В редакторі Word за допомогою Майстра створити персональний сайт, з трьох сторінок:
  - а) перша містить загальну інформацію про Вас (Прізвище, ім'я, по батькові, дата народження, група навчання, і т.п.);
  - б) друга сторінка містить інформацію про Ваші захоплення та інтереси (хобі), бажано на сторінку вставити малюнок;



в) третя сторінка містить Ваші координати для зв'язку (адресу, телефон, E-mail).

2. У папці власної групи створити нову папку *САЙТ* і записати в неї всі файли створеного сайту.
3. Переглянути перелік створених майстром файлів та папок з малюнками.
4. Відкрити створений сайт за допомогою браузера.
5. Переглянути HTML-код сторінки. Віднайти основні структурні HTML-елементи сторінки.
6. Створити за допомогою Netscape Composer сторінку *Мої друзі*.
7. Вставити на першу сторінку сайту гіперпосилання на сторінку *Мої друзі* та на будь-який графічний файл та поштовий сервер.
8. Відредагувати першу сторінку за допомогою Front Page. Переглянути можливості по графічному оформленню сторінок.
9. Переглянути створений сайт і роботу гіперпосилань за допомогою браузера.

### **7.7.2. Особливості інтерфейсу Web-редактора Macromedia Homesite 5.0**

1. Завантажити Web-редактор HomeSite 5.
2. Знайти основні структурні елементи робочого вікна.
3. Здійснити налаштування робочого вікна за допомогою меню *Вид*. Записати перелік команд цього меню.
4. Переглянути розділи довідки. До яких змін у вікні це привело?
5. Охарактеризувати набір тегів, що автоматично з'являються при створенні нового документа
6. Відкрити документи, створені на попередньому занятті.
7. Переглянути структуру сайту. Скопіювати структуру в текст звіту.
8. За допомогою панелі швидкої вставки тегів, розбити текстову частину сторінки на три рядки (тег *<BR>* вкладки **Common**).
9. Відкрити/закрити Вікно результатів.
10. Скористатися панеллю інструментів текстового редактора для зміни параметрів редагування документа.
11. Переглянути зроблені зміни за допомогою вкладки **Обзор**.

### **7.7.3. Проектування структури сайту та оформлення тексту на Web-сторінці**

Розробка структури сайту.

1. Спроекувати структуру сайту комерційної фірми, що надає послуги по розробці сайтів: Підібрати матеріал (текст, малюнки); Розбити матеріал по темам; Визначити спосіб переходу між сторінками; Вибрати структуру сайту; Обґрунтувати доцільність вибраної структури.

Результат роботи представити у вигляді текстового файлу.

2. Використовуючи доступний вам редактор, почати розробку сайту, записуючи теги та їх вміст. Зберігати сторінки потрібно з розширенням \*.HTML.

Переглядати отримані результати можна, застосовуючи будь-який браузер, через меню „Файл-Открыть” або “File-Open”. Якщо у файл-менеджері двічі клацнути лівою кнопкою мишки по такому файлу, то його вміст буде показано тим браузером, який визначено як системний.

3. Створити документ HTML наступного змісту і форматування:

### **Портал і К°**

Фірма „Портал” лідер на ринку розробників професійних сайтів в Україні, надає консультаційні послуги по створенню, дизайну та супроводу сайтів. Фірма створена у **1996** році з метою впровадження web-технологій в бізнес-діяльність комерційних підприємств. *В 1998 році фірма стала власником домену .ks.ua.*

4. Фон сторінки – блідо-жовтий, колір тексту сторінки – сірий.
5. Назву оформити заголовком найвищого рівня та синім кольором.
6. Зберегти під іменем start.html.
7. Створити документ HTML наступного змісту:

**Адреса:** м. Дніпропетровськ вул. О. Гончара, 18 , офіс 35, Фірма „Портал”, *Телефон:* (0562)48-17-28, *E-mail:* [ip@portal.ua](mailto:ip@portal.ua)

8. Зберегти під іменем connect.html.
9. Створити документ HTML наступного змісту:

Фірма надає послуги:

- 1) Пошук інформації:
  - Надання цінової інформації;
  - Пошук конкуруючих структур;
  - Аналіз інформації;
- 2) Розробка сайтів.
- 3) Розрахунок кошторису сайту.
- 4) Розміщення сайтів на власному порталі.
- 5) Супроводження сайтів.
- 6) Розміщення реклами в Internet:
  - Розробка банерної реклами;
  - Підбір сайтів для оптимального розміщення реклами.

10. Зберегти під іменем services.html.

11. Переглянути отриманий результат у вікні браузера.

#### **7.7.4. Розміщення гіперпосилань та графічних елементів на Web-сторінках**

Ця робота є продовженням попередньої, тому всі створені раніше сторінки треба доповнити наступними даними:

1. Створити на сторінці start гіперпосилання на файли services і connect.
2. Створити на сторінці start перехід в кінець сторінки.

3. Створити додаткову сторінку „Наші проекти" де вказати гіперпосилання на наступні сервери:
  - Міністерство Аграрної політики України – <http://www.minagro.kiev.ua>
  - Сайт м. Дніпропетровськ – <http://www.gorod.dp.ua>
  - Сайт Дніпропетровського державного аграрного університету – <http://www.dsau.dp.ua>
  - Пошукова система Google – <http://www.Google.com>
  - Пошукова система Апорт – <http://www.aport.ru>
4. На створеній сторінці завантаження вузлів повинно проводитися в нове вікно браузеру.
5. Вставити логотип компанії на першу сторінку.
6. Вставити малюнок, створений в будь-якому графічному редакторі.
7. Зробити малюнок рухомим, щоби він з’являвся з будь-якої сторони сторінки.
8. Оцінити результат у вікні браузера.

#### **7.7.5. Оформлення Web–сторінок з використанням стилів. Створення таблиць у Web-документах. Управління розміщенням елементів Web-сторінки за допомогою таблиць.**

Ця робота є продовженням попередніх, тому всі створені раніше сторінки треба доповнювати наступними даними:

1. Створити стиль оформлення фону сторінки і записати в окремому файлі.
2. Підключити створений стиль до всіх сторінок сайту.
3. Задати для сторінок окремо стилі оформлення заголовків і абзаців, використовуючи класи.
4. Розробити стиль оформлення таблиці за допомогою редактора TopStyle, зберегти в окремому файлі.
5. Підключити стильову таблицю.
6. Створити таблицю, приведену нижче:
7. Переглянути результати в браузері.

#### **7.7.6. Публікація Web-вузла в Internet**

1. Підготувати файли до розміщення в глобальній мережі.
2. За допомогою системи пошуку знайти сайти, що пропонують безкоштовний хостинг.
3. Зробити порівняльну характеристику умов розміщення сайту на приведених серверах.
4. Створити невеличкий сайт, вбудованими засобами створення web-сторінок. Темою сайту може бути: порівняльна характеристика хостингових сайтів.
5. Вибрати сервер для розміщення, переглянувши запропоновані умови.

6. Зареєструвати власний сайт.
7. Скопіювати файли на сервер та перевірити роботу Web-вузла.

Вартість послуг, що надаються фірмою.

Вид робіт	Одиниця ви- міру	Розцінка, грн.	Додаткові доплати	
			За термінове виконання	За обробку ма- теріалу
Пошук інформації:				
• Надання цінової ін- формації:	Друк. арк.	1	10%	15%
• Пошук конкурую- чих структур;				
• Аналіз інформації:				
Розробка сайтів.	Сторінка	20	10%	15%
Розрахунок коштори- су сайту.		безкош- товно		

### Контрольні запитання

1. HTML-сторінка і Web-сторінка це одне і те саме?
2. Що таке структура сайту?
3. Різниця між сайтами з лінійною і деревовидною структурою?
4. Перелічити і дати короткий опис основним елементам інтерфейсу HomeSite 5.
5. За допомогою яких тегів і атрибутів задаються параметри шрифту?
6. Назвіть теги, за допомогою яких створюються таблиці.
7. За допомогою яких атрибутів виконується об'єднання клітинок таблиці?
8. Як задається розмір зображення?
9. У чому полягає попередня робота зі створення Web-сайту?
10. На який сервер не можна помістити вашу HTML-сторінку?
11. Перелічити основні функціональні можливості редактора HomeSite 5.
12. Призначення CSS.
13. Наведіть порядок вміщення Web-сторінок на сайт.

*Подано опис тегів мови HTML, принципи формування таблиць стилів CSS та елементи мови програмування JavaScript. Приведено порядок вміщення Web-сторінок на сайти*

## 8. ПРОГРАМНІ ЗАСОБИ ДЛЯ РОБОТИ ЗІ СТРУКТУРОВАНИМИ ДОКУМЕНТАМИ

*У розділі можна отримати поняття щодо програм, які виконують обробку різних елементів комп'ютерної інформації.*

### 8.1. Програмні засоби для роботи з текстом

До програмних засобів для роботи з текстом відносяться електронні записники, текстові редактори, текстові процесори, редакційні та видавничі системи), програми-перекладачі, програми автореферування тексту, програми і програмні модулі перевірки орфографії, підбору синонімів, лінгвістичні коректори, системи, що здійснюють інтелектуальний пошук і інтелектуальну обробку текстів, розміщених в мережах і ін.

Основні функції *текстового редактора*:

- забезпечення введення тексту з клавіатури або з існуючого файлу;
- редагування тексту (додавання, зміна, видалення або копіювання фрагментів тексту – символів, слів і т.д.);
- оформлення тексту (вибір шрифтів, способу вирівнювання, встановлення міжрядкового і міжабзацного інтервалів і т.п.);
- розміщення тексту на сторінці (установка розміру сторінки, полей, відступів; розбиття на колонки; розстановка номерів сторінок, колонтитулів і ін.);
- збереження тексту у файлі на зовнішньому носіїві або отримання твердої копії (друк тексту);
- перевірка орфографії, підбір синонімів, контекстний пошук і заміна;
- система підказок і т.д.

Найбільш поширені текстові редактори: Лексикон, Edit, Слово і справа, Ched, NotePad, Write, Word Pad, Блокнот.

*Текстовий процесор* відрізняється від текстового редактора ширшими функціо-альними можливостями, а саме:

- меню, що настроюється користувачем;
- використання контекстного меню;
- супровід тексту таблицями і проведення в них простих розрахунків;
- вставка графічних об'єктів (малюнків, діаграм, заголовків і ін.) або створення малюнків за допомогою вбудованих інструментів;
- вставка формул, графіків, діаграм;
- оформлення тексту списками, буквицею;
- використання інструменту автокорекції тексту і його автореферування;
- фонові перевірки орфографії, синтаксису і багато що інше.

Найбільш поширені текстові процесори: Word (Microsoft Office), Word Pro (Lotus SmartSuite), WordPerfect (Perfect Office), WordExpress, Accent, StratusPad, Writer (рис. 8.1).

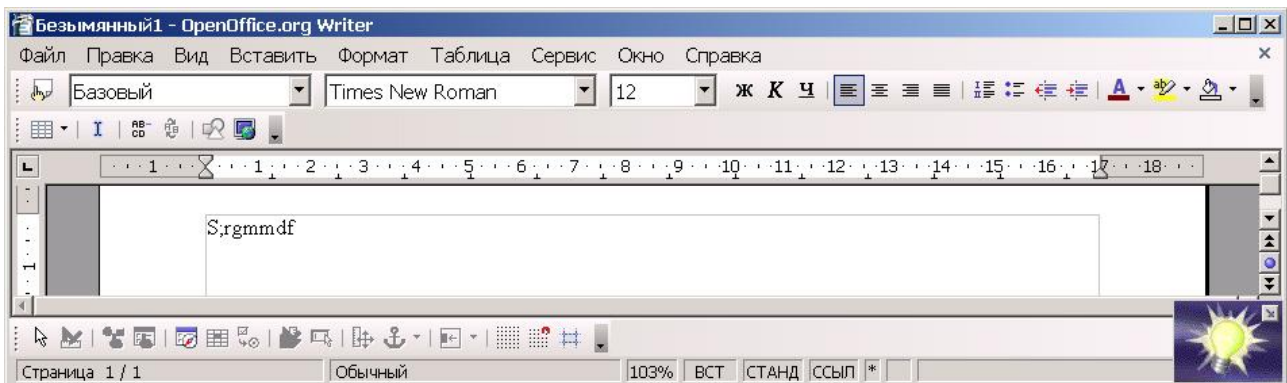


Рис. 8.1. Інтерфейс текстового процесора Writer

*Редакційно-видавничі системи* (програми верстки) повинні забезпечувати всі функції текстового процесора, а також:

- сприймати тексти, створені в різних текстових редакторах;
- сприймати відскановані або намальовані в графічних редакторах ілюстрації, створені на різних платформах ПК, і коректувати їх кольору;
- мати більший набір шрифтів і можливість їх графічного перетворення (стиснення, розтягування або симетричне віддзеркалення по вертикалі або горизонталі);
- мати можливості для різного "обтікання" малюнка текстом;
- забезпечувати автоматичне складання змісту тексту, автоматичне оптимальне розміщення тексту на сторінці;
- забезпечувати адаптацію до різних друкуючих пристроїв і т.д.

Найбільш популярними програмами професійної верстки є QuarkXPress, Ventura Publisher, PageMaker, FrameMaker.

## 8.2. Програмні засоби для обчислювальних робіт

Електронні таблиці, пакети прикладних програм (ППП) для статистичної обробки даних, спеціалізовані математичні ППП

**Електронні таблиці** (SuperCalc, Excel, Lotus, Calc (рис. 8.2) Quattro Pro, SDSS Spreadsheet, VistaCalc, GS-Calc і ін.) відносяться до класу систем обробки числової інформації, званих Spreadsheet. Буквальний переклад терміну "spreadsheet" з англійської мови означає "розстелений лист (папери)".

*Області застосування* електронних таблиць: бухгалтерський і банківський облік; планування розподілу ресурсів; проектний-кошторисні роботи; інженерно-технічні розрахунки; статистична обробка великих масивів інформації; дослідження динамічних процесів.

*Основні можливості* електронних таблиць:

- рішення розрахункових задач, проведення обчислень за формулами, заданим користувачем;
- рішення оптимізаційних задач;
- аналіз і моделювання на основі результатів обчислень;
- оформлення таблиць, звітів;
- побудова діаграм необхідного вигляду;
- створення і ведення баз даних з можливістю вибору записів по заданому критерію і сортування по будь-якому параметру;
- перенесення (вставка) в таблицю інформації з документів, створених в інших програмних засобах;
- друк підсумкових документів;
- колективне використання даних, що зберігаються в таблицях, розповсюдження і переглядання електронних таблиць всіма учасниками робочої групи.

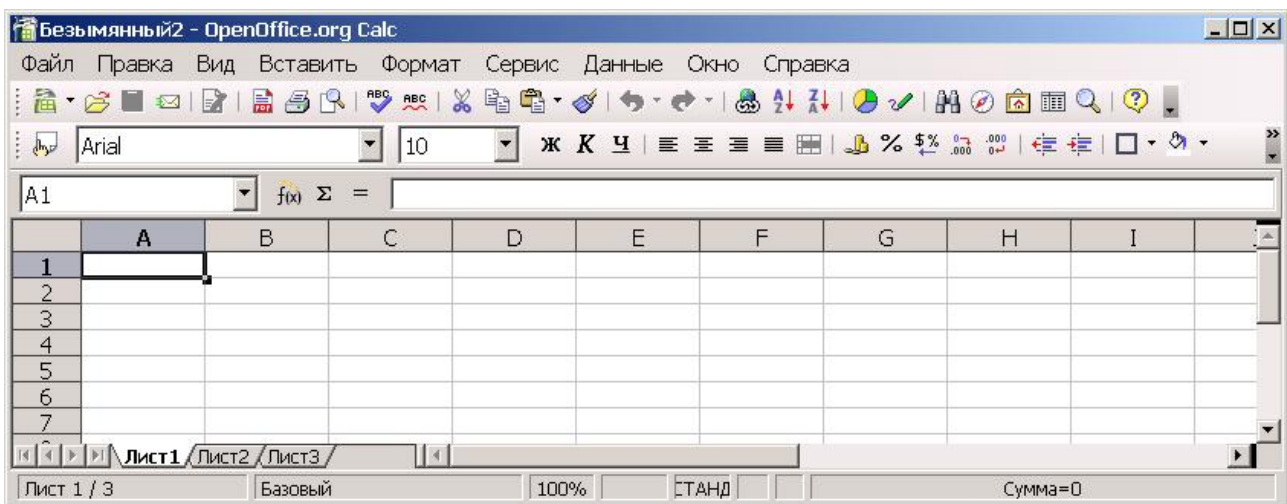


Рис. 8.2. Інтерфейс електронних таблиць Calc

*Елементами таблиці є стовпці, рядки, клітинки, блоки клітинок.*

Найчастіше *рядки* пронумеровані (1, 2, 3, 4 ...), а *стовпці* поіменовані латинськими буквами і комбінаціями букв (A, B, 3 ..., AA, AB..., IV).

Елемент, що знаходиться на перетині стовпця і рядка, називається *клітинкою*.

Прямокутна область таблиці називається *блоком* (діапазоном, інтервалом) клітинок. Блок задається адресами верхньої лівої і правої нижньої клітинок блоку, записаними найчастіше через двокрапку.

### **Модель клітинки в Excel та Calc**

Кожен елемент таблиці має наступні характеристики: адреса; вміст; зображення; формат; ім'я; примітка (коментар).

*Адреса* клітинки – ім'я (номер) стовпця і номер рядка, на перетині яких знаходиться клітинка. Використовується у формулах у вигляді відносного, абсолютного або змішаного посилання, а також для швидкого переміщення по таблиці.

Вмістом клітинки може бути: **число** (ціле із знаком або без (-345), дріб з фіксованою крапкою (253,62) або з плаваючою крапкою (2,5362e + 2)); **текст**; **формула**.

**Формула** – завжди починається із знаку "=" і може містити числові константи, абсолютні або відносні посилання на адреси клітинок, вбудовані функції

Аргументи функцій завжди полягають в круглі дужки. Стандартні функції можна як ввести з клавіатури, так і скористатися меню ВСТАВКА / ФУНКЦІЯ або кнопкою f(x).

*Зображення* – те, що користувач бачить на екрані монітора

Якщо вмістом клітинки є формула, то зображенням буде обчислене числове значення.

Текст, поміщений в клітинку, може бути "видний" цілком, або (якщо сусідня клітинка не порожня) буде видно стільки символів, яка ширина клітинки.

Зображення числа залежить від вибраного формату. Одне і те ж число в різних форматах ("дата", "відсоток", "грошовий" і т.д.) матиме різне зображення.

*Формат клітинки* – визначається форматом чисел, шрифтом, кольором символів, видом рамки, кольором фону, вирівнюванням по межах клітинки, наявністю захисту клітинки.

*Ім'я* – уживається як заміна абсолютної адреси клітинки для використання його у формулах.

*Коментар* – супровідний текст до вмісту клітинки. Ввести примітку в клітинку можна за допомогою меню ВСТАВКА / ПРИМІТКА. Клітинка, що має примітку, наголошується в робочому листі крапкою в правому верхньому кутку.

*Переваги* використання ЕТ при рішенні задач:

1. Рішення задач за допомогою електронних таблиць звільняє від складання докладного алгоритму рішення задачі і відладки відповідної програми. Потрібно тільки певним чином записати в таблицю початкові дані і математичні співвідношення, що входять в модель рішення задачі
2. При використанні однотипних формул немає необхідності вводити їх багато разів, можна скопіювати формулу в потрібні клітинки. При цьому відбудеться автоматичний перерахунок адрес клітинок, що зустрічаються у формулі. Якщо ж необхідно, щоб при копіюванні формули посилання на якусь клітинку не змінилося, то існує можливість завдання абсолютної (незмінної) адреси клітинки.
3. Зміна вмісту будь-якої клітинки призводить до автоматичного перерахунку значень всіх клітинок таблиці, в яких є посилання на дану.
4. Початкові дані і результати розрахунків можна аналізувати як в числовому вигляді, так і представити їх за допомогою ділової графіки (гістограми, секторні діаграми, графіки залежностей і ін.). Причому,



зміна даних, по яких будувалися графіки, автоматично відіб'ється в зміні графічного образу.

**Пакети статистичної обробки** (Systat, Statistica, Stadia і ін.) призначені, як це ясно з назви, для проведення статистичної обробки великих масивів даних.

Відмітимо, що багато електронних таблиць дозволяють користувачеві розрахувати не тільки прості статистичні показники, такі як максимальне, мінімальне і середнє значення або суму квадратів відхилення від середнього значення, але і провести складніші розрахунки з використанням вбудованих статистичних функцій: обчислення коефіцієнтів кореляції, що характеризують ступінь схожості результатів різних вимірювань, ранг числа в списку чисел, коефіцієнти функцій розподілу даних і багато що інше. Пакети статистичної обробки включають реалізацію складніших статистичних методів. Наприклад, вони істотно полегшують проведення *регресійного* (встановлення зв'язку між змінними – результатами статистичних вимірювань), *кластерного* (визначення основних класів, складових процес, що вивчається, або явище), *чинника* (виявлення чинників, що пояснюють результати статистичних вимірювань) і інших видів аналізу даних.

Як правило, пакети статистичної обробки мають розвинені засоби графічного представлення початкових даних і результатів розрахунку. Причому, це не тільки двовимірні діаграми і графіки, але і багатовимірні зображення.

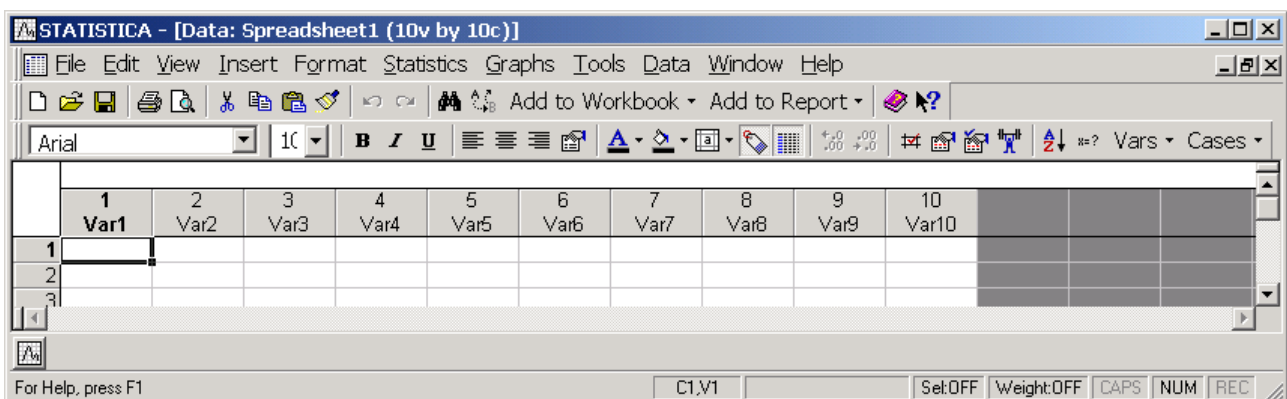


Рис. 8.3. Інтерфейс програми Statistica

**Математичні пакети** (Eureka, Mathcad, Mathcad Professional, Matlab, Maple, Mathematica, Maxima і ін.) дозволяють вирішити практично будь-яку математичну задачу і представити результати розрахунків в табличному або графічному вигляді. Причому, багато математичних пакетів мають розвинені засоби побудови тривимірних поверхонь, що задаються за допомогою функцій.

### 8.3. Програмні засоби роботи з графікою

До програмних засобів роботи з графікою відносяться графічні редактори (Paint графічні редактори; графічні процесори, аніматори; програмні засоби для роботи з тривимірною графікою; засоби ділової графіки; засоби для створення презентацій, засобу моделювання і проектування.

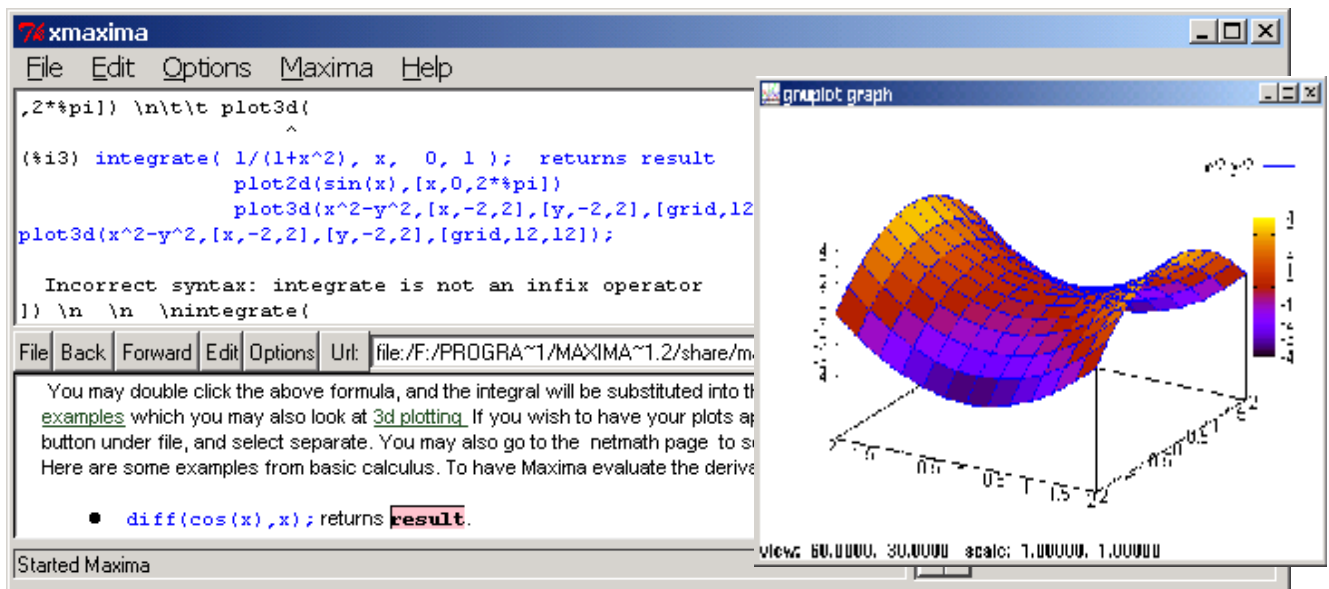


Рис. 8.4. Інтерфейс математичного пакету Maxima

По своєму "професійному" призначенню засобу комп'ютерної графіки і анімації можна підрозділити на наступні групи:

- пакети комп'ютерної графіки для поліграфії – дозволяють доповнювати текст ілюстраціями різного походження, створювати дизайн сторінок і виводити поліграфічну продукцію на друк з високою якістю;
- програми двовимірного комп'ютерного живопису – графічні редактори;
- презентаційні пакети, використовувані як засоби створення різноманітних слайдів для супроводу доповідей, виступів, рекламних акцій;
- програми двовимірної анімації, використовувані для створення динамічних зображень і спецефектів в кіно;
- програми для двовимірного і тривимірного моделювання, вживані для дизайнерських і інженерних розробок;
- пакети тривимірної анімації, використовувані для створення рекламних і музичних кліпів і кінофільмів;
- комплекси для обробки відеозображень, необхідні для накладення анімаційних спецефектів на відеозапис;
- програми для наукової візуалізації.

*Графічні редактори* (Painter, Corel Draw, FreeHand, Picture Man, Draw і ін.) призначені переважно для перегляду, створення і редагування площинних (двовимірних) статичних зображень.

*Графічні процесори* (Adobe Photoshop, Adobe Illustrator) призначені не стільки для створення, скільки для перетворення існуючих зображень, отриманих шляхом сканування, цифрової фотографії і т.д. Вони включають як інструменти для обробки зображень різноманітні ефекти.

Програми *аніматори* (Animator Pro, PowerAnimator, Animation Works Interactive, Animo і ін.) можуть створювати і працювати як з двовимірними, так і з тривимірними зображеннями. На відміну від традиційної анімації, де кожен кадр малюється уручну, в комп'ютерній двовимірній анімації значну частину

рутинної роботи бере на себе програма. Використовуючи спеціальні інструменти призначеного для користувача меню таких програм, можна задати рух по певній траєкторії (автоматично будуть створені відповідні проміжні кадри) або плавно змінити палітру протягом декількох кадрів (наприклад, поступово затемнити зображення або прибрати частину кольорів).

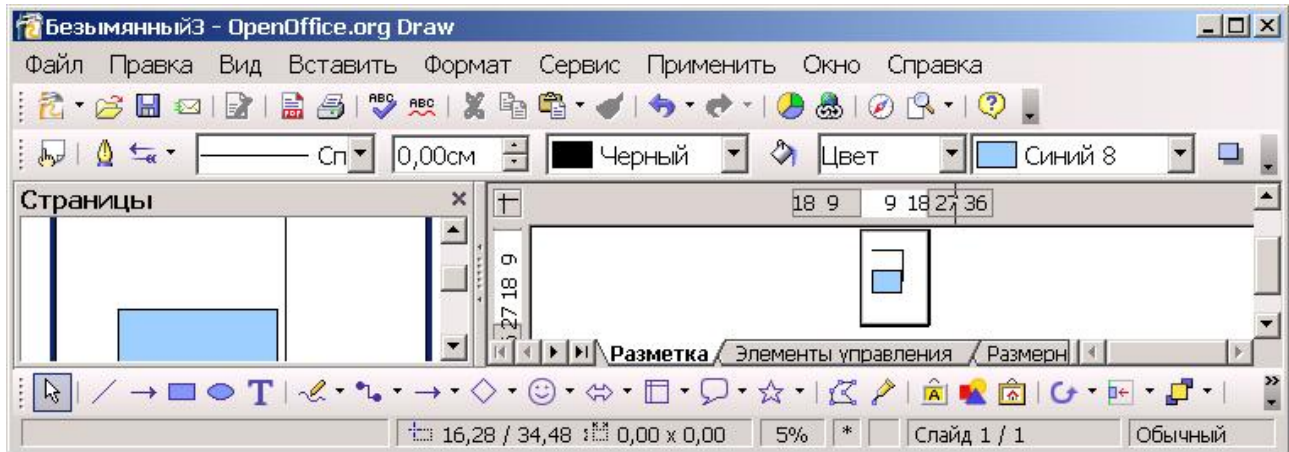


Рис. 8.5. Інтерфейс графічного редактора Draw

#### 8.4. Програмні засоби роботи зі звуком

До програмних засобів роботи із звуком відносять музичні редактори, синтезатори звуків, зокрема, синтезатори мови, системи автоматичного розпізнавання мови, звукові редактори, голосові навігатори, що дозволяють реалізувати мовний інтерфейс користувача, програми диктування, що дозволяють перетворювати мову в "письмовий" текст, програми для поліпшення якості фонограм і ін.

З появою в 1989 р. звукових карт, перед користувачами відкрилися нові можливості. І річ навіть не в тому, що на порядок покращала якість звуку. З'явилася *звукова підсистема* — комплекс програмно-апаратних засобів, призначений для:

- записи звукових сигналів, що поступають від зовнішніх джерел, наприклад, мікрофону або магнітофона. В процесі запису вхідні аналогові звукові сигнали перетворюються в цифрові і далі можуть бути збережені на вінчестері ПК;
- відтворення записаних раніше звукових даних за допомогою зовнішньої акустичної системи або головних телефонів (навушників), відтворення звукових компакт-дисків;
- обробки звукових сигналів: редагування, об'єднання або розділення фрагментів сигналу, фільтрації, зміни його рівня і т.п.;
- генерування за допомогою синтезатора звучання музичних інструментів (мелодійних і ударних), а також людської мови і будь-яких інших звуків;
- микширования (змішування) при записі або відтворенні сигналів від декількох джерел;

- управління панорамою стереофонічного звукового сигналу (розташуванням джерел звуку, що здається) і рівнем сигналу в кожному каналі при записі і відтворенні;
- управління роботою зовнішніх електронних музичних інструментів через спеціальний інтерфейс MIDI (Musical Instrument Device Interface);
- управління комп'ютером і введення тексту за допомогою мікрофону.

До програмних засобів введення і обробки звукової інформації відносяться в основному музичні редактори, синтезатори звуків, зокрема, синтезатори мови, програми для розпізнавання мови, редактори оцифровок реальних звуків (смплов), звукові редактори, генератори стилів звучання музичних інструментів, програми для поліпшення якості фонограм і ін.

Найбільш популярними програмними засобами для **синтезу, обробки і відтворення звуку** є Adagio, TiMidity, Playmidi, Tracker, Gmod, MikMod, XAudio, S3mod, Nspmod, Yampmod і ін.

*Створення (синтез) звуку* в основному переслідує дві мету:

- імітація різних природних звуків (шум вітру і дощу, звук кроків, спів птахів і т.п.), а також акустичних музичних інструментів;
- отримання принципово нових звуків, що не зустрічаються в природі.

*Обробка звуку* зазвичай направлена на отримання нових звуків з що вже існують (наприклад, голос робота), або додання їм додаткових якостей або усунення тих, що існують (наприклад, додавання ефекту хору, видалення шуму або клацань).

*Програми обробки цифрового звуку* (Cool Editor, Sound Forge, Samplitude, Software Audio Workshop) дають можливість прослуховувати вибрані ділянки, робити вирізки і вставки, амплітудні і частотні перетворення, звукові ефекти (луну, реверберацію, фленжер, дисторшн), накладення інших оцифровок, зміну частоти оцифровки, генерувати різні види шумів, синтезувати звук.

Які основні методи використовуються для обробки звуку? *Монтаж і редагування*. Полягає у вирізуванні із запису одних ділянок, вставці інших, їх заміні, розмноженні і т.п. Практично кожен музичний редактор має такі можливості редагування. Все сучасні звуко- і відеозаписи в тій чи іншій мірі піддаються монтажу.

*Амплітудні перетворення*, наприклад, посилення або ослаблення звуку.

*Частотні (спектральні) перетворення*, наприклад, фільтрація – посилення або ослаблення певних смуг частот.

*Фазові перетворення*. Слуховий апарат людини використовує фазу для визначення напрямку на джерело звуку. Фазові перетворення стереозвуку дозволяють отримати ефекти звуку, що обертається, рухомого джерела звуку і подібні до них.

*Тимчасові перетворення*. Полягають в додаванні до основного сигналу його копій, зрушених в часі на різні величини. При невеликих зрушеннях (порядку менше 20 мс) це дає ефект розмноження джерела звуку (ефект хору), при великих – ефект луни.

*Формантніє перетворення* оперують з формантами – характерними смугами частот, що зустрічаються в звуках, вимовних людиною. Кожному звуку відповідає своє співвідношення амплітуд і частот декілька формант, яке визначає тембр і розбірливість голосу. Змінюючи параметри формант, можна підкреслювати або затушовувати окремі звуки, міняти одну голосну на іншу, зрушувати реєстр голосу і т.п.

Обробка мовної інформації включає *синтез мови* і *автоматичне розпізнавання мови*.

В даний час сфера застосування *синтезаторів мови* безперервно розширюється – використовуються різні автоматизовані інформаційно-довідкові системи, системи автоматизованого контролю, здатні голосом попередити людину про стан контрольованого об'єкту, і інші системи.

Розроблені пристрої, що дозволяють перетворити письмовий текст у відповідне йому фонемне уявлення, що дозволяє відтворювати у вигляді мови довільний текст, що зберігається в пам'яті комп'ютера.

Немало зусиль було покладено на те, щоб забезпечити програми і операційні системи графічним інтерфейсом користувача. Зараз розвивається новий напрям – *мовний інтерфейс користувача*. Голосові навігатори (PilotVoice, Listen, Just Voice, Speech Recognizer, QwickSwitchBitWare, Voice Assist) управляють програмами, в якійсь мірі замінюючи клавіатуру і мишу.

Росте популярність засобів *автоматичного розпізнавання мови* (Automated Speech Recognition, ASR). Системи ASR (програми диктування DragonDictate, Office TalkKolvox Communication, Power Secretary, VoiceType Dictation) перетворюють мову в закодований "письмовий" текст. Для цього проводиться спектральний аналіз оцифрованої мови і визначаються за допомогою спеціальних математичних методів мінімальні звукові одиниці мови – фонемі.

### **Контрольні запитання**

1. У чому різниця поміж текстовим редактором і текстовим процесором?
2. Назвіть основні відміни електронних таблиць від текстових редакторів.
3. Чи потрібно знати якусь мову програмування для проведення розрахунків за допомогою програми Maxima?
4. Які ви знаєте форми представлення графіки?
5. Скільки відтінків можна відтворити на комп'ютері?
6. Поясніть різницю поміж аналоговим та оцифрованим звуком.

*У розділі викладено поняття щодо програм для роботи з текстом, для обчислювальних робіт, для роботи з графікою, для роботи зі звуком.*

# 9. ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ

Вивчити можливості SQL мови для витягу інформації з таблиці, студент зможе зробити запит будь-якого числа таблиць, освоїти використання запитів по витягу значень з бази даних (БД). Вивчити реляційні операції, які використовують в SQL булеві операції.

## 9.1. Основні поняття реляційних баз даних

Основою сучасних систем, що застосовують бази даних, є *реляційна модель даних*. У цій моделі дані, що представляють інформацію про предметну область, організовані у виді двомірних таблиць, названих *відносинами*. На Рис. 9. 1 приведений приклад такої таблиці-відносини і пояснюються основні терміни реляційної моделі.



Рис. 9.1. Приклад таблиці-відносини реляційної бази даних

Відношення – це таблиця, подібна приведеній на рис. 9.1, яка складається з рядків і стовпців. Верхній рядок таблиці-відносини називається *заголовком відносини*. Терміни *відношення* і *таблиця* звичайно вживаються як синоніми, однак у мові SQL використовується термін *таблиця*.

- Рядки таблиці-відносини називаються *кортежами*, або *записами*. Стовпці називаються *атрибутами*. Терміни – атрибут, стовпець, стовпчик, поле – звичайно використовуються як синоніми. Кожен атрибут має ім'я, що повинне бути унікальним у конкретній таблиці-відносини, однак у різних таблицях імена атрибутів можуть збігатися.
- Кількість кортежів у таблиці-відносини називається *кардинальним числом* відносини, а кількість атрибутів – *ступенем* відносини.

- *Ключ*, або *первинний ключ* відносини – це унікальний ідентифікатор рядків (кортежів), тобто такий атрибут (набір атрибутів), для якого в будь-який момент часу у таблиці-відносини не існує рядків з однаковими значеннями цього атрибута (набору атрибутів). На приведеному рисунку таблиці, осередок з імям ключового атрибута має нижню границю у виді риси.
- *Домен* відносини – це сукупність значень, з яких можуть вибиратися значення конкретного атрибута. Тобто конкретний набір наявних у таблиці значень атрибута в будь-який момент часу повинен бути підмножиною безлічі значень домена, на якому визначений цей атрибут. У загальному випадку на тому самому домені можуть бути визначені значення різних атрибутів. Важливим є те, що домени вводять обмеження на операції порівняння значень різних атрибутів. Ці обмеження полягають у тому, що коректним чином можна порівнювати між собою тільки значення атрибутів, визначених на тому самому домені.

Щодо властивості реляційного відношення, що стосується відсутності кортежів-дублікатів, варто зробити важливе зауваження. У цьому пункті SQL не цілком відповідає реляційній моделі. А саме: у відносинах, що є результатами запитів, SQL *допускає* наявність однакових рядків. Для їх усунення в запиті використовується ключове слово DISTINCT (див. нижче).

Інформація в реляційних базах даних, як правило, зберігається не в одній таблиці-відношенні, а в декількох. При створенні декількох таблиць взаємозалежної інформації з'являється можливість виконання більш складних операцій з даними, тобто більш складної їхньої обробки. Для роботи зі зв'язаними даними з декількох таблиць важливим є поняття так званих *зовнішніх ключів*.

*Зовнішнім ключем* таблиці називається атрибут (набір атрибутів) цієї таблиці, кожне значення, якого в поточному стані таблиці завжди збігається зі значенням атрибутів, що є ключем, в іншій таблиці. Зовнішні ключі використовуються для зв'язування значень атрибутів з різних таблиць. За допомогою зовнішніх ключів забезпечується так названа посилальна цілісність бази даних, тобто погодженість даних, що описують ті самі об'єкти, але зберігаються в різних таблицях.

## 9. 2. Складові частини SQL

І інтерактивний, і убудований SQL підрозділяються на наступні складові частини.

*Мова визначення даних* – DDL (Data Definition Language) – дає можливість створення, зміни і видалення різних об'єктів бази даних (таблиць, індексів, користувачів, привілеїв і т.д.)

У число додаткових функцій DDL можуть бути включені і засоби обмеження цілісності даних, визначення порядку структур їх збереження, опису елементів фізичного рівня збереження даних.

*Мова обробки даних* – DML (Data Manipulation Language) – надає можливість вибірки інформації з бази даних і її перетворення.

Проте, це не дві різні мови, а компоненти єдиного SQL.

### 9.3. Типи даних SQL

У мові SQL є засоби, що дозволяють для кожного атрибута вказувати тип даних, якому повинні відповідати усі значення цього атрибута.

Слід зазначити, що визначення типів даних є тією частиною, у якій комерційні реалізації мови не цілком погодяться з вимогами офіційного стандарту SQL. Це зумовлено, зокрема, бажанням забезпечити сумісність SQL з іншими мовами програмування.

#### 9.3.1. Символьні функції

Стандарт підтримує тільки один тип представлення тексту – CHARACTER (CHAR). Цей тип даних являє собою символьні рядки фіксованої довжини. Його синтаксис має вигляд:

CHARACTER [(довжина)] або  
CHAR [(довжина)].

Текстові значення поля таблиці, визначеного як тип CHAR, мають фіксовану довжину, що визначається параметром довжина. Цей параметр може приймати значення від 1 до 255, тобто рядок може містити до 255 символів. Якщо в текстовій константі, яка вводиться в поле, фактичне число символів менше числа, зазначеного параметром довжина, то ця константа автоматично доповнюється праворуч пробілами до заданого числа символів.

Деякі реалізації мови SQL підтримують, як тип даних, рядки перемінної довжини. Цей тип може позначатися ключовими словами VARCHAR (CHARACTER VARYING або CHAR VARYING). Він описує текстовий рядок, що може мати довільну довжину до зазначеного конкретною реалізацією SQL максимуму (у Oracle – до 2000 символів). На відміну від типу CHAR у цьому випадку при введенні текстової константи, фактична довжина якої менше заданої, не робиться її доповнення пробілами до заданого максимального значення.

Константи, що мають тип CHARACTER і VARCHAR, у вираженнях SQL полягають в одиночних лапках, наприклад, ' текст'.

Наступні пропозиції еквівалентні:

VARCHAR [ (довжина) ], CHAR VARYING [ (довжина) ], CHARACTER VARYING [ (довжина) ].

Якщо довжина рядка не зазначена реально, вона покладається рівною одному символу в усіх випадках.

У порівнянні з типом CHAR тип даних VARCHAR дозволяє більш ощадливо використовувати пам'ять, відведена для збереження текстових значень, і виявляється більш зручним при виконанні операцій, зв'язаних з порівнянням текстових констант.

#### 9.3.2. Числові типи даних

Стандартними числовими типами даних SQL є:

INTEGER – використовується для представлення цілих чисел у діапазоні від  $-2^{31}$  до  $+2^{31}$ .



SMOLLINT – використовується для представлення цілих чисел у меншому, чим для INTEGER, діапазоні, а саме – від  $-2^{15}$  до  $+2^{15}$ .

DECIMAL (точність[,масштаб]) – десяткове число з фіксованою крапкою, точність визначає кількість значущих цифр у числі. Масштаб указує максимальне число цифр праворуч від крапки.

NUMERIC (точність[,масштаб]) – десяткове число з фіксованою крапкою, таке ж, як і DECIMAL.

FLOAT [(точність)] – число з крапкою, що плаває, і зазначеною мінімальною точністю.

REAL – число таке ж, як при типі FLOAT, за винятком визначення точності за замовчуванням (у залежності від конкретної реалізації SQL).

DOUBLE PRECISION – число аналогічне REAL, але точність у два рази вище точності REAL.

СУБД Oracle і Base використовує додатково тип даних NUMBER для представлення усіх числових даних, цілих, з фіксованою або крапкою, що плаває. Його синтаксис: NUMBER [(точність [,масштаб])].

Якщо значення параметра точність не зазначена явно, воно покладається рівним 38. Значення параметра масштаб за замовчуванням передбачається рівним 0. Значення параметра *точність* може змінюватися від 1 до 38; значення параметра *масштаб* може змінюватися від – 84 до 128. Використання негативних значень масштабу означає зрушення десяткової крапки у бік старших розрядів. Наприклад, визначення NUMBER (7,3) означає округлення до тисяч.

Тип DECIMAL (іноді позначається як DEC) і NUMERIC цілком еквівалентні типові NUMBER. Синтаксис: DECIMAL [(точність[,масштаб])],  
DEC [(точність[,масштаб])],  
NUMERIC [(точність[,масштаб])].

### 9.3.3. Дата і час

Тип даних, призначений для представлення дати і часу, також є нестандартним, хоча і надзвичайно корисним. Для точного визначення типів даних, що підтримуються конкретною СУБД, варто звертатися до її документації.

У СУБД Oracle, мається тип DATE, що використовується для збереження дати і часу. Підтримуються дати, починаючи від 1 січня 4712 року до н.е. і до 31 грудня 4712 року. У Access тип Дата/Час містить значення дати і часу і допускає введення дат з діапазону від 100 до 9999 років. За замовчуванням при визначенні дати без уточнення часу приймається час на півночі.

Наявність типу даних для збереження дати і часу дозволяє підтримувати спеціальну арифметику дат і часів. Додавання до перемінного типу DATE цілого числа означає збільшення дати на відповідне число днів, а вирахування відповідає визначенню більш ранньої дати.

Константи типу DATE записуються в залежності від формату, прийнятого в операційній системі. Наприклад, '03.05.1999', або '12/06/1989', або '03-янв-1999', або '03-янв-99'.

### 9.3.4. Невизначені або пропущені дані (NULL)

Для позначення відсутніх, пропущених або невідомих значень атрибута, у SQL використовується ключове слово NULL. Досить часто можна зустріти словосполучення «*атрибут має значення NULL*». Відверто говорячи, NULL не є значенням у звичайному розумінні, а використовується саме для позначення того факту, що дійсне значення атрибута насправді пропущене або невідомо. Це приводить до ряду особливостей, що варто враховувати при використанні значень атрибутів, які можуть знаходитися в стані NULL.

- У функціях, що дозволяють одержувати зведену інформацію з безлічі значень атрибута, наприклад, сумарне або середнє значення, для забезпечення точності й однозначності тлумачення результатів, відсутні або NULL-значення атрибутів ігноруються.
- Умовні оператори від булевої двозначної логіки TRUE/FALSE розширюються до тризначної логіки TRUE/FALSE/UNKNOWN.
- Всі оператори, за винятком оператора конкатенації рядків «+», повертають порожнє значення (NULL), якщо значення кожного з операндів відсутнє (має «значення NULL»).
- Для перевірки на порожнє значення варто використовувати оператори `is NULL` і `is NOT NULL` (використання з цією метою оператора порівняння «=» є помилкою).
- Функції перетворення типів, що мають NULL як аргумент, повертають порожнє значення (NULL).

## 9.4. Використані терміни і позначення

*Ключові слова* – це використані у виразах SQL слова, що мають спеціальне призначення (наприклад, конкретні команди SQL). Ключові слова не можна використовувати для інших цілей, наприклад, такі як імена об'єктів бази даних. У книзі вони виділяються шрифтом: КЛЮЧОВЕ слово.

*Команди, або пропозиції*, є інструкціями, за допомогою яких SQL звертається до бази даних. Команди складаються з однієї або більш логічних частин, які називають реченнями. Пропозиції починаються ключовим словом і складаються з ключових слів і аргументів.

Об'єкти бази даних, що мають імена (таблиці, атрибути й ін.), у книзі також виділяються особливим образом: ТАБЛИЦЯ1, АТРИБУТ\_2.

В описі синтаксису команд SQL:

- оператор визначення «:=» розділяє обумовлений елемент (ліворуч від оператора) і власне його визначення (праворуч від оператора);
- квадратні дужки «[ ]» указують *необов'язковий* елемент синтаксичної конструкції;
- багато крапка «... » визначає, що вираз, може повторюватися будь-яке число раз;
- фігурні дужки «{ }» поєднують послідовність елементів у *логічну групу*, один з елементів якої повинен бути обов'язково використаний;

- вертикальна риса «|» указує, що частина визначення, яка впливає за цим символом, є одним з можливих варіантів;
- у кутові дужки «< >» полягають елементи, що пояснюються в міру того, як вони вводяться.

## 9.5. Навчальна база даних

У прикладах, що приводяться в посібнику, побудови SQL-запитів і контрольних вправ використовується база даних, яка складається з табл. 9.1-9.6.

Таблиця 9.1

### STUDENT (Студент)

STUDENT_ID (код студента)	SURNAME (прізвище)	NAME (ім'я)	STIPEND (розмір стипендії)	KURS (курс)	CITY (місто)	BIRTHDAY (дата народження)	UNIV_ID (код університету)
1	Іванов	Іван	15	1	Орел	3/12/19	10
3	Петров	Петро	20	3	Курськ	1/12/19	10
6	Сидоров	Вадим	15	4	Київ	7/06/19	22
10	Кузне-	Борис	0	2	Брянськ	8/12/19	10
12	Зайцева	Ольга	25	2	Липецьк	1/05/19	10
265	Павлов	Анд-	0	3	Дніпропетровськ	5/11/19	10
32	Котів	Павло	15	5	Белгород	NULL	14
654	Лукин	Артем	20	3	Дніпропет-	1/12/19	10
276	Петров	Антон	20	4	NULL	5/08/19	22
55	Белкін	Вадим	20	5	Дніпропет-	7/01/19	10

Таблиця 9.2

### LECTURER (Викладач)

LECTURER_ID (код викладача)	SURNAME (прізвище викладача)	NAME (ім'я)	CITY (місто)	UNIV_ID (код університету)
24	Колісників	Борис	Дніпропет-	10
46	Ніконов	Іван	Дніпропет-	10
74	Лагутін	Павло	Київ	22
108	Струков	Микола	Київ	22
276	Миколаїв	Віктор	Дніпропет-	10
328	Сорокін	Андрій	Орел	10

Таблиця 9.3

## SUBJECT (Предмет навчання)

SUBJ_ID (код предмета)	SUBJ_NAME (назва пред- мету)	HOUR (кількість годин)	SEMESTER (семестр)
10	Інформатика	56	1
22	Фізика	34	1
43	Математика	56	2
56	Історія	34	4
94	Англійський	56	3
73	Фізкультура	34	5

Таблиця 9.4

## UNIVERSITY (Університети)

UNIV_ID (код університе- ту)	UNIV_NAME (назва університету)	RATING	CITY
22	КГУ	606	Київ
10	ВГУ	296	Дніпропетровськ
11	НГУ	345	Новосибірськ
32	РГУ	416	Ростов
14	БГУ	326	Белгород
15	ТГУ	368	Томськ
18	ВГМА	327	Дніпропетровськ

Таблиця 9.5

## EXAM\_MARKS (Екзаменаційні оцінки)

EXAM_ID (код іспиту)	STUDENT_ID (код студента)	SUBJ_ID (код предмета)	MARK (оцінка)	EXAM_DATE (дата іспиту)
145	12	10	5	12/01/2000
34	32	10	4	23/01/2000
75	55	10	5	05/01/2000
238	12	22	3	17/06/1999
639	55	22	NULL	22/06/1999
43	6	22	4	18/01/2000

Таблиця 9.6

## SABJ\_LECT

(Навчальні дисципліни викладачів)

LECTURER_ID (Код викладача)	SUBJCD (Код предмета)
24	24
46	46
74	74

LECTURER_ID (Код викладача)	SUBJCD (Код предмета)
108	108
276	276
328	328

## 9.6. Найпростіші SELECT-запити

Оператор SELECT (вибрати) мови SQL є найважливішим оператором, який використовується найчастіше. Він призначений для вибірки інформації з таблиць бази даних. Спрощений синтаксис оператора SELECT виглядає в такий спосіб.

```
SELECT [DISTINCT] <список атрибутів>
FROM <список таблиць>
[WHERE <умова вибірки>]
[ORDER BY <список атрибутів>]
[GROUP BY <список атрибутів>]
[HAVING <умова>]
[UNION <вираження з оператором SELECT>];
```

У квадратних дужках зазначені елементи, які у запиті можуть бути відсутні.

Ключове слово SELECT повідомляє базі даних, що дана речення є запитом на витяг інформації. Після слова SELECT через кому перелічуються найменування полів (список атрибутів), зміст яких запитується.

Обов'язковим ключовим словом у запиті пропозиції SELECT є слово FROM (з). За ключовим словом FROM указується список розділених комами імен таблиць, з яких витягається інформація.

Наприклад,  
SELECT NAME,SURNAME  
FROM STUDENT;

SQL-запит повинен закінчуватися символом «;» (крапка з комою).

Приведений запит здійснює вибірку всіх значень полів NAME і SURNAME з табл.1 STUDENT. Його результатом є таблиця наступного виду:

Таблиця 9.7

NAME	SURNAME
Іван	Іванов
Петро	Петров
Вадим	Сидоров
Борис	Кузнецов
Ольга	Зайцева
Андрій	Павлов
Павло	Котів
Артем	Лукин
Антон	Петров
Вадим	Белкін

Порядок проходження стовпців у цій таблиці відповідає порядку полів NAME і SURNAME, зазначеному в запиті, а не їх порядку у вхідній таблиці STUDENT.

Якщо необхідно вивести значення *усіх*, стовпців таблиці, то можна замість перерахування їх імен використовувати символ «\*» (зірочка).

```
SELECT *  
FROM STUDENT;
```

У даному випадку результатом виконання запиту буде вся таблиця STUDENT.

Ще раз зверніть увагу на те, що одержувані в результаті SQL-запиту таблиці не повною мірою відповідають визначенню реляційної відносини. Зокрема, у них можуть виявитися кортежі (рядки) з однаковими значеннями атрибутів.

Наприклад, запит «Одержати список назв міст, де проживають студенти, зведення про яких знаходяться в таблиці STUDENT», можна записати в наступному виді:

```
SELECT CITY  
FROM STUDENT;
```

Його результатом буде табл. 9.8

Видно, що в таблиці зустрічаються однакові рядки (виділені жирним шрифтом).

Для вилучення з результату SELECT-запиту записів, які повторюються, використовується ключове слово DISTINCT (відмінний). Якщо запит SELECT витягає безліч полів, то DISTINCT *виключає* дублікати рядків, у яких значення *всіх* обраних полів ідентичні.

Попередній запит можна записати в наступному виді:

```
SELECT DISTINCT CITY  
FROM STUDENT;
```

В результаті одержимо табл.9.9, у якій дублікати рядків вилучені.

Ключове слово ALL (усі), на відміну від DISTINCT, робе протилежну дію, тобто при його використанні рядки, які повторюються, *включаються* до складу вихідних даних. Режим, що задається ключовим словом ALL, діє за замовчуванням, тому в реальних запитах для цих цілей воно практично не використовується.

Використання в операторі SELECT пропозиції, обумовленого ключовим словом WHERE (де), дозволяє задавати вираз умови (предикат), що приймає значення *істина* або *неправда* для значень полів рядків таблиць, до яких звертається оператор SELECT. Речення WHERE визначає, *які рядки зазначених таблиць повинні бути обрані*. У таблицю, що є результатом запиту, включають-

Таблиця 9.8

CITY
Орел
Курск
Киев
Брянск
Липецк
<b>Дніпропетровськ</b>
Белгород
<b>Дніпропетровськ</b>
NULL
<b>Дніпропетровськ</b>

Таблиця 9.9

CITY
Орел
Курск
Киев
Брянск
Липецк
Дніпропетровськ
Белгород
NULL

ся тільки ті рядки, для яких умова (предикат), зазначена в пропозиції WHERE, приймає значення *істина*.

#### Приклад:

Написати запит, що виконує вибірку імен (NAME) всіх студентів із прізвищем (SURNAME) Петров, дані про яких знаходяться в таблиці STUDENT.

```
SELECT SURNAME, NAME
FROM STUDENT
WHERE SURNAME = 'Петров';
```

Результатом цього запиту буде таблиця:

Таблиця 9.10

SURNAME	NAME
Петров	Петро
Петров	Антон

В умовах, що задаються в пропозиції WHERE можуть використовуватися операції порівняння, обумовлені операторами = (дорівнює), > (більше), < (менше), >= (більше або дорівнює), <= (менше або дорівнює), <> (не дорівнює), а також логічні оператори:

AND, OR і NOT.

Наприклад, запит для одержання імен і прізвищ студентів, що навчаються на *третьому* курсі й одержують стипендії (розмір стипендії *більше нуля*), буде виглядати в такий спосіб:

```
SELECT NAME, SURNAME
FROM STUDENT
WHERE KURS = 3 AND STIPEND > 0;
```

Результат виконання цього запиту має вигляд:

Таблиця 9.11

SURNAME	NAME
Петров	Петро
Лукин	Артем

### 9.7. Оператори IN, BETWEEN, LIKE, is NULL

При завданні логічної умови в пропозиції WHERE можуть бути використані оператори IN, BETWEEN, LIKE, is NULL.

Оператори IN (дорівнює кожному зі списку) і NOT IN (не дорівнює жодному зі списку) використовуються для порівняння значення, що перевіряється, поля з заданим списком. Цей список значень вказується в дужках праворуч від оператора IN.

Побудований з використанням IN предикат (умова) вважається щирим, якщо значення поля, ім'я якого зазначено ліворуч від IN, збігається (мається на увазі точний збіг) з одним зі значень, перерахованих у списку, зазначеному в дужках праворуч від IN.

Предикат, побудований з використанням NOT IN, вважається щирим, якщо значення поля, ім'я якого зазначено ліворуч від NOT IN, не збігається з жодним зі значень, перерахованих у списку, зазначеному в дужках праворуч від NOT IN.

Приклади:

Одержати з таблиці EXAM\_MARKS дані про студентів, що мають екзаменаційні оцінки тільки 4 і 5.

```
SELECT *
FROM EXAM_MARKS
WHERE MARK IN (4, 5 );
```

Одержати дані про студентів, що не мають ні однієї екзаменаційної оцінки, рівної 4 і 5.

```
SELECT *
FROM EXAM_MARKS
WHERE MARK NOT IN (4, 5 );
```

Оператор BETWEEN використовується для перевірки умови входження значення поля в заданий інтервал, тобто замість списку значень атрибута цей оператор задає границі його зміни.

Наприклад, запит на висновок записів про предмети, на вивчення яких приділяється кількість годин, що знаходиться в межах між 30 і 40, має вигляд:

```
SELECT *
FROM SUBJECT
WHERE HOUR BETWEEN 30 AND 40;
```

Граничні значення, у даному випадку значення 30 і 40, входять у безліч значень, з якими виробляється порівняння. Оператор BETWEEN може використовуватися як для числових, так і для символічних типів полів.

Оператор LIKE застосуємо тільки до символічних полів типу CHAR або VARCHAR (див. розділ 9.3 «Типи даних SQL»).

Цей оператор переглядає строкові значення полів з метою визначення, чи входить заданий в операторі LIKE підрядок (зразок пошуку) у символічному рядку - значення поля, що перевіряється.

Для вибірки строкових значень по заданому зразку можна застосовувати шаблон шуканого зразка рядка, що використовує наступні символи:

- символ знак питання «?», зазначений у шаблоні, визначає можливість наявності в зазначеному місці *одного будь-якого* символу;
- символ «\*» допускає присутність у зазначеному місці рядка послідовності, що перевіряється, будь-яких символів довільної довжини.

**Приклад.**

Написати запит, що вибирає з таблиці STUDENT дані про студентів, прізвище яких починаються на букву «Р».

```
SELECT *
FROM STUDENT
WHERE SURNAME LIKE 'Р*';
```

У разі потреби включення в зразок самих символів «?» і «\*» застосовують так звані *escape-символи*. Якщо escape-символ передає знакові «?» і «\*», то ці



знаки будуть сприйматися буквально. Наприклад, можна задати зразок пошуку за допомогою наступного виразу:

```
LIKE '?\?P' ESCAPE 'V'.
```

У цьому виразі символ 'V' за допомогою ключового слова ESCAPE є escape-символом. Перший символ «?» у заданому шаблоні пошуку '?\?P' буде відповідати, як і раніше, будь-якому символу в рядку, що перевіряється. Однак другий символ «?», що впливає після символу 'V', оголошеного escape-символом, вже буде інтерпретуватися буквально як звичайний символ, так само як і символ 'P' у заданому шаблоні.

Звертаємо увагу на те, що розглянуті вище оператори порівняння «=, <, >, <=, >=, <>» і оператори IN, BETWEEN і LIKE ні в якому разі не можна використовувати для перевірки вмісту поля на наявність у ньому порожнього значення NULL (див. розділ 9.3 «Типи даних SQL»). Для цих цілей призначені спеціальні оператори IS NULL (є порожнім) і IS NOT NULL (не є порожнім).

## 9.8. Числові і символні константи

Незважаючи на те, що SQL працює з даними в поняттях рядків і стовпців таблиць, мається можливість застосування значень виразів, побудованих з використанням вбудованих функцій, констант, імен стовпців, обумовлених як свого роду віртуальні стовпці. Вони містяться в списку стовпців і можуть супроводжуватися псевдонімами.

Якщо в запиті замість специфікації стовпця SQL виявляє *число*, то воно інтерпретується як *числова константа*.

*Символьні константи* повинні вказуватися в одинарних лапках. Якщо одинарні лапки повинні виводитися як частина строкової константи, то її потрібно випередити іншими одинарними лапками.

Наприклад, результатом виконання запиту:

```
SELECT 'ПРИЗВИЩЕ', SURNAME, 'Ім'я', NAME, 100  
FROM STUDENT;
```

є таблиця наступного виду:

Таблиця 9.12

	SURNAME		NAME	
Прізвище	Іванов	Ім'я	Іван	100
Прізвище	Петров	Ім'я	Петро	100
Прізвище	Сидоров	Ім'я	Вадим	100
Прізвище	Кузнєцов	Ім'я	Борис	100
Прізвище	Зайцева	Ім'я	Ольга	100
Прізвище	Павлов	Ім'я	Андрій	100
Прізвище	Котів	Ім'я	Павло	100
Прізвище	Лукин	Ім'я	Артем	100
Прізвище	Петров	Ім'я	Антон	100
Прізвище	Белкін	Ім'я	Вадим	100

## 9.9. Арифметичні операції для перетворення числових даних

- Унарний (одиначний) оператор «-» (знак мінус) змінює знак числового значення, перед яким він зазначений, на протилежний.

- Бінарні оператори «+», «-», «\*» і «/» надають можливість виконання арифметичних операцій додавання, вирахування, множення і розподілу. Наприклад, результат запиту:

```
SELECT SURNAME, NAME, STIPEND, (STIPEND*KURS)/2
FROM STUDENT WHERE KURS = 4 AND STIPEND > 0;
```

виглядає в такий спосіб:

Таблиця 9.13

SURNAME	NAME	STIPEND	KURS	
Сидоров	Вадим	150	4	-300
Петров	Антон	200	4	-400

Операція конкатенації «+» дозволяє з'єднувати («склеювати») значення двох або більш стовпців символічного типу або символічних констант в один рядок.

Ця операція має синтаксис:

<значимий символічний вираз > {+} <значимий символічний вираз>.

Наприклад:

```
SELECT SURNAME + '_' + NAME, STIPEND FROM STUDENT
WHERE KURS = 4 AND STIPEND > 0;
```

Результат запиту буде виглядати в такий спосіб:

Таблиця 9.14

	STIPEND
Сидоров_Вадим	150
Петров_Антон	200

## 9.10. Функції роботи з числами

- ABS – абсолютне значення, ABS (<значимий числовий вираз>);
- FLOOR – урізує значення числа з крапкою, що плаває, до найбільшого цілого, що не перевершує задане число, FLOOR (<значимий числовий вираз >);
- CEIL – найменшого цілого, рівного або більше заданого числа CEIL (<значимий числовий вираз >);
- Функція округлення – ROUND,  
ROUND (<значимий числовий вираз >, <точність>) аргумент <точність> задає точність округлення (див. приклад нижче);
- Функція усікання – TRUNC,  
TRUNC (<значимий числовий вираз >, <точність>);
- Тригонометричні функції: COS (<значимий числовий вираз >); SIN (<значимий числовий вираз >); TAN (<значимий числовий вираз >).
- Гіперболічні функції: COSH (<значимий числовий вираз >); SINH (<значимий числовий вираз >); TANH (<значимий числовий вираз >).

- Експонентна функція: EXP (<значимий числовий вираз >).
- Логарифмічні функції – LN, LOG: LN (<значимий числовий вираз >); LOG (<значимий числовий вираз >).
- Функція зведення в ступінь – POWER, POWER (<значимий числовий вираз >, <експонента>).
- Визначення знака числа – SIGN, SIGN (<значимий числовий вираз >).
- Обчислення квадратного кореня – SQRT, SQRT (<значимий числовий вираз >).

Приклад запиту:

```
SELECT UN IV NAME, RATING, ROUND (RATING, 1),
TRUNC (RATING) FROM UNIVERSITY;
```

поверне результат:

Таблиця 9.15

Перетворення висновку й вбудовані функції

UN IV NAME	RATING		
КГУ	606	610	600
ВГУ	296	300	290
НГУ	345	350	340
РГУ	416	420	410
БГУ	326	330	320
ТГУ	368	370	360
ВГМА	327	330	320

### 9.11. Агрегатовані і групові функції

Агрегатовані функції дозволяють одержувати з таблиці зведену (агреговану) інформацію, виконуючи операції над групою рядків таблиці. Для задання в SELECT-запиті агрегатованих операцій використовуються наступні ключові слова:

- COUNT визначає кількість рядків або значень поля, обраних за допомогою запиту і не є NULL-значенням;
- SUM обчислює арифметичну суму всіх обраних значень даного поля;
- AVG обчислює середнє значення для всіх обраних значень даного поля;
- MAX обчислює найбільше з усіх обраних значень даного поля;
- MIN обчислює найменше з усіх обраних значень даного поля.

У SELECT - запиті агрегатовані функції використовуються аналогічно іменам полів, при цьому останні (імена полів) використовуються як аргументи цих функцій.

Функція AVG призначена для підрахунку середнього значення поля на безлічі записів таблиці.

Наприклад, для визначення середнього значення поля MARK (оцінки) за всіма записами таблиці EXAM\_MARKS можна використовувати запит з функцією AVG наступного виду:

```
SELECT AVG (MARK) FROM EXAM_MARKS;
```

Для підрахунку загальної кількості рядків у таблиці варто використовувати функцію COUNT із зірочкою.

```
SELECT COUNT* FROM EXAM_MARKS;
```

Аргументи DISTINCT і ALL дозволяють, відповідно, виключати і включати дублікати оброблюваних функцією COUNT значень, при цьому необхідно враховувати, що при використанні опції ALL значення NULL все одно не ввійдуть в число значень, що підраховуються.

```
SELECT COUNT (DISTINCT SUBJ_ID ) FROM SUBJECT;
```

Речення GROUP BYGROUP BY (групувати по) дозволяє групувати записи в підмножини, обумовлені значеннями будь-якого поля, і застосовувати Transact-SQL функції вже не до всіх записів таблиці, а роздільно до кожної сформованої групи.

Припустимо, потрібно знайти максимальне значення оцінки, отриманої кожним студентом. Запит буде виглядати в такий спосіб:

```
SELECT STUDENT_ID, MAX (MARK) FROM EXAM_MARKS GROUP BY  
STUDENT_ID;
```

Обрані з таблиці EXAM\_MARKS записи групуються за значеннями поля STUDENT\_ID, зазначеного в пропозиції GROUP BY, і для кожної групи шукається максимальне значення поля MARK. Речення GROUP BY дозволяє застосовувати агрегативні функції до кожної групи, обумовленої загальним значенням поля (або полів), зазначених у цій пропозиції. У даному запиті розглядаються групи записів, згруповані по ідентифікаторах студентів.

У конструкції GROUP BY для групування може бути використане більш одного стовпця. Наприклад:

```
SELECT STUDENT_ID, SUBJ_ID, MAX (MARK)  
FROM EXAM_MARKS GROUP BY  
STUDENT_ID, SUBJ_ID;
```

У цьому випадку рядки спочатку групуються за значеннями першого стовпця, а усередині цих груп – у підгрупи за значеннями другого стовпця. Таким чином, GROUP BY не тільки встановлює стовпці, за якими здійснюється групування, але і вказує порядок розбивки стовпців на групи.

Варто мати на увазі, що в пропозиції GROUP BY повинні бути зазначені всі обрані стовпці, приведені після ключового слова SELECT, крім стовпців, зазначених як аргумент в агрегативних функціях.

При необхідності частина сформованих за допомогою GROUP BY груп може бути виключена за допомогою пропозиції HAVING.

Речення HAVING визначає критерій, по якому групи варто включати у вихідні дані, за аналогією з пропозицією WHERE, що здійснює це для окремих рядків.

```
SELECT SUBJ_NAME, MAX (HOUR) FROM SUBJECT GROUP  
BY SUBJ_NAME HAVING MAX (HOUR) >= 72;
```

В умові, що задається пропозицією HAVING, указують тільки поля або вирази, що на виході мають єдине значення для кожної виведеної групи.

## 9.12. Результат дії тризначних умовних операторів

Умовні оператори при відсутності порожніх значень повертають або TRUE (істина), або FALSE (неправда). Якщо ж у стовпці присутні порожні значення, то може бути повернуте і третє значення: UNKNOWN (невідомо). У цій схемі, наприклад, умова WHERE A = 2, де A – ім'я стовпця, значення якого можуть бути невідомі, при A = 2 буде відповідати TRUE, при A = 4 в результаті буде отримане значення FALSE, а при відсутнім значенні A (NULL-значення) результат буде UNKNOWN. Порожні значення впливають на використання логічних операторів NOT, AND і OR.

### Оператор NOT.

Звичайний унарний оператор NOT звертає оцінку TRUE у FALSE і навпаки. Однак NOT NULL як і раніше буде повертати порожнє значення NULL. При цьому варто відрізнити випадок NOT NULL від умови IS NOT NULL, що є протилежністю IS NULL, відокремлюючи відомі значення від невідомих.

### Оператор AND.

- Якщо результат двох умов, об'єднаних оператором AND, відомий, то застосовуються правила булевої логіки, тобто при обох твердженнях TRUE складене твердження також буде TRUE. Якщо ж хоча б одне з двох тверджень буде FALSE, то складене твердження буде FALSE.
- Якщо результат одного з тверджень невідомий, а інший оцінюється як TRUE, то стан невідомого твердження є визначальним, і, отже, підсумковий результат також невідомий.
- Якщо результат одного з тверджень невідомий, а інший оцінюється як FALSE, підсумковий результат буде FALSE.
- Якщо результат обох тверджень невідомий, то результат також залишається невідомим.

### Оператор OR.

- Якщо результат двох умов, об'єднаних оператором OR, відомий, то застосовуються правила булевої логіки, а саме: якщо хоча б одне з двох тверджень відповідає TRUE, то і складене твердження буде TRUE, якщо обидва твердження оцінюються як FALSE, то складене твердження буде FALSE.
- Якщо результат одного з тверджень невідомий, а інший оцінюється як TRUE, підсумковий результат буде TRUE.
- Якщо результат одного з тверджень невідомий, а інший оцінюється як FALSE, то стан невідомого твердження має визначальне значення. Отже, підсумковий результат також невідомий. Якщо результат обох тверджень невідомий, то результат також залишається невідомим.

**Примітка:**

Відсутні (NULL) значення доцільно використовувати в стовпцях, призначених для агрегування, щоб витягти переваги зі способу обробки порожніх значень у функціях COUNT і AVERAGE. Практично в усіх інших випадках порожніх значень варто уникати, тому що при їх наявності істотно ускладнюється коректна побудова умов добору, яка іноді призводить до непередбачених результатів вибірки. Для індикації ж відсутніх, непридатних або з якоїсь причини невідомих даних можна використовувати значення за замовчуванням, які встановлювані заздалегідь (наприклад, за допомогою команди CREATE TABLE).

### 9.13. Вкладені підзапити

SQL дозволяє використовувати одні запити усередині інших запитів, тобто вкладати запити один в одного. Припустимо, відоме прізвище студента («Петров»), але невідоме значення поля STUDENT\_ID для нього. Щоб витягти дані про всі оцінки цього студента, можна записати наступний запит:

```
SELECT *  
FROM EXAM_MARKS WHERE STUDENT_ID =(SELECT  
STUDENT_ID FROM STUDENT SURNAME = 'Петров');
```

Як працює запит SQL зі зв'язаним підзапитом?

- Вибирається рядок з таблиці, ім'я якого зазначене в зовнішньому запиті.
- Виконується підзапит і отримане значення застосовується для аналізу цього рядка в умові пропозиції WHERE зовнішнього запиту.
- По результату оцінки цієї умови приймається рішення про включення або виключення рядка до складу вихідних даних.
- Процедура повторюється для наступного рядка таблиці зовнішнього запиту.

Варто звернути увагу, що приведений вище запит коректний тільки в тому випадку, якщо в результаті виконання зазначеного в дужках підзапиту повертається *єдине значення*. Якщо в результаті виконання підзапиту буде повернуто кілька значень, то цей підзапит буде помилковим. У даному прикладі це відбудеться, якщо в таблиці STUDENT буде кілька записів зі значеннями поля SURNAME = 'Петров'.

У деяких випадках для гарантії одержання єдиного значення в результаті виконання підзапиту використовується DISTINCT. Одним з видів функцій, що автоматично *завжди* видають в результаті єдине значення для будь-якої кількості рядків, є агрегатована функція.

Оператор IN також широко застосовується в підзапитах. Він задає список значень, з якими порівнюються інші значення для визначення істинності предиката, що задається цим оператором.

Дані про всі оцінки (таблиця EXAM\_MARKS) студентів з Дніпропетровська можна вибрати за допомогою наступного запиту:

```
SELECT *  
FROM EXAM_MARKS WHERE STUDENT_ID IN
```

```
(SELECT STUDENT_ID FROM STUDENT WHERE CITY =  
'Дніпропетровськ');
```

Підзапити можна застосовувати всередині пропозиції HAVING. Нехай потрібно визначити кількість предметів навчання з оцінкою, що перевищує середнє значення оцінки студента з ідентифікатором 301:

```
SELECT COUNT (DISTINCT SUBJ_ID , MARK)  
FROM EXAM_MARKS GROUP BY MARK HAVING MARK >  
(SELECT AVG(MARK)  
FROM EXAM_MARKS WHERE STUDENT_ID = 301) ;
```

#### 9.14. Формування зв'язаних підзапитів

При використанні підзапитів у внутрішньому запиті можна посилатися на таблицю, ім'я якої зазначено в пропозиції FROM зовнішнього запиту. У цьому випадку такий *зв'язаний* підзапит виконується один раз для *кожного* рядка таблиці основного запиту.

**Приклад:** вибрати зведення про всі предмети навчання, по яких проводився іспит 20 січня 1999 р.

```
SELECT *  
FROM SUBJECT SU WHERE '20/01/1999' IN  
(SELECT EXAM_DATE FROM EXAM_MARKS EX  
WHERE SU.SUBJ_ID = EX.SUBJ_ID);
```

У деяких СУБД для виконання цього запиту може знадобитися перетворення значення дати в символічний тип. У приведенному запиті SU і EX є псевдонімами, тобто іменами, які спеціально вводяться та можуть бути використані в даному запиті замість дійсних імен. У приведенному прикладі вони використовуються замість імен таблиць SUBJECT і EXAM\_MARKS.

Цю саму задачу можна вирішити за допомогою операції з'єднання таблиць:

Можна використовувати підзапити, що зв'язують таблицю зі своєю власною копією. Наприклад, треба знайти ідентифікатори, прізвища і стипендії студентів, що одержують стипендію вище середньої на курсі, на якому вони навчаються.

```
SELECT DISTINCT STUDENT_ID, SURNAME, STIPEND FROM  
STUDENT E1 WHERE STIPEND > (SELECT AVG ( STIPEND)  
FROM STUDENT E2 WHERE E1.KURS = E2.KURS);
```

Той же результат можна одержати за допомогою наступного запиту:

```
SELECT DISTINCT STUDENT_ID , SURNAME , STI PEND  
FROM STUDENT E1, (SELECT KURS, AVG (STIPEND) AS  
AVG_STIPEND FROM STUDENT E2 GROUP BY E2 .KURS) E3 WHERE  
E1. STIPEND > AVG_STIPEND AND E1 .KURS=E3 .KURS;
```

Зверніть увагу — другий запит буде виконаний набагато швидше. Справа в тім, що в першому варіанті запиту агрегатована функція AVG виконується над таблицею, зазначеною в підзапиті, для *кожного* рядка зовнішнього запиту. В іншому варіанті друга таблиця (псевдонім E2) обробляється агрегати-

ваною функцією один раз, в результаті чого формується допоміжна таблиця (у запиті вона має псевдонім E3), з рядками якої потім з'єднуються рядки першої таблиці (псевдонім E1). Варто мати на увазі, що реальний час виконання запиту у великому ступені залежить від оптимізатора запитів конкретної СУБД.

### 9.15. Зв'язані підзапити в HAVING

У пп. 9.9 вказувалося, що речення GROUP BY дозволяє групувати виведені SELECT-запитом записи за значенням якогось поля. Використання пропозиції HAVING дозволяє при висновку здійснювати фільтрацію таких груп. Предикат пропозиції HAVING оцінюється не для кожного рядка результату, а для кожної групи вихідних записів, сформованої пропозицією GROUP BY зовнішнього запиту.

Нехай, наприклад, необхідно за даними з таблиці EXAM\_MARKS визначити суму отриманих студентами оцінок (значень поля MARK), згрупувавши значення оцінок за датами іспитів та виключив ті дні, коли число студентів, що здавали протягом дня іспити, було менше 10.

```
SELECT EXAM_DATE, SUM (MARK)
FROM EXAM_MARKS A GROUP BY EXAM_DATE HAVING
10 < (SELECT COUNT (MARK) FROM EXAM_MARKS Y
WHERE A.EXAM_DATE = B.EXAM_DATE);
```

Підзапит обчислює кількість рядків з однієї і тією ж датою, що збігається з датою, для якої сформована чергова група основного запиту.

### 9.16. Використання оператора EXISTS

Використовуваний у SQL оператор EXISTS (існує) генерує значення істина або неправда, подібно булевому виразу. Використовуючи підзапити як аргумент, цей оператор оцінює результат виконання підзапита як щирий, якщо цей підзапит генерує вихідні дані, тобто у випадку існування (повернення) хоча б одного знайденого значення. У протилежному випадку результат підзапита помилковий. Оператор EXISTS не може приймати значення UNKNOWN (не відомо).

Нехай, наприклад, потрібно витягти з таблиці EXAM\_MARK дані про студентів, що одержали хоча б одну незадовільну оцінку.

```
SELECT DISTINCT STUDENT_ID
FROM EXAM_MARKS A WHERE EXISTS
(SELECT *
FROM EXAM_MARKS Y WHERE MARK < 3
AND B.STUDENT_ID = A. STUDENT_ID);
```

При використанні зв'язаних підзапитів речення EXISTS аналізує кожен рядок таблиці, на який мається посилання в зовнішньому запиті. Головний запит одержує строки-кандидати на перевірку умови. Для кожного рядка-кандидата виконується підзапит. Як тільки підзапит знаходить рядок, де в



стовпці MARK значення задовольняє умові, вона припиняє виконання і повертає значення істина зовнішньому запитові, що потім аналізує свій рядок-кандидат. Наприклад, потрібно одержати ідентифікатори предметів навчання, екзамену з яких здавалися не одним, а декількома студентами:

```
SELECT DISTINCT SUBJ_ID
FROM EXAM_MARKS A WHERE EXISTS (SELECT * FROM
EXAM_MARKS Y WHERE A.SUBJ_ID = Y.SUBJ_ID
AND A.STUDENT_ID < > Y.STUDENT_ID);
```

Часто EXISTS застосовується з оператором NOT (за російською NOT EXISTS перекладається, як «не існує»). Якщо попередній запит сформулювати в такий спосіб — знайти ідентифікатори предметів навчання, що здавалися одним, і тільки одним студентом (іншими словами, для яких не існує іншого студента, що здавав,), то досить просто поставити NOT перед EXISTS.

Варто мати на увазі, що в підзапиті, що вказується в операторі EXISTS, *не можна використовувати агрегативовані функції*.

Можливості застосування вкладених запитів досить різноманітні. Наприклад, нехай з таблиці STUDENT потрібно витягти рядок для кожного студента, що здав більш одного предмета.

```
SELECT *
FROM STUDENT FIRST WHERE EXISTS
(SELECT SUBJ_ID FROM EXAM_MARKS SECOND
GROUP BY SUBJ_ID HAVING COUNT (SUBJ_ID) > 1
WHERE FIRST. STUDENT_ID = SECOND. STUDENT_ID);
```

## 9.17. Оператор об'єднання UNION

Оператор UNION використовується для об'єднання вихідних даних двох або більш SQL-запитів у єдину безліч рядків і стовпців. Наприклад, для того щоб одержати в одній таблиці прізвища й ідентифікатори студентів і викладачів з Москви, можна використовувати наступний запит:

```
SELECT 'Студент', SURNAME, STUDENT_ID
FROM STUDENT
WHERE CITY = 'Київ' UNION SELECT 'Викладач', SURNAME,
LECTURER_ID
FROM LECTURER
WHERE CITY = 'Київ';
```

Зверніть увагу на те, що символом «;» (крапка з комою) закінчується тільки останній запит. Відсутність цього символу наприкінці SELECT-запиту означає, що наступний за ним запит так само, як і він, є частиною загального запиту з UNION.

Використання оператора UNION можливо тільки при об'єднанні запитів, стовпці яких *сумісні по об'єднанню*, тобто відповідні числові поля повинні мати цілком співпадаючий тип і розмір, символічні поля повинні мати точно співпадаючу кількість символів. Якщо NULL-значення заборонені для стовп-

ця хоча б одного будь-якого підзапиту об'єднання, то вони повинні бути заборонені і для усіх відповідних стовпців в інших підзапитах об'єднання.

### 9.18. Усунення дублювання в UNION

На відміну від звичайних запитів UNION автоматично виключає з вихідних дані дублікати рядків, наприклад, у запиті,

```
SELECT CITY FROM STUDENT UNION  
SELECT CITY FROM LECTURER;
```

співпадаючі найменування міст будуть виключені.

Якщо все-таки необхідно в кожному запиті вивести всі рядки незалежно від того, чи маються *такі* ж рядки в інших поєднаних запитах, то варто використовувати в множинному запиті конструкцію з оператором UNION ALL. Так, у запиті,

```
SELECT CITY FROM STUDENT UNION ALL  
SELECT CITY FROM LECTURER;
```

дублікати значень міст, виведені другою частиною запиту, не будуть виключатися.

### 9.19. Використання UNION з ORDER BY

Речення ORDER BY застосовується для упорядкування вихідних дані об'єднання запитів так само, як і для окремих запитів. Останній приклад, при необхідності упорядкування вихідних даних запиту по прізвищах студентів і датам іспитів, може виглядати в такий спосіб:

```
SELECT 'МАКСОЦ', A.STUDENT_ID, SURNAME, E.MARK,  
E.EXAM_DATE FROM STUDENT A, (SELECT B.STUDENT_ID,  
B.MARK, B.EXAM_DATE FROM EXAM_MARKS B, (SELECT MAX  
(MARK) AS MAX_MARK, C.EXAM_DATE FROM EXAM_MARKS 3  
GROUP BY C.EXAM_DATE) D WHERE  
B.EXAM_DATE=D.EXAM_DATE AND B.MARK=MAX_MARK) E  
WHERE A.STUDENT_ID=E.STUDENT_ID UNION ALL  
SELECT 'МИНОЦ', A.STUDENT_ID, SURNAME, E.MARK,  
E.EXAM_DATE FROM STUDENT A, (SELECT B.STUDENT_ID,  
B.MARK, B.EXAM_DATE FROM EXAM_MARKS B,  
(SELECT MIN(MARK) AS MIN_MARK, C.EXAM_DATE FROM  
EXAM_MARKS 3 GROUP BY C.EXAM_DATE) D WHERE  
B.EXAM_DATE=D.EXAM_DATE AND B.MARK=MIN_MARK) E WHERE  
A.STUDENT_ID=E.STUDENT_ID ORDER BY SURNAME, .EXAM_DATE;
```

### 9.20. Зовнішнє об'єднання

Часто буває, корисна операція об'єднання двох запитів, в якій другий запит вибирає рядки, виключені першими. Така операція називається зовнішнім об'єднанням.

Розглянемо приклад. Нехай у таблиці STUDENT є записи про студентів, у яких не зазначений ідентифікатор університету. Потрібно скласти список студентів із указівкою найменування університету. З'єднання, що використовує предикати, засновані на рівностях, називається *еквіз'єднанням*. Розглянутий приклад з'єднання таблиць відноситься до виду так званого *внутрішнього (INNER) з'єднання*. При цьому з'єднуються тільки ті рядки таблиць, для яких щирим є предикат, що задається в пропозиції ON виконуваного запиту.

Ключове слово INNER у запиті може бути опущено, тому що ця опція в операторі JOIN діє за замовчуванням.

Приклад повного з'єднання (декартовий добуток) таблиць з використанням ключового слова JOIN буде виглядати в такий спосіб:

```
SELECT * FROM STUDENT JOIN UNIVERSITY;
```

що еквівалентно

```
SELECT * FROM STUDENT, UNIVERSITY;
```

Помітимо, що в СУБД Oracle задається стандартом мови SQL оператор JOIN не підтримується.

## 9.21. Команди маніпулювання даними

У SQL для виконання операцій введення даних у таблицю, їх зміни і видалення призначені три команди мови маніпулювання даними (DML). Це команди INSERT (вставити), UPDATE (оновити), DELETE (видалити).

Команда INSERT здійснює вставку в таблицю нового рядка. У найпростішому випадку вона має вигляд:

```
INSERT INTO <ім'я таблиці> VALUES (<значення>, <значення>,);
```

При такому записі зазначені в дужках після ключового слова VALUES значення вводяться в поля доданого в таблицю нового рядка в тім порядку, в якому відповідні стовпці зазначені при створенні таблиці, тобто в операторі CREATE TABLE.

Наприклад, введення нового рядка в таблицю STUDENT може бути здійснено у такий спосіб:

```
INSERT INTO STUDENT
```

```
VALUES (101,'Іванов','Олександр', 200, 3,'Київ'1, '6/10/1979', 15);
```

Щоб така команда могла бути виконана, таблиця з зазначеним у ній ім'ям (STUDENT) повинна бути попередньо визначена (створена) командою CREATE TABLE. Якщо в яке-небудь поле необхідно вставити NULL-значення, то воно вводиться як звичайне значення:

```
INSERT INTO STUDENT
```

```
VALUES (101, 'Іванов', NULL, 200, 3, 'Київ','6/10/1979', 15);
```

У випадках, коли необхідно ввести значення полів у порядку, відмінному від порядку стовпців, заданого командою CREATE TABLE, або потрібно ввести значення не в усі стовпці, варто використовувати наступну форму команди INSERT:

```
INSERT INTO STUDENT (STUDENT_ID, CITY, SURNAME, NAME)  
VALUES (101, 'Київ', 'Іванов', 'Сашко');
```

Стовпцям, найменування яких не зазначені в приведеному в дужках списку, автоматично привласнюється значення за замовчуванням, якщо воно призначено при описі таблиці (команда CREATE TABLE), або значення NULL.

За допомогою команди INSERT можна витягти значення з однієї таблиці і розмістити його в іншій, наприклад, запитом наступного виду:

```
INSERT INTO STUDENT1  
SELECT * FROM STUDENT1 WHERE CITY = 'Київ';
```

При цьому таблиця STUDENT1, повинна бути попередньо створена командою CREATE TABLE, і мати структуру, ідентичну таблиці STUDENT.

Видалення рядків з таблиці здійснюється за допомогою команди DELETE.

Наступний вираз видаляє всі рядки таблиці EXAM\_MARKS1.

```
DELETE FROM EXAM_MARKS1;
```

В результаті таблиця стає порожньою (після цього вона може бути вилучена командою DROP TABLE).

Для видалення з таблиці відразу декількох рядків, що задовольняють деякій умові, можна скористатися пропозицією WHERE, наприклад:

```
DELETE FROM EXAM_MARKS1 WHERE STUDENT_ID = 103;
```

Можна видалити групу рядків:

```
DELETE FROM STUDENT1 WHERE CITY = 'Київ';
```

Команда UPDATE дозволяє змінювати, тобто обновляти значення деяких або всіх полів в існуючому рядку або рядках таблиці. Наприклад, щоб для всіх університетів, дані про які знаходяться в таблиці UNIVERSITY1, змінити рейтинг на значення 200, можна використовувати конструкцію:

```
UPDATE UNIVERSITY1 SET RATING = 200;
```

Для вказування конкретних рядків таблиці, значення полів яких повинні бути змінені, у команді UPDATE можна використовувати предикат, що вказується в пропозиції WHERE.

```
UPDATE UNIVERSITY1 SET RATING = 200 WHERE CITY = 'Київ';
```

В результаті виконання цього запиту буде змінений рейтинг тільки в університетів, розташованих у Києві.

Команда UPDATE дозволяє змінювати не тільки один, але і безліч стовпців. Для вказівки конкретних стовпців, значення яких повинні бути модифіковані, використовується речення SET.

Наприклад, найменування предмета навчання 'Математика' (для нього SUBJ\_ID = 43) повинне бути замінене на назву 'Вища математика', при цьому ідентифікаційний номер необхідно зберегти, але у відповідні поля рядка таблиці ввести нові дані про цей предмет навчання. Запит буде виглядати в такий спосіб:

```
UPDATE SUBJECT1  
SET SUBJ_NAME = Вища математика, HOUR = 36, SEMESTER = 1  
WHERE SUBJ_ID = 43;
```

У пропозиції SET команди UPDATE можна використовувати скалярні вирази, що вказують спосіб зміни значень поля, у які можуть входити значення змінюваного й іншого полів.

```
UPDATE UNIVERSITY1 SET RATING = RATING*2;
```

Наприклад, для збільшення в таблиці STUDENT значення поля STIPEND у два рази для студентів з Києва можна використовувати запит:

```
UPDATE STUDENT 1  
SET STIPEND = STIPEND*2  
WHERE CITY = 'Київ';
```

Речення SET не є предикатом, тому в ньому можна вказати значення NULL у такий спосіб:

```
UPDATE UNIVERSITY 1 SET RATING = NULL WHERE CITY = 'Київ';
```

## 9.22. Використання підзапитів у INSERT

Застосування оператора INSERT з підзапитом дозволяє завантажувати відразу кілька рядків в одну таблицю, використовуючи інформацію з іншої таблиці. У той час як оператор INSERT, що використовує VALUES, додає тільки один рядок, INSERT з підзапитом додає в таблицю стільки рядків, скільки підзапит витягає з іншої таблиці. При цьому кількість і тип стовпців, які повертаються підзапитом повинна відповідати кількості і типові стовпців таблиці, у яку вставляються дані.

Наприклад, нехай таблиця STUDENT1 має структуру, що цілком збігається зі структурою таблиці STUDENT. Запит, що дозволяє заповнити таблицю STUDENT1 записами про всіх студентів з Москви з таблиці STUDENT, виглядає в такий спосіб:

```
INSERT INTO STUDENT1  
SELECT * FROM STUDENT WHERE CITY = 'Київ';
```

Для того, щоб додати в таблицю STUDENT1 дані про всіх студентів, що *навчались* в Москві, можна використовувати в пропозиції WHERE відповідний підзапит. Наприклад,

```
INSERT INTO STUDENT1  
SELECT * FROM STUDENT WHERE UNIV_ID IN  
(SELECT UNIV_ID FROM UNIVERSITY WHERE CITY = 'Київ');
```

## 9.23. Використання підзапитів з DELETE

Нехай філія університету у Васильковці ліквідована і потрібно видалити з таблиці STUDENT записи про студентів, які там навчалися. Цю операцію можна виконати за допомогою запиту:

```
DELETE  
FROM STUDENT WHERE UNIV_ID IN (SELECT UNIV_ID FROM  
UNIVERSITY WHERE CITY = 'Васильковці');
```

У предикаті пропозиції FROM (підзапиту) не можна посилатися на таблицю, з якої здійснюється видалення. Однак можна посилатися на поточний рядок з таблиці, що є кандидатом на видалення, тобто на рядок, що у даний час перевіряється в основному предикаті.

```
DELETE FROM STUDENT WHERE EXISTS (SELECT *
```

```
FROM UNIVERSITY1 WHERE RATING = 401
AND STUDENT.UNIV_ID = UNIVERSITY.UNIV_ID);
```

Частина AND предиката внутрішнього запиту посиляється на таблицю STUDENT. Команда видаляє дані про студентів, що навчаються в університетах, що мають рейтинг, рівний 401. Існують і інші способи вирішення цієї задачі.

```
DELETE
FROM STUDENT WHERE 401 IN (SELECT RATING FROM
UNIVERSITY WHERE STUDENT.UNIV_ID = UNIVERSITY.UNIV_ID);
```

Нехай потрібно знайти найменше значення оцінки, отриманої в щодня здачі іспитів, і видалити з таблиці дані про студента, що одержав цю оцінку. Запит буде мати вигляд:

```
DELETE
FROM STUDENT WHERE STUDENT_ID IN
(SELECT STUDENT_ID FROM EXAM_MARKS A WHERE MARK =
(SELECT MIN (MARK) FROM EXAM_MARKS
WHERE A.EXAM_DATE = B.EXAM_DATE));
```

Тому що стовпець STUDENT\_ID є первинним ключем, то видаляється єдиний рядок.

Якщо в якийсь день складався тільки один іспит (тобто отримана тільки одна мінімальна оцінка) і з якої-небудь причини запис, в якому знаходиться ця оцінка, потрібно залишити, то рішення буде мати вигляд:

```
DELETE
FROM STUDENT WHERE STUDENT_ID IN
(SELECT STUDENT_ID FROM EXAM_MARKS A WHERE MARK =
(SELECT MIN (MARK) FROM EXAM_MARKS
WHERE A.EXAM_DATE = B.EXAM_DATE AND
1 < (SELECT COUNT (SUBJ_ID) FROM
EXAM_MARKS WHERE A.EXAM_DATE =
B.EXAM_DATE));
```

## 9.24. Використання підзапитів з UPDATE

За допомогою команди UPDATE можна застосовувати підзапити в будь-якій формі, прийнятній для команди DELETE.

Наприклад, використовуючи зв'язані підзапити, можна збільшити значення розміру стипендії на 20 у записах студентів, що здала іспити на 4 і 5.

```
UPDATE STUDENT1
SET STIPEND = STIPEND + 20
WHERE 4 >= (SELECT MIN(MARK) FROM EXAM_MARKS
WHERE EXAM_MARKS.STUDENT_ID = STUDENT1 .STUDENT_ID);
```

Інший запит: «Зменшити величину стипендії на 20 усім студентам, що одержали на іспиті мінімальну оцінку».

```
UPDATE STUDENT1 SET STIPEND = STIPEND - 20
WHERE STUDENT_ID IN (SELECT STUDENT_ID
FROM EXAM_MARKS A WHERE MARK
```

```
(SELECT MIN(MARK) FROM EXAM_MARKS У  
WHERE A.EXAM_DATE = B.EXAM_DATE));
```

### 9.25. Індивідуальне завдання №9. Вивчення середовища СУБД Access

- Ціль:** 1. Вивчити середовище СУБД Access, включаючи систему меню, вікна, панелі інструментів і довідкову систему.
2. Одержати навички в роботі з файлами БД, таблицями, запитам, формами і звітами демонстраційної БД Борею.
3. Освоїти проектування бази даних, підготовку даних для БД свого варіанта.

#### **Проектування і підготовка даних для бази даних свого варіанта.**

**Вхідні дані:** Індивідуальні завдання на створення бази даних (варіанти баз даних).

#### **Перелік робіт:**

- коротко описати фрагмент предметної області;
- скласти та заповнити реальними даними по одному екземпляру вхідних документів, що є джерелами даних;
- скласти часткові універсальні таблиці, що містять усі поля проектованої бази даних;
- виконати нормалізацію універсальних таблиць, склавши список таблиць;
- скласти логічну структуру бази даних, індивідуального варіанта;
- заповнити всі таблиці бази даних вхідними даними по 10-15 рядків;
- для кожної таблиці скласти її структуру (опис таблиці);
- скласти логічну структуру (модель) бази даних;
- скласти списки таблиць, запитів, форм і звітів, що підлягають створенню в базі даних.

### **ВАРІАНТИ**

**Варіант №1.** База даних *Викладачі* створюється для одноіменного фрагмента предметної області *Навчальний процес*.

**Варіант №2.** База даних *Облік роботи і заохочення співробітників*.

**Варіант №3.** База даних *Туристичне агентство*.

**Варіант №4.** База даних *Постачальник* для однойменного фрагмента предметної області *Склад*.

**Варіант №5.** База даних мережі магазинів, що спеціалізується на продажі.

**Варіант №6.** База даних *Курси (Викладачі)* може бути створена для курсів різного призначення, а саме: *Курси підвищення кваліфікації, Курси по перепідготовці, Курси вивчення іноземних мов, Підготовчі курси*. В усіх випадках БД повинна зберігати дані про слухачів курсів, про викладачів і досліджувані предмети. У даній темі розглядається фрагмент предметної області *Курси (Викладачі)*.

**Варіант №7.** База дані *Курси (Слухачі)* може бути створена для курсів різного призначення. У темі створюється БД для фрагмента *Слухачі* предметної області *Курси*.

**Варіант №8.** База даних *Гараж*.

**Варіант №9.** База даних *Фірма по виробництву*.

**Варіант №10.** База даних *Склад*.

**Варіант №11.** База даних *Відділ постачання*.

**Варіант №12.** База даних *Відділ збуту готової продукції*.

**Варіант №13.** Бази даних *Відділ маркетингу*.

**Варіант №14.** База даних *Уведення техніки в експлуатацію*.

**Варіант №15.** База, даних *Склад фабрики*.

**Варіант №16.** База даних *Бібліотека*.

**Варіант №17.** База даних *Устаткування підприємств*.

**Варіант №18.** База даних *Комп'ютерний центр комерційної інформації*.

**Варіант №19.** База даних *Посередницька фірма без складу*.

**Варіант №20.** База даних *Калькуляція*.

**Варіант №21.** База даних *Посередницька фірма зі складом*.

**Варіант №22.** Бази даних *Відділ кадрів*.

**Варіант №23.** База даних *Фірма (організація) по ремонту...*

**Варіант №24.** База даних *Мала частка торговельне підприємство*.

**Варіант №25.** База даних *Банківські операції*.

**Варіант №26.** База даних *Довідник підприємства*.

**Варіант №27.** База даних *Абітурієнт* створюється для фрагмента предметної області *вищій навчальний заклад*.

**Варіант №28.** База даних *Відділ головного енергетика*.

**Варіант №29.** База даних *Рекламне агентство*.

**Варіант №30.** База даних *Фірма по охороні нерухомості*.

## **9.26. Індивідуальне завдання №10. Використання SQL для розв'язувань задач БД свого варіанту**

**Ціль:** Вивчити можливості SQL мови для витягу інформації з таблиці. Навчитися пропускати або переупорядковувати стовпці, автоматично усувати надмірність даних у вашому висновку, встановлювати умови, щоб визначити, які рядки таблиці використовуються у висновку.

**Вхідні дані:** База даних *свого варіанта*.

**Вхід у середовище SQL:**

Завантажите MS Access і базу даних *свого варіанту*.

Виберіть об'єкт БД *Запити* і команду *Створення запиту в режимі конструктора*. Завантажитесь бланк запиту і вікно *Додавання таблиці*. Закрийте це вікно і виконаєте команду *Вид/Режим SQL*. Відкриється діалогове вікно *SQL*, у якому можна створювати власні *SQL* – вираження або редагувати існуючі. Піс-



ля введення або редагування *SQL* – вираження. Збережіть запит натиснувши кнопки *Пі Зберегти* потім *Запуск*.

## ВАРІАНТИ

**Варіант №1. Перелік задач, розв'язуваних у БД *Викладачі*:** скласти список викладачів навчального закладу з вказівкою *Прізвища імені та по батькові, Кафедри, Посади та Окладу*; скласти список молодих викладачів, педагогічний стаж яких менше або дорівнює 3 рокам, із вказівкою *Прізвища, Імені та по батькові, Кафедри, Посади*; скласти список предметів із вказівкою кількості закріплених за ними викладачів; видалити з БД реквізити двох викладачів, що звільнені з роботи.

**Варіант №2. Перелік задач, розв'язуваних у БД *Облік роботи і заохочення співробітників*:** скласти список співробітників фірми, із вказівкою *Коду співробітника, Прізвища імені по батькові, Посади й Окладу*; скласти звіт за системою заохочень; скласти список співробітників, що не виконують планових завдань, із вказівкою *Прізвища, імені, по батькові* співробітника, *Назви роботи, Планового завдання, Кількості виконаних робіт, Обсягу недовиконаних робіт*; скласти відомість на виплату премій із вказівкою *Прізвища імені та по батькові* співробітника, *Назви робіт, Сум премій*; скласти список співробітників із закріпленими за ними роботами з вказівкою *Коду співробітника, Прізвища, імені, по батькові, Назви робіт, Сум премій*.

**Варіант №3. Перелік задач, розв'язуваних у БД *Туристичне агентство*:** скласти список клієнтів із вказівкою коду клієнта, *Прізвища імені та по батькові*, маршруту і дати від'їзду; скласти список від'їжджаючих (у будь-який день) із вказівкою *Прізвища імені та по батькові* клієнтів, маршруту і виду транспорту; скласти відомість вартості путівки з урахуванням знижки, у перелік включити *Прізвища імені та по батькові* клієнта, маршрут, кількість днів, вартість без знижки, розмір знижки, вартість зі знижкою; скласти список клієнтів, що відпочивають за пільговими путівками зі знижкою з вказівкою *Прізвища імені та по батькові*, соціального стану і розміру знижки; скласти прайс-лист із вказівкою маршруту, ціни за один день, виду транспорту, місця проживання.

**Варіант №4. Перелік задач, розв'язуваних у БД *Постачальник*:**

скласти перелік товарів (виробів) із вказівкою найменування виробів, ціни, одиниці фізичної величини; скласти список замовників із вказівкою замовника, найменування і кількості виробів; скласти список договорів на постачання виробів із вказівкою номера і дати договору, найменування замовника і суми договору; скласти перелік відвантажених товарів із вказівкою найменування замовника, товару (виробу) і кількості (відвантажене); скласти відомість оплати товарів замовниками з вказівкою найменування замовника, номера платіжної вимоги і суми (оплачене).

**Варіант №5. Перелік задач, розв'язуваних у БД *Мережа магазинів*:** скласти список магазинів фірми з вказівкою номера і назви магазину, телефону;

скласти перелік товарів із вказівкою коду і найменування товару, оптової ціни і роздрібною ціни; скласти планову заявку для будь-якого магазину мережі; скласти термінову заявку для будь-якого магазину; скласти відомість реалізації товарів будь-якого магазину.

**Варіант №6. Перелік задач, розв'язуваних у БД *Курси*:** скласти список викладачів із вказівкою прізвищ, імен, по батькові, педагогічного стажу, ученому ступеню і посаді; скласти список викладачів пенсійного віку з вказівкою прізвищ, імен, по батькові, віку, педагогічного стажу, посади; скласти список предметів з їх найменуванням і кількістю годин (усього); скласти список предметів із вказівкою закріплених за ними викладачів; скласти список предметів, за яких закріплений тільки один викладач.

**Варіант №7. Перелік задач, розв'язуваних у БД *Курси (Слухачі)*:** скласти список викладачів курсів із вказівкою прізвищ, імен, по батькові викладачів, посади, ученому ступені; скласти список предметів із вказівкою найменування предмета і кількості годин; скласти список слухачів будь-якої групи з вказівкою прізвищ, імен, по батькові слухача, утворення, спеціальності і дати надходження; скласти відомість успішності слухачів курсів із вказівкою групи, прізвищ, імен, по батькові слухача, найменування предмета й оцінки; скласти звіт у виді посвідчення.

**Варіант №8. Перелік розв'язуваних задач у БД *Гараж*:** скласти перелік автомобілів із вказівкою типів, марок, року випуску і дати техогляду; скласти список водіїв категорії В; визначити кількість перевезень за місяць, виконаних кожним водієм; визначити витрати палива за місяць із вказівкою прізвищ, імен, по батькові водія, типу і марки автомобіля і марки палива; визначити витрати палива за місяць із вказівкою марки, кількості і вартості палива.

**Варіант №9. Перелік задач, розв'язуваних у БД *Фірма по виробництву*:** скласти перелік виробів (продукції), що випускається фірмою, вказати код виробу, найменування, місяць випуску і кількість; скласти перелік матеріалів (запчастин та ін.), необхідних для випуску продукції фірми, указати назву матеріалу, найменування і кількість виробів; скласти розрахунок собівартості продукції фірми; скласти таблицю розрахунку трудовитрат (г) на випуск виробів: найменування виробів, кількість і трудовитрат (г); скласти список постачальників, виділивши назву та адресу постачальника, назву матеріалу.

**Варіант №10. Перелік задач, розв'язуваних у БД *Склад*:** скласти перелік товарів із вказівкою найменування товарів, одиниці фізичної величини і ціни постачальника; скласти список постачальників із вказівкою постачальника, його адреси і телефону; скласти список клієнтів, що одержують даний товар; скласти список товарів, що поставляються зазначеним постачальником; обчислити вартість товарів, що поставляються кожним постачальником.

**Варіант №11. Перелік задач, розв'язуваних у БД *Відділ постачання*:** підрахунок кількості недопоставленого товару; висновок інформації про договір у зручному виді (без кодів); відновлення цін на товари в зв'язку з інфляцією; підрахунок кількості товару по категоріях; висновок інформації про замовлення по підприємству у виді електронної таблиці; видалення з бази даних товару.

**Варіант №12. Перелік задач, розв'язуваних у БД *Відділ збуту готової продукції*:** підрахунок загальної вартості і кількості виробу за накладною; список відпущених виробів зі складу за визначений проміжок часу; підрахунок вартості виробу з урахуванням ПДВ; видалення з БД замовника; визначення кількості накладних, виписаних на склад, за зазначену дату; створення бланка договору.

**Варіант №13. Перелік задач, розв'язуваних у БД *Відділ маркетингу*:** підрахувати кількість реальних покупців по кожному виду виробу; підрахувати загальну суму реклами по кожному виду виробу; додати дані про потенційного покупця в таблицю реальних покупців; видалити дані про потенційного покупця з таблиці потенційних покупців; перерахувати вартість виробів з урахуванням інфляції.

**Варіант №14. Перелік задач, розв'язуваних у БД *Уведення техніки в експлуатацію*:** скласти список устаткування, встановленого за визначений період часу; визначити суму замовлення по кожному виду устаткування; визначити загальну суму замовлення; видалити з бази даних замовника; визначити кількість оформлених замовлень кожного замовника; змінити ціни на устаткування з урахуванням інфляції.

**Варіант №15. Перелік задач, розв'язуваних у БД *Склад фабрики*:** підрахунок загальної вартості товару за накладною; список відпущених товарів зі складу за визначений проміжок часу; підрахунок загальної вартості товару по рахунку включаючи ПДВ; визначення кількості товару, відпущеного за готівку; видалення з БД замовника; визначення кількості накладних, виписаних на склад, за зазначену дату.

**Варіант №16. Перелік задач, розв'язуваних у БД *Бібліотека*:** вивести кількість книг, виданих читачам на зазначену дату; вивести кількість книг, виданих читачам за визначений проміжок часу; визначити кількість книг, що знаходяться в кожного читача; видалити з бази даних читача; визначити кількість відвідувачів у день; скласти список читачів із вказівкою номера абонементу, прізвищ і ініціалів, назви й автора книги, дати видачі.

**Варіант №17. Перелік задач, розв'язуваних у БД *Устаткування підприємств*:** вивести інформацію про замовлення за визначений період; видалити з БД замовників, що більш 2-х років нічого не замовляли; вивести звіт про устаткування; оновити ціну устаткування в зв'язку з подорожчанням.

**Варіант №18. Перелік задач, розв'язуваних у БД *Комп'ютерний центр комерційної інформації*:** підрахувати загальну вартість кожного замовлення; видалити з БД реквізити виконаних замовлень; збільшити ціну на продукцію одного з виробників на 10%; скласти список замовлень на визначений товар, кількість якого більше заданого числа; скласти список товарів із вказівкою кількості замовників, що оформили замовлення.

**Варіант №19. Перелік задач, розв'язуваних у БД *Посередницька фірма без складу*:** вести облік кількості відвантаженого замовникові; вести облік замовників, що недоплатили за продукцію; змінити ціни на товари в зв'язку зі зміною закупівельних цін; вести облік постачальників і пропонованих ними то-

варів; стежити за відновленням прайс-листа з роздрібними цінами; вести облік доходу фірми за поточний період.

**Варіант №20. Перелік задач, розв'язуваних у БД *Калькуляція*:** обчислити отриману підприємством прибуток за конкретний проміжок часу; скласти список виробів, нерентабельних для даного підприємства; видалити з БД реквізити нерентабельних виробів; підрахувати загальну суму угод по договорах, термін дії яких минає до визначеного часу; відібрати замовлення, що відносяться до конкретного договору і включають визначені вироби (указуються викладачем).

**Варіант №21. Перелік задач розв'язуваних у БД *Посередницька фірма зі складом*:** вести облік інформації про кількість і загальну суму товару, що зберігається на складі; вести облік товарів, яких необхідно поповнювати; вести облік товару, користується попитом; видалити з БД товар, що не закуповується; вивести інформацію про постачальників на конкретний товар і ціну для порівняльного аналізу; обновити ціну товару в зв'язку з подорожчанням; вести облік недоплати за товар.

**Варіант №22. Перелік задач, розв'язуваних у БД *Відділ кадрів*:** облік співробітників (дані про наймання і звільнення); скласти список молодих співробітників, стаж яких менше або дорівнює 3 рокам із вказівкою прізвища, імені та по батькові, відділу, посади; скласти список співробітників-ювілярів; провести індексацію окладів; видалити з БД реквізити двох співробітників, звільнених з роботи; створити звіт про лікарняні аркуші виданих за поточний місяць; вивести звіт про відпустки за поточний рік; скласти відомість по зарплаті.

**Варіант №23. Перелік задач, розв'язуваних у БД *Фірма (організація) по ремонту*:** визначити загальну вартість усіх матеріалів на складі; визначити одержуваний прибуток, вартість послуг, що робляться кожному замовникові, а також вартість матеріалів, що витрачаються на ремонт; скласти список співробітників, стаж роботи яких більше 10 років; змінити ціни на деякі матеріали; створити перехресну таблицю для висновку загальної суми зароблених грошей співробітником за виконану послугу.

**Варіант №24. Перелік задач, розв'язуваних у БД *Мале приватне торговельне підприємство*:** обчислити прибуток підприємства на визначену дату; збільшити ціну на всі реалізовані товари на 15%; видалити з БД реквізити двох товарів; скласти список товарів, ціна реалізації яких перевищила ціну постачання на 10%.

**Варіант №25. Перелік задач, розв'язуваних у БД *Банківські операції*:** скласти звіт по філіях; скласти звіт по особових рахунках клієнтів; підрахувати суму внесків за визначену дату; підрахувати суму видач за визначену дату; вивести кількість клієнтів, що відкрили особові рахунки за визначену дату по філіях.

**Варіант №26. Перелік задач, розв'язуваних у БД *Довідник підприємства*:** скласти список працівників підприємства з вказівкою прізвищ і ініціалів, посади, 9.кімнати, телефону; скласти список працівників підрозділу з вказівкою стажу роботи; скласти список підрозділів із вказівкою кількості підлеглих відділів; змінити початкову цифру телефонів підрозділів; змінити загальний стаж

роботи працівників з початком нового року; видалити з довідника підприємства дані про звільненого співробітника.

**Варіант №27. Перелік задач, розв'язуваних у БД *Абітурієнт*:** підрахувати загальну кількість балів за результатами іспитів кожного абітурієнта; сформувати екзаменаційну відомість; підрахувати кількість абітурієнтів, що надходять за контрактом; підрахувати вартість навчання по обраній спеціальності за 5 років; підрахувати кількість абітурієнтів по кожній спеціальності.

**Варіант №28. Перелік задач, розв'язуваних у БД *Відділ головного енергетика*:** видати інформацію про підрозділи, де перевитрата енергії; підрахувати загальну суму витрати енергії по підприємству; підрахувати суму витрати енергії по підрозділах; створити відомість споживання енергії; підрахувати суму витрати енергії по підрозділах за визначений проміжок часу.

**Варіант №29. Перелік задач, розв'язуваних у БД *Рекламне агентство*:** облік клієнтів; видалити з БД реквізити клієнтів, яких уже не обслуговують; обновити прайс-лист рекламного агентства; вести облік клієнтів, що не оплатили за рекламу; вести облік клієнтів, термін договору яких минув; видати службову інформацію про клієнтів за поточний період.

**Варіант №30. Перелік задач, розв'язуваних у БД *Фірма по охороні нерухомості*:** вести облік клієнтів і охоронюваних об'єктів; вести облік клієнтів, термін дії договору яких минув; вести облік оплати за охорону; видалити з БД клієнтів, які розірвали договір; обновляти прайс-лист із розцінками на установку й обслуговування.

### Контрольні запитання

1. Які поля приведених таблиць є первинними ключами?
2. Які дані зберігаються в стовпці 2 таблиці SUBJECT «Предмет навчання»?
3. Як по-іншому називається рядок, стовпець?
4. Напишіть запит для висновку ідентифікатора (номера) предмета навчання, його найменування, семестру, у якому він читається, і кількості годин, що відводяться на цей предмет, для всіх рядків таблиці SUBJECT.
5. Напишіть запит, що дозволяє вивести всі рядки таблиці EXAM\_MARKS, у яких предмет навчання має номер (SUBJ\_ID), рівний 12.
6. Напишіть запит на висновок в таблиці EXAM\_MARKS номерів предметів навчання, екзамени з яких здавалися між 10 і 20 січня 1999 року.
7. Напишіть запит, що вибирає дані про всі предмети навчання, екзамени з яких здані студентами, що мають ідентифікатори 12 і 32.
8. Напишіть запит на висновок назв предметів навчання, що починаються на букву «И».
9. Вивести прізвища, імена студентів і величину одержуваних ними стипендій, при цьому значення стипендій повинні бути збільшені в 100 разів.
10. Напишіть запит для підрахунку кількості студентів, що склали екзамен із предмета навчання з ідентифікатором, рівним 20

11. Напишіть запит, що дозволяє підрахувати в таблиці EXAM\_MARKS кількість різних предметів навчання.
12. Напишіть запит, що виконує вибірку для кожного студента значення його ідентифікатора і мінімальної з отриманих їм оцінок.
13. Напишіть запит, що здійснює вибірку для кожного студента значення його ідентифікатора і максимальної з отриманих ним оцінок.
14. Припустимо, що стипендія всім студентам збільшена на 20%. Напишіть запит до таблиці STUDENT, що виконує висновок номера студента, прізвище студента і величину збільшеної стипендії. Вихідні дані упорядкувати: а) за значенням останнього стовпця (величині стипендії); б) за абеткою прізвищ студентів
15. Напишіть запит, що по таблиці EXAM\_MARKS дозволяє знайти а) максимальні і б) мінімальні оцінки кожного студента і який виводить їхній разом з ідентифікатором студента.
16. Напишіть команду, що вводить у таблицю SUBJECT рядок для нового предмета навчання з наступними значеннями полів:
17. Нехай існує таблиця з ім'ям STUDENT1, визначення стовпців якої цілком збігаються з визначеннями стовпців таблиці STUDENT. Вставити в цю таблицю зведення про студентів, що успішно здала іспити більш ніж по п'ятих предметах навчання.
18. Створіть таблицю EXAM\_MARKS так, щоб не допускалося введення в таблицю двох записів про оцінки одного студента по конкретним іспитам і предмету навчання і щоб не допускалося проведення двох екзаменів з будь-яких предметів в один день.
19. Створити таблицю предметів навчання SUBJECT так, щоб кількість годин, що відводяться на предмет, за замовчуванням дорівнювало 36, не допускалися записи з відсутньою кількістю годин, поле SUBJ\_ID було первинним ключем таблиці і значення семестрів (поле SEMESTER) лежали в діапазоні від 1 до 12.

*Розглянуто основи SQL, правила конструювання складних умов добору, використовуючи булеві функції, що забезпечує високу точність у рішенні поставлених задач, а також створення таблиць і змінювання їх вмісту.*

# 10. ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ

*Вивчивши матеріали цього розділу, студент опанує основні прийоми програмування для офісних додатків операційної системи.*

Як мова програмування, VBA у всьому схожий на Visual Basic (про це говорить вже сама назва VBA – Visual Basic для додатків). Не слід плутати VBA з Visual Basic, оскільки це фізично різні речі – Visual Basic є самостійним пакетом для розробки програмного забезпечення, що не має прямого відношення до пакету MS Office. Програми на мові Visual Basic – це самостійні виконувані файли, такі ж, як MS Word або MS Excel, тоді як VBA-програми "живуть" тільки в робочому середовищі документів Office, оскільки VBA – це один з вбудованих інструментальних засобів додатків Office.

Проте, з погляду програміста – в синтаксичному, і в багатьох інших відношеннях – Visual Basic і VBA (Visual Basic для додатків), це одне і те ж. Більш того, у VB і VBA є ще третій "брат-близнюк": мова опису Web-сценаріїв VBScript, який також сумісний на рівні синтаксису з VBA і VB. У цьому полягає велика сила і величезна перевага технології Visual Basic, яка використовує одну універсальну мову для вирішення таких різних завдань.

І в цьому ж полягає головна причина, по якій будь-якій людині, що має справу з комп'ютерами, варто хоч трохи володіти синтаксисом Visual Basic – ніколи не знаєш, де він зможе тобі стати в нагоді. Мова програмування Visual Basic дозволяє займатися як офісним програмуванням на VBA, так і створенням Windows-додатків за допомогою пакету Visual Basic, а ще можна створювати Web-сценарії на мові VBScript. "Три в одному" – чи це не причина для того, щоб хоч трохи освоїти ази мови програмування Visual Basic (особливо, враховуючи той факт, що багато додатків сторонніх виробників у наш час забезпечуються вбудованими мовами, дуже схожими на Visual Basic)?

## 10.1. Дані VBA

Вивчення будь-якої мови програмування починають з даних, якими оперують програми на цій мові. Насправді, неможливо зрозуміти, як виконати ту або іншу операцію над яким-небудь об'єктом, не розуміючи, що цей об'єкт є.

У разі Visual Basic справа йде і простіше, і в той же час складніше. Ніяка "серйозна" мова програмування не дозволить використовувати змінні, які не були заздалегідь оголошені. У Visual Basic дозволено використовувати дані без явної або неявної вказівки їх типу. Можна взагалі не замислюватися про дані, придумуючи змінні "на ходу" і тут же використовуючи їх в своєму коді, як за-

манеться, оголошуючи змінну без вказівки її типу і використовуючи її для зберігання даних практично будь-якого типу.

Проте за все треба платити, і платою за таку свободу буде низька ефективність програмного коду, а також велика кількість помилок. Як показує практика, тільки невеликі і нескладні програми можна створювати подібним способом.

Але Visual Basic гнучкий і універсальний, і якщо ви хочете працювати серйозно, і створювати скільки-небудь складні програми, то у вас є можливість поводитися з даними так, як це роблять професійні програмісти.

### 10.1.1. Змінні і константи

Для зберігання даних використовують змінні і константи. Різниця між ними полягає в тому, що змінна може міняти своє значення в процесі виконання програми, а константа – це просто якесь значення, якому для зручності поводження з ним привласнено ім'я.

**Константи** застосовують у випадках, коли потрібно багато раз використовувати в програмі одне і те ж значення. Тоді для того, щоб змінити це значення, досить буде змінити тільки текст оголошення константи. Окрім цього, константи роблять текст програми "прозорішим" і легшим для розуміння. Наприклад, при виклику багатьох функцій використовуються зумовлені константи Visual Basic, наприклад:

```
If MsgBox("Створити лист?", vbYesNo, "Негативне сальдо") = vbYes ...
```

Тут константи vbYes і vbYesNo – це просто числа, запам'ятовувати які немає потреби, оскільки імена констант запам'ятати набагато легше. Константа vbYesNo означає, що у вікні повідомлення повинні бути присутніми кнопки **Yes (Aa)** і **No (fao)**, а константа vbYes означає, що користувач вибрав кнопку **Yes(Aa)**. Існує величезне число таких зумовлених констант, за допомогою яких задають параметри для функцій, властивостей і методів об'єктів, а також використовують в багатьох інших випадках.

Програміст може сам визначити константи для власних потреб. Наприклад, можна визначити константу для процентної ставки, використовуваної в обчисленнях:

```
Public Const Stavka1 As Single = 0.16667
```

Тут Stavka1 – це ім'я константи: далі в програмі можна використовувати це ім'я скрізь, де потрібне значення 0.16667. Зарезервоване слово Public означає, що константа буде "видна" у всіх модулях і у всіх процедурах. Такі дані повинні оголошуватися на рівні модуля, тобто поза *текстом* якої-небудь процедури. Тип числового значення тут визначений, як Single (про типи див. далі). Якщо замість Public використовувати слово Private, то константу "розумітимуть" тільки ті процедури, які знаходяться тільки в цьому ж модулі.

Далі можна оголосити ще одну константу, використовуючи при цьому вже оголошене ім'я Stavka1:



```
Public Const Stavka2 As Single = Stavka1 + 0.007
```

І знову треба буде використовувати ім'я Stavka2 скрізь, де потрібне значення  $0.16667 + 0.007$ . В результаті можна буде, змінивши всього один рядок коду, поміняти значення всіх ставок, які беруть участь в обчисленнях.

Константи, які використовуються локально, тобто тільки в тій процедурі, де вони оголошені, задаються без слів Public або Private, їх оголошення поміщається усередині тексту процедури. Наприклад, можна визначити константа-ми текст повідомлення про помилку і максимальне число помилок:

```
Const ErrorMessage1 As String = "Відбулася помилка  
1" Const MaxErrors As Integer = 100
```

**Змінні**, в протилежність константам, можуть міняти своє значення в процесі виконання програми. Але, як і до констант, звернення до змінних проводиться по їх іменах. Змінна може належати до одного з простих типів або ж до типу, визначеного користувачем, як комбінація простих типів. Окрім цього, змінна може бути оголошена, як масив.

Розглянемо наступні приклади оголошення змінних:

```
Dim WordRunning As Boolean
```

Логічна (булева) змінна WordRunning може приймати тільки значення True (Істина) або False (Брехня).

```
Dim WordCount, WordPointer As Integer
```

Цілочисельним змінним WordCount і WordPointer відтепер дозволено привласнювати як значення тільки цілі числа.

А ось добре знайомі приклади оголошення об'єктних змінних:

```
Dim MyParagraph As Paragraph
```

```
Dim MyDocument As Document
```

Якщо тип змінної явно не вказаний, за умовчанням буде прийнятий тип Variant (про типа Variant і взагалі про типи див. далі).

Автоматично, у момент запуску програми, змінні чисельних типів ініціалізувалися значенням 0, а змінні рядкових типів – значенням "Порожній рядок". Змінні типа Variant ініціалізувалися значенням Empty, а змінні об'єктних типів – значенням Nothing.

Оголошення змінних з областю видимості на рівні модуля необхідно поміщати в початок модуля, поза текстом якої-небудь процедури. Оголошення змінних усередині процедури слід поміщати в початок процедури (це підвищує "прозорість" тексту і полегшує його розуміння). Скрізь, де це можливо, бажано використовувати локальні змінні на рівні процедури – непомірне використання змінних, оголошених на рівні модуля, ускладнює текст і утрудняє його розуміння.

**Масивом** називають сукупність змінних одного типу, доступ до яких здійснюється по їх загальному імені (*імені масиву*) і номеру змінної в масиві (*індексу*). Застосування масивів спрощує операції з *масивами даних*. Наприклад:

```
Const MaxWords = 3000
```

```
Const WordsInDictionary = 200
```

```
Dim Words(0 To MaxWords) As String
```

```
Dim WordFrq(0 To MaxWords) As Integer
```

Якщо нижня межа індексу не вказана, то за умовчанням вона приймається рівною 0. За допомогою директиви Option Base значення нижньої межі за умовчанням можна змінити, наприклад Option Base 1 задасть відлік нижньої межі за умовчанням з 1.

Таким чином, масив Words, наприклад, можна було б визначити, як

```
Dim Words(3000) As String
```

Якби потрібно було використовувати дві розмірності в масиві, досить було б вказати два набори меж:

```
Dim Words(0 To 30, 0 To 100) As String
```

У Visual Basic підтримуються *динамічні масиви*, тобто масиви, розмірність яких міняється при виконанні програми. Наприклад, можна було б не вказувати розмірність масиву Words при його оголошенні:

```
Dim Words () As String
```

Але потім, перед першим зверненням до елементів такого масиву, необхідно буде задати його розмірність за допомогою оператора ReDim:

```
ReDim Words(0 To 3000)
```

При необхідності змінити розмірність, її знову можна перевизначити:

```
ReDim Words(0 To 4000)
```

Якщо нова розмірність більше старою, в масиві просто з'являться додаткові порожні елементи, в даному випадку рядки. Якщо нова розмірність менше старої розмірності, то значення частини елементів буде загублені.

За допомогою функцій LBound і UBound можна набути поточних значень нижньої і верхньої меж динамічного масиву відповідно. Оператори

```
MsgBox LBound(Words)
```

```
MsgBox UBound(Words)
```

виведуть на екран значення 0 і 4000.

*Об'єктну змінну* можна оголосити, як абстрактний об'єкт (тип Object), або вказати для неї певний об'єктний тип. Змінна типа Object (посилання на об'єкт) може вказувати на *об'єкт будь-якого типу*.

Змінна, оголошена, як посилання на об'єкт певного типу може вказувати на об'єкт тільки цього типу. Привласнення їй як значення об'єкту іншого типу викличе помилку.

Привласнення об'єктного значення (точніше, *посилання на об'єкт*) проводиться за допомогою зарезервованого слова Set:

```
Dim MyParagraph As Paragraph
```

```
Dim MyDocument As Document
```

```
Set MyDocument = Documents.Add
```

```
Set MyParagraph = MyDocument.Paragraphs.Add
```

З таким же успіхом можна оголосити обидві змінні, як об'єкт, і привласнити їм як значення об'єкти типу "документ" і "абзац" ...

```
Dim MyParagraph As Object
```

```
Dim MyDocument As Object
```

```
Set MyDocument = Documents.Add
```

```
Set MyParagraph = MyDocument.Paragraphs.Add
```

... або навпаки:

```
Set MyParagraph = Documents.Add  
Set MyDocument = MyDocument.Paragraphs.Add
```

У будь-якому випадку при цьому відбудеться так зване "пізніє скріплення" і фактичний тип об'єктів MyDocument і MyParagraph визначиться вже в процесі виконання програми. Такий спосіб роботи з об'єктами пов'язаний з певними складнощами, і використовувати його слід тільки в тих випадках, коли тип об'єкту, що повертається, дійсно наперед невідомий.

Визначити тип об'єкту, посилання на який отримане звідкись ззовні (наприклад, як значення якоїсь функції, що повертається), можна за допомогою оператора TypeOf і операції Is:

```
Dim MyObject As Object  
...  
Set MyObject = MyFunction()  
...  
If TypeOf MyObject Is Document Then  
    MyObject.Paragraphs.Add  
    ...  
End If  
If TypeOf MyObject Is Workbook Then  
    MyObject.WorkSheets.Add  
    ...  
End If
```

### 10.1.2. Типи даних

Всі дані, якими оперує VBA-код, відносяться до певного типу. Тип змінної можна не указувати при її оголошенні, але це не означає, що вона не відноситься ні до якого типу: просто неявним чином буде використаний універсальний тип за умовчанням, іменованій *Variant*.

Навіщо ж оголошувати тип змінної, якщо цього можна не робити? Існує, щонайменше, дві серйозні причини для явної вказівки типу даних, з якими передбачається мати справу.

Перша причина полягає в тому, що за свободу в обігу з типами доводиться платити ефективністю. Якщо потрібна змінна I для зберігання цілочисельних значень, то можна оголосити її з явною вказівкою цілочисельного типа Integer:

```
Dim I As Integer  
або ж просто як  
Dim I  
що те ж саме, що  
Dim I As Variant
```

Як вже указувалося, якщо тип не вказаний, то за умовчанням використовується тип Variant. Так можна робити, і програма нормально працюватиме. Якщо такі змінні небагато і звернення до них відбувається невелике число разів, то ніякої різниці програма "Не відчує". Але різниця все ж таки є. Змінна типа Variant займає у декілька разів більше місця в пам'яті комп'ютера, чим

змінна типа Integer, а звернення до неї відбувається у багато разів повільніше. Тому, якщо використовуються великі масиви і цикли, то змінні типа Variant здатні істотно уповільнити роботу програми.

Друга причина для явної вказівки типів всіх змінних полягає у все тих же друкарських помилках і помилках, які неминуче трапляються при роботі з початковим текстом. Якщо тип змінної I, яка згадувалася вищим, не вказаний явним чином, то можна привласнити їй будь-яке значення, наприклад рядкове:

```
I = "Рядкове значення"
```

У такій можливості криється невичерпне джерело помилок. Наприклад, вираз

```
I = "1" + "1"
```

жодним чином не еквівалентно виразу

```
I = 1 + 1
```

У першому випадку змінна I набуде рядкового значення "11", а в другому – числове значення 2. Якщо тип I не вказаний явним чином, то обидва варіанти однаково правомірні. Але якщо I – це змінна типа Integer, то спроба привласнити їй рядкове значення викличе помилку виконання і програміст виявить, що він щось наплутав. Інакше пошуки подібних помилок можуть стати нескінченними.

Всі **основні (прості)** типи даних, зазвичай використовувани в мовах програмування, підтримуються і в VBA. Розглянемо їх.

*Boolean* може приймати тільки значення True (Істина) або False (Брехня). Змінні типа Boolean використовуються головним чином при організації циклів і галужень – зазвичай вони грають роль прапорців, що сигналізують про стан програми. Наприклад, можна зберегти в Boolean-змінній результат логічної операції, скажімо, результат порівняння двох чисел:

```
Result As Boolean
Dim N1 As Integer
Dim N2 As Integer
...
N1 = 10
N2 = 5
...
Result = N1 > N2
If Result=True Then
    MsgBox "Перше число більше другого!"
End If
```

*Integer*. Цим змінним можна привласнювати тільки цілочисельні значення. Найменше значення: -32768 Найбільше значення: +32767.

Integer-змінні відрізняються малим об'ємом займаної пам'яті і високою швидкістю звернення. Часто їх використовують як лічильник циклу:

```
Dim Counter As Integer
...
For Counter = 1 to 1000
...

```

Next Counter

Змінним *Long* також можна привласнювати тільки цілочисельні значення. Від *Integer* цей тип відрізняється ширшим діапазоном значень.

Найменше значення: -2147483648

Найбільше значення: +2147483647

*Long*-змінні відрізняються високою швидкістю звернення, і найкращим чином підходять для завдань цілочисельної арифметики. Наприклад:

```
Dim MyDistance As Long
```

```
Dim MyTotalTime As Long
```

```
Dim MySpeed As Long
```

```
MySpeed = 2521
```

```
MyTotalTime = 811
```

```
MyDistance = MySpeed * MyTotalTime
```

Тип *Single* використовується для обчислень з плаваючою крапкою. *Single*-змінні забезпечують точність до 6-го знаку після десяткової крапки.

Найменше негативне значення: -3.402823E+38

Найбільше негативне значення: -1.401298E45

Найменше позитивне значення: 1.401298E-45

Найбільше позитивне значення: 3.402823E+38

Наприклад, за допомогою змінних типа *Single* можна виконувати обчислення з дробами:

```
Dim MyWeight As Single
```

```
Dim MyPrice As Single
```

```
Dim PriceToWeight As Single
```

```
MyWeight = 3.1
```

```
MyPrice = 121.77
```

```
PriceToWeight = MyPrice / MyWeight
```

Тип *Double* також використовується для обчислень з плаваючою крапкою. *Double*-змінні забезпечують точність до 14-го знаку після десяткової крапки.

Найменше негативне значення: -1.79769313486232E+308

Найбільше негативне значення: -4.94065645841247E-324

Найменше позитивне значення: 4.94065645841247E-324

Найбільше позитивне значення: 1.79769313486232E+308

Тип *Double* використовують, коли потрібно забезпечити високу точність обчислень з дробовими числами. Наприклад, от як може виглядати обчислення площі кола:

```
Dim MYPI As Double
```

```
Dim MyRadius As Double
```

```
Dim MySquare As Double
```

```
MYPI = 3.14159265358979
```

```
MyRadius = 9.79921235715865
```

```
MySquare = MYPI * MyRadius * 2
```

Значення грошового типа *Currency* використовують при фінансових розрахунках різного роду. Необхідно пам'ятати, що в грошовій арифметиці діють декілька інші закони, чим в арифметиці звичайної. Якщо йдеться про грошові су-

ми, то коректність арифметичних операцій буде забезпечена тільки при використанні спеціального грошового типу *Currency*.

Найменше значення: -922'337'203'685'477.5808

Найбільше значення: +922'337'203'685'477.5807

У значеннях типу *Currency* допускається використання не більше ніж чотирьох знаків після десяткової крапки. *Currency*-змінні відрізняються високою швидкістю обчислень, оскільки для внутрішнього представлення даних цього типу використовуються цілочисельні формати.

От як, наприклад, слід виконувати обчислення з грошовими сумами в перерахунку на валюту по курсу:

```
Dim MySum As Currency
Dim Kurs As Currency
Dim MyUSDSum As Currency
MySum = 123.45
Kurs = 6.77
MyUSDSum = MySum / Kurs
```

Дані типу *Date* служать для операцій із значеннями дати-часу.

Найменше значення: 1 січня 100 року

Найбільше значення: 31 грудня 9999 року

Значення дати і часу обмежуються символами #...# або "...", наприклад:

```
MyDate = "04/02/1962"
або
MyDate = #17/04/1972#
```

Значення дати і часу мають внутрішнє уявлення у вигляді числа з десятковим дробом, при цьому ціла частина числа відповідає даті, а дріб – часу. Це дозволяє використовувати операції складання і віднімання відносно дат і часу. Такі особливості, як різна кількість днів в різних місяцях, а також особливості високосного року, враховуються автоматично. Ось приклад обчислення із значеннями дати – на екран виводиться повідомлення про кількість днів, що пройшли з дня народження:

```
Dim BirthDay As Date
Dim TodayDay As Date
Dim DaysAge As Long
BirthDay = "17/04/1972"
TodayDay = #26/01/2001#
DaysAge = TodayDay - BirthDay
MsgBox DaysAge
```

Для виконання дій з текстовими рядками використовують змінні типу *String*. Максимальна довжина рядка: 2 мільярди символів. Рядкові значення необхідно брати в лапок:

```
Dim MyString As String
MyString = "Рядкове значення"
```

Спеціальним видом рядка є рядок фіксованої довжини. Її максимальна довжина дорівнює 65536 символів (конкретна довжина вказується при оголо-

шенні). Поводження з рядками фіксованої довжини відрізняється деякими особливостями. Наприклад:

```
Dim MyString1 As String
Dim MyString2 As String *30
MyString1 = "Рядкове значення1"
MyString2 = "Рядкове значення2"
```

В результаті змінна MyString1 містить рядок "Рядкове значення1", що складається рівно з 19 символів, а в змінною MyString2 міститься рядок завдовжки рівно 30 символів. При цьому перші 19 символів – це "Рядкове значення2", а наступні 11 позицій заповнено пропусками. На операції з рядками фіксованої довжини накладається ряд обмежень, тому їх використовують тільки в спеціальних випадках.

Змінна типа *Variant* здатна прийняти значення будь-якого з простих типів. Причому конкретний тип значення, яке приймає змінна, визначається у момент привласнення. *Variant* – тип за умовчанням для всіх змінних, які не були оголошені або ж були оголошені без вказівки типу.

*Variant*-змінна займає більше місця в пам'яті, чим змінна будь-якого простого типу, чії функції здатна виконувати *Variant*-змінна. Застосування цього типу слід по можливості уникати, оскільки звернення до даних типа *Variant* відбувається повільніше, місця в пам'яті вони займають більше, а програму з їх використанням важче відлагоджувати. Наприклад:

```
Dim MyFloat As Variant
Dim MyInteger As Variant
Dim MyText As Variant
MyInteger = 19
MyFloat = 2.001
MyText = MyInteger * MyFloat & " папуг"
```

В результаті в змінною MyText виявиться рядкове значення "38,019 папуг", причому *рядок* "38,019" вийшла неявним чином з числа 38,019, яке, у свою чергу, отримане в результаті операції множення цілого числа 19 і числа з плаваючою крапкою 2.001.

**Визначувані користувачем** типи даних називають також *записами* або *структурами*. Такий тип є якоюсь сукупністю типів (простих або визначених раніше), об'єднаних загальним ім'ям. Якщо потім оголосити змінну, що належить до знов певного типу, то доступ до її компонентів можна буде здійснювати по імені змінної і, відокремленому крапкою, імені компоненти, яким воно було задане при оголошенні типу. На відміну від масивів такий механізм дозволяє об'єднувати в єдину структуру змінні різних типів.

Припустимо, потрібно створити спеціальний призначений для користувача тип для процедури, яка оброблятиме особисті дані співробітників фірми. В цьому випадку необхідно оголосити тип (назвемо його, наприклад, *Sotrudnik*) на рівні модуля, тобто поза текстом процедури, яка його використовує:

```
Private Type Sotrudnik
    PersNum As Long
    Name As String
    BirthDay As Date
```

```
Salary As Currency
End Type
```

Після цього можна використовувати в процедурі тип `Sotrudnik` точно так, як і будь-який простий тип. Наприклад, можна оголосити дві змінні цього типу:

```
Dim MySotrudnik1 As Sotrudnik
Dim MySotrudnik2 As Sotrudnik
```

Для того, щоб привласнити значення компонентам змінної `MySotrudnik1`, необхідно через крапку вказувати ім'я відповідної компоненти:

```
MySotrudnik1.PersNum = 121
MySotrudnik1.Name = "Євгенія"
MySotrudnik1.BirthDay = "26/01/2000"
MySotrudnik1.Salary = 7900.11
```

Далі можна звертатися із змінній `MySotrudnik1`, як з єдиним цілим. Наприклад, привласнити її значення змінної такого ж типу:

```
MySotrudnik2 = MySotrudnik1
```

В результаті вираз `MySotrudnik2 . BirthDay` поверне значення дати "26/01/2000".

## 10.2. Синтаксичні конструкції мови VBA

У простих випадках, коли не відбувається спілкування з користувачем або операційною системою, не виконуються складні перетворення даних, програмування зводиться до побудови виразів, які беруть участь в операціях привласнення, і управління ходом виконання програми за допомогою спеціальних операторів і функцій, які передають управління різним фрагментам коду залежно від виконання тих або інших умов.

### 10.2.1. Вирази і операції

Якщо змінна – це те, що може знаходитися в лівій частині оператора привласнення, то вираз – це те, що може знаходитися в правій частині оператора привласнення.

Під виразом в Visual Basic розуміють іншу змінну, функцію, властивість деякого об'єкту або значення (числове, рядкове і т.д.), або результат дозволених над відповідними типами даних операцій. Звичайно, тип операндів і результату повинен бути сумісним з типом змінної.

#### 10.2.1.1. Арифметичні операції

**Арифметичні операції** виконуються над числовими значеннями. Їх результатом також є числове значення. Як операнди можуть виступати змінні, константи і вирази. У табл. 10. 1 приведений набір арифметичних операцій, допустимих в VBA-програмах:

Таблиця 10.1

Арифметичні операції в VBA

Операція	Опис	Операція	Опис
+	Додавання	-	Віднімання або зміна знаку



Операція	Опис
/	Ділення
*	Множення
^	Піднесення до ступеня
\	Цілочисельне ділення

Операція	Опис
Mod	Ділення по модулю (залишок від цілочисельного ділення)

Одномісні операції піднесення до ступеня і зміни знаку виконуються раніше всіх інших, операції \*, /, \, Mod виконуються раніше + і -. Операції, що мають однаковий пріоритет, виконуються зліва направо. Послідовність виконання операцій можна явно задати за допомогою дужок.

### 10.2.1.2. Логічні операції

**Логічні операції** виконуються над логічними (булевими) значеннями. Окрім цього, логічні операції можуть виконуватися над цілими числами – в цьому випадку мова йтиме про виконання логічної операції над відповідними двійковими розрядами операндів.

У табл. 10.2 приведений набір логічних операцій, допустимих в VBA-програмах:

Таблиця 10.2

Логічні операції в VBA

Операція	Опис
Not	Логічне заперечення
And	Логічне "І"
Or	Логічне "АБО"

Операція	Опис
Xor	Виняткове "АБО"
Imp	Імплікація
Eqv	Еквівалентність

### 10.2.1.3. Операції порівняння

**Операції порівняння** використовуються для порівняння даних. Дані можуть бути як числові і рядкові, так і об'єктні. Ці операції можна виконувати над змінними, константами і виразами. Результатом будь-якої операції порівняння завжди є логічне значення True або False.

Операції порівняння допустимі тільки відносно однорідних даних.

При порівнянні рядків використовується порядок символів в кодовій сторінці.

При порівнянні плаваючого значення з плаваючим подвійній точності останнє буде заокруглено.

У табл. 10.3 приведений набір операцій порівняння, допустимих в VBA-програмах.

Таблиця 10.3

Операції порівняння в VBA

Операція	Опис
<	Менше
<=	Менше або дорівнює
>	Більше
>=	Більше або дорівнює

Операція	Опис
=	Рівно
o	Не рівно
Is	Ідентичність об'єктів або відповідність типу

Операція	Опис
Like	Відповідність масці

Операція *Is* застосовна до об'єктних змінних (посиланням на об'єкти). Результат буде рівним `True` тільки у тому випадку, коли обидві змінні указують на один і той же об'єкт. Якщо другим операндом є ідентифікатор типу, то операція *Is* повертає значення `True` у тому випадку, коли об'єкт, на який посилається об'єктна змінна (перший операнд) належить до вказаного типу (другий операнд).

Наприклад, наступний макрос може визначити, чи існує в робочій книзі лист з ім'ям "Лист1":

```
Dim MyObject As Object
Set MyObject = Sheets("Лист1")
If MyObject Is Nothing Then
    MsgBox("У даній книзі немає такого листа")
End If
```

Результат операції *Like* – логічне значення, що показує, відповідає рядок заданій масці, чи ні. У табл. 10.4 приведений набір маскуючих символів, допустимих в VBA-програмах.

Таблиця 10.4

Маскуючі символи VBA

Маскуючий символ	Призначення	Маскуючий символ	Призначення
?	Довільний символ в даній позиції		позиції
*	Довільне число довільних символів, що починаються в даній позиції	[     ]	Один з символів, що містяться в списку
#	Довільна цифра в даній	[!...]	Будь-який символ, окрім символів, що містяться, в списку

За допомогою директиви `Option Compare Binary` можна задати режим бінарного порівняння операндів, тобто порівняння на основі збігу числових значень байтів. Режим текстового порівняння задається за допомогою директиви `Option Compare Text`. При цьому відмінність між заголовними і рядковими літерами не приймається до уваги.

#### 10.2.1.4. Операції з рядками

Об'єднання рядкових значень можна виконати за допомогою знаків "+" або "&". Обидва способи ідентичні по суті виконуваної дії, проте, відрізняються за деякими супутніми обставинами, які можуть супроводжувати їх використання.

Розглянемо наступний код:

```
Dim MyString As String
MyString = "Раз "
MyString = MyString + "два "
MyString = MyString & "три."
MsgBox MyString
```

Неважко переконатися, що в змінною MyString міститься значення "Разів, два, три." Зазвичай не рекомендують використовувати символ "+" з тієї причини, що цим же символом позначається операція арифметичного складання. Якщо використовувати дані типу Variant і не указувати явно тип змінних, якими передбачається оперувати, то звичка використовувати знак "+" для об'єднання рядків може зіграти з програмістом злий жарт. Наприклад, думаючи, що об'єднуються рядки "1" і "1", насправді вийде арифметична сума – число 2 замість рядка "11".

Проте символ "&", який зазвичай рекомендують використовувати замість символу "+" для об'єднання рядків, також не позбавлений деякої підступності. Річ у тому, що ради сумісності із старими варіантами мови BASIC, в Visual Basic підтримується "стародавній" спосіб скороченої вказівки типу змінних. Зокрема, символ "&" в кінці імені змінної відповідає типові Long. Якщо пропустити пропуск перед "&", то конструкція буде сприйнята компілятором, як оголошення цілочисельній змінній.

### 10.2.2. Конструкції мови, що управляють

Під конструкціями, що управляють, розуміють зазвичай операторів (інструкції) і вбудовані функції мови, за допомогою яких здійснюється управління ходом виконання програми. Сюди відносять цикли різного роду, умовні і безумовні переходи, а також конструкції галуження (умовні оператори), завдяки яким управління отримує той або інший фрагмент початкового коду.

#### 10.2.2.1. Перехід по мітці

Оператор **переходу по мітці** GoTo – самий, напевно, старовинний спосіб передачі управління в програмі. З настанням епохи структурного програмування цей оператор був зраджений анафемі, як головне джерело помилок і проблем при відладці програм, і використання його стало ознакою поганого смаку. Але все таки, зустрічаються іноді ситуації, коли застосування безумовного переходу GoTo може бути виправданим. Наприклад, при організації обробки помилок необхідно виконати безумовний перехід до фрагмента коду, оброблювального помилку:

```
On Error GoTo MyError1
...
Exit Sub
MyError1:
    MsgBox "Відбулася помилка 1"
End Sub
```

Інший варіант безумовного переходу представлений оператором Go Sub. Цей оператор також передає управління на задану мітку, проте, потім, коли у виконуваному коді зустрінеться оператор Return, управління повернеться до рядка коду, наступного за GoSub. Такий механізм використовують для виклику підпрограм. Наприклад:

```
GoSub MySubSub1
...
```

```

GoSub MySubSub2
...
Exit Sub
MySubSub1:
    MsgBox "Викликана підпрограма MySubSub1"
Return
MySubSub2:
    MsgBox "Викликана підпрограма MySubSub2"
Return
End Sub

```

### 10.2.2.2. Оператори циклу

**Циклічні конструкції** різного роду давно вже стали необхідним інструментом при виконанні масових операцій над даними. Існує декілька класичних циклів, кожний з яких характеризується певними особливостями. При рішенні конкретної задачі слід вибрати той варіант циклу, який найкращим чином для цього підходить.

Група операторів, включена між рядками **While** і **Wend** виконується циклічно до тих пір, поки вираз в рядку **While** повертає значення **True**. Якщо умова не виконується вже при вході в цикл, тіло циклу не буде виконано, а управління програмою буде передано операторові, наступному за оператором **Wend**. От як, наприклад, можна було б організувати "сканування" діапазону 20...100 у зворотному напрямі, від 100 до 20:

```

Dim Counter As Integer
...
Counter = 100
While Counter >= 20
...
Counter = Counter - 1
Wend

```

Цей цикл буде виконаний 81 раз.

Цикл *Do . Loop* це найбільш універсальний варіант циклу, відповідний для вирішення будь-яких циклічних завдань. Циклічна конструкція *Do..Loop* існує в двох варіантах, що відрізняються способом завдання умови виходу з циклу. Умова **While** (варіант **Do While . . .**) означає "виконувати, поки...". Умова **Until** (варіант **Do Until. . .**) означає "виконувати, поки не...".

Група операторів, включена між рядками **Do** і **Loop** виконується циклічно, поки умову **While** зберігає значення **True**, або поки умову **Until** не прийме значення **True**. Наприклад:

```

Do While ...
    If ... Then
        Exit Do
    End If
...
Loop

```

У будь-якому місці циклу можна використовувати оператора Exit Do для дострокового виходу їх циклу при виконанні якоїсь додаткової умови.

Перевірку виконання умови можна помістити в початок або в кінець циклу. В цьому випадку допустимі наступні варіанти Do While [умова] ... Loop, Do Until [умова] ... Loop, Do ... Loop While [умова] і Do ... Loop Until [умова] .

Наприклад:

```
Do
...
    If ... Then
        Exit Do
    End If
...
Loop Until ...
```

Кількість проходів циклу *For . Next* визначається початковим і кінцевим значеннями лічильника, а також значенням кроку. Якщо крок циклу не вказаний, то він буде прийнятий рівним 1. Наступний цикл буде виконаний 100 разів, причому лічильник циклу (змінна I) на кожному проході збільшуватиме своє значення на 1:

```
For I=1 To 100
...
Next I
```

Наступний цикл буде виконаний 10 разів, причому лічильник циклу (змінна I) на кожному проході збільшуватиме своє значення на 10:

```
For I=10 To 100 Step 10
...
Next I
```

Відлік можна проводити і у зворотному напрямі, для цього значення кроку повинне бути негативним:

```
For I=100 To 10 Step -10
...
Next I
```

За допомогою конструкції *For. . . Each* можна організувати перебір всіх об'єктів, що належать до деякого сімейства. На кожному проході змінна циклу міститиме в собі черговий об'єкт сімейства. Доступ до властивостей і методів об'єкту при цьому стає можливим через ім'я змінної циклу. Якщо оголошена об'єктна змінна MyWord:

```
Dim MyWord As Object
```

то перебір всіх слів в поточному документі можна організувати таким чином:

```
For Each MyWord In ActiveDocument.Words
...
Next MyWord
```

### 10.2.2.3. Прийняття рішень: оператори галуження (умовні конструкції)

Для ухвалення рішень в процесі виконання програми використовують різноманітні умовні конструкції. Подібно до циклів, багато хто з них взаємозамінний, – по суті, будь-які види ухвалення рішень (так само, як і будь-які циклічні конструкції) можна реалізувати за допомогою простого оператора If ... Then ... Else ... End If. Але часто вибір відповідної конструкції значно спрощує роботу програміста. Такі конструкції називають іноді *операторами галуження*, оскільки вони позначають шляхи виконання програми, що розгалужуються.

Функція-перемикач Switch служить для привласнення одного із значень при виконанні однієї з умов. Функція Switch повертає перше значення, якщо істинною виявиться перша умова, друге значення, якщо істинною виявиться друга умова і т.д. Перевірка умов виконується зліва направо. Наприклад:

```
Dim MyNum As Integer
Dim MyText As String
MyNum = ...
MyText = Switch(MyNum <= 11, "Замало буде!", MyNum >
11, "О.К.", ...)
MsgBox MyText
```

Якщо жодне з умов в списку не виконалося, функція повертає значення Null. Незалежно від того, яке по рахунку умова виконалася, функція Switch перевіряє *всі* умови в списку, і в цьому полягає її істотна відмінність від інших умовних конструкцій.

Ф

У  
Н  
К  
Ц  
І  
Я

С  
h  
o  
o  
s  
e

У

В  
С  
Ь  
О  
М

У  
с  
х  
о  
ж  
а  
  
н  
а  
  
S  
w  
i  
t  
c  
h  
  
—  
  
В  
і  
Д  
М  
і  
Н  
Н  
і  
С  
Т  
Ь  
  
П  
О  
Л  
Я  
Г  
а  
Є  
  
В  
  
Т  
О  
М

У  
,

Щ  
О

В  
И  
Р  
А  
З  
-  
П  
Е  
Р  
Е  
М  
И  
К  
А  
Ч

У  
К  
А  
З  
У  
Є  
Т  
Ь  
С  
Я

О  
Д  
И  
Н

Р  
А  
З

Н  
А



П  
О  
Ч  
А  
Т  
К  
У

С  
П  
И  
С  
К  
У

В  
А  
Р  
І  
А  
Н  
Т  
І  
В  
.

Ф  
У  
Н  
К  
Ц  
І  
Я

П  
О  
В  
Е  
Р  
Т  
А  
Є

Т  
Е

З  
Н  
а  
ч  
е  
н  
н  
я

і  
з

с  
п  
и  
с  
к  
у  
,

п  
о  
р  
я  
д  
к  
о  
в  
и  
й

н  
о  
м  
е  
р

я  
к  
о  
г  
о

в

і  
Д  
П  
О  
В  
і  
Д  
а  
є

з  
н  
а  
ч  
е  
н  
н  
ю

в  
и  
р  
а  
з  
у  
-  
п  
е  
р  
е  
м  
и  
к  
а  
ч  
а  
.

Н  
а  
п  
р  
и  
к  
л

а  
д  
:

```
Dim KodKlienta As Integer
Dim Klient As String
KodKlienta = 2
Klient = Choose(KodKlienta, "Іванов", "Петров",
"Сидоров")
```

У рядковій змінній Klient в результаті міститиметься значення "Петров". Якщо значення виразу-перемикача не відповідає числу елементів в списку, функція поверне значення Null.

Функція *IIf* повертає одне з двох значень залежно від логічного значення, яке прийме вираз-умову. Наприклад:

```
Dim MyNum As Integer
Dim MyText As String
MyNum = ...
MyText = IIf(MyNum <= 11, "Замало буде!",
"О.К.")
MsgBox MyText
```

Конструкція *If... Then ... Else ... End If* забезпечує виконання одного або декількох операторів залежно від істинності або помилковості заданої умови. Роль умови повинен грати вираз, що повертає логічне (булевий) значення True або False. Існує два варіанти синтаксису умовного оператора If. От як виглядає варіант запису в один рядок що не вимагає завершення цієї конструкції словом End If:

```
If ... Then . . . Else . . .
```

Якщо в "гілках" Then або Else повинні виконуватися декілька операторів, то конструкція розгортається в декілька рядків:

```
If ... Then
...
Else
...
End If
```

Якщо умова в рядку If виконується або ж там міститься логічне значення True (наприклад, змінна типа Boolean), то виконуватиметься гілка Then, інакше виконується гілка Else (яка може бути відсутнім в обох варіантах синтаксису). Наприклад:

```
Dim MyNum As Integer
MyNum = ...
If MyNum <= 11 Then
    MsgBox "Замало буде!"
Else
    MsgBox "О.К."
End If
```

Конструкція Select Case "ухвалює рішення" на основі значення виразу-перемикача і передає управління однієї з груп операторів.

Список значень для кожного з варіантів можна задати декількома способами.

- За допомогою символів операцій порівняння ">", "<", "<>", "<=" і т.д.
- У вигляді інтервалу "Начальноезначеніє" То "Конечноезначеніє".
- Простим перерахуванням: "Значение1", "Значение2", "Значение3".

В результаті буде виконана група операторів, наступна за тим Case-рядком, в чиєму списку значень опиниться результат виразу, заданого в рядку Select Case.

Після цього управління отримає рядок, наступний після рядка End Select.

Якщо ні в один з Case-рядків не "спрацює", виникне помилка виконання. Щоб цього уникнути, слід використовувати рядок Case Else, де указуються оператори, що підлягають виконанню в решті всіх випадків. Наприклад:

```
Dim MyNum As Integer
Dim MyText As String
MyNum = ...
Select Case MyNum
Case < 11
    MyText = "Замало буде!"
Case 18
    MyText = "Рівно вісімнадцять"
Case 19 To 25
    MyText = "Від 19 до 25"
Case 30, 40, 50, 60
    MyText = "Круглі числа!"
Case >100
    MyText = "Більше ста!"
Case Else
    MyText = "Не знаю, що тут сказати..."
End Select
MsgBox MyText
```

### 10.3. Оператори і вбудовані функції мови VBA

Дуже часто різниця між оператором (інструкцією) мови і вбудованою функцією малопомітна. Насправді, наприклад, оператора MsgBox і функція MsgBox вирішують одне і те ж завдання – виведення повідомлення на екран і часто немає необхідності замислюватися, про що саме йдеться: про функцію або про оператора. Просто одне і те ж зарезервоване (ключове) слово MsgBox використовується двома способами: його можна записати, як оператор з параметрами або ж, як функцію, що повертає значення, з параметрами, увязаними в дужки.

Функція, як відомо, повертає якесь значення, а оператор мови по використанню подібний до процедури-підпрограми, досить вказати зарезервоване ключове слово `і`, при необхідності, додати параметри. У попередніх підрозділах не було зроблено строгої відмінності між операторами і функціями: і ті, та інші були розділені по тій ролі, яку вони грають в програмі. Тепер займемося вивченням тих операторів і функцій, які, власне кажучи, щось *роблять*, тобто забезпечують виконання програмою своїх функцій, того, ради чого взагалі створюються програми.

### 10.3.1. Інтерфейс користувача

Прості завдання призначеного для користувача інтерфейсу полягають у виведенні повідомлень користувачеві і в отриманні реакції користувача в якому-небудь певному вигляді – будь то рядок, введений з клавіатури або ж вибір однієї з кнопок у вікні повідомлення.

Функція `MsgBox` може використовуватися як функція, що повертає значення (при цьому вона повертає ціле число, що сигналізує про те, яку з кнопок у вікні повідомлення вибрав користувач), або ж, як звичайний оператор, або процедура з параметром – текстом повідомлення.

Функція `InputBox` служить для введення рядкових значень. При цьому у вікні міститься текст запрошення, подібний до тексту повідомлення функції `MsgBox`, а в полі введення може міститися "рядок за умовчанням" – рядок, який користувач може видалити (першою буде натиснута клавіша, відповідна одному з друкованих символів) або відредагувати (першою буде натиснута одна з клавіш редагування).

### 10.3.2. Взаємодія з операційною системою

Користувач – не єдиний об'єкт зовнішнього світу, з яким доводиться взаємодіяти програмі. Іноді макрос може потребувати обміну даними з операційною системою. Найпоширеніша взаємодія такого роду – це файлові операції, коли програма читає дані з дискового файлу або записує дані у файл. Можливі також випадки, коли програмі необхідно управляти роботою іншого додатку, причому це може здійснюватися не тільки за допомогою технології `Automation` (врешті-решт, далеко не всі додатки підтримують `Automation`), але і іншими способами.

Функція `Shell` здійснює асинхронний запуск іншого додатку. Асинхронність означає, що функція не чекає ніякої реакції від запущеної програми і виконання VBA-коду продовжується незалежно від результату виконання цього додатку. Функція повертає цілочисельне (`Double`) значення у форматі `Variant`, відповідне ідентифікатору додатку, що стартує (коду завдання); якщо програму запустити не вдалося, функція `Shell` повертає значення 0.

От як, наприклад, VBA-код може запустити редактор системних файлів `Windows`:

```
Dim MyAppNum As Variant
    On Error Resume
Next
```

```

MyAppNum = Shell("c:\windows\system\sysedit.exe", 1)
If MyAppNum=0 Then
    MsgBox "Запуск програми неможливий"
End If

```

Оператор AppActivate робить поточним вікно заданого додатку. Щоб вказати необхідне додатку, необхідно використовувати його ідентифікатор (код завдання), повернений функцією Shell. Замість ідентифікатора можна вказати текст титульного рядка додатку, який необхідно зробити поточним.

От як, наприклад, можна активізувати додаток з використанням ідентифікатора (коду завдання), поверненого функцією Shell:

```

Dim MyAppNum As Variant
On Error Resume
Next
MyAppNum = Shell("c:\windows\system\sysedit.exe", 1)
AppActivate MyAppNum

```

Приклад, активізації додатка з використанням тексту з рядка заголовка вікна додатка:

```

AppActivate "Редактор файлів настройки", True

```

За допомогою другого параметра (логічне значення True) визначається, що активація додатку здійснюється тільки після того, як додаток, що викликає, стане активним. За умовчанням цей параметр приймається рівним False і активація необхідного додатку відбувається негайно.

Якщо не розглядати виконувани і системні файли, фізичний доступ до яких – прерогатива операційної системи, то всі файли даних насправді нічим не відрізняються один від одного – відрізняється лише режим, в якому програма відкриває конкретний файл і те, яким чином організовані дані у файлі. Далі згадуються, наприклад, "файли послідовного доступу" і "файли довільного доступу", а також "двійкові файли" – з цього не виходить, що це якісь спеціальні різновиди файлів. Будь-який файл можна відкрити в будь-якому з режимів – все залежить від того, що програма збирається з цим файлом робити і під яким як передбачається розглядати що містяться у файлі дані.

VBA-програми часто мають справу з файлами неявним чином, взаємодіючи з Word, Excel, Access або PowerPoint, вони одночасно звертаються до файлів даних, що належать до цих додатків. Всі подібні операції виконуються автоматично. Тут мова піде про файлові операції, які можуть потрібно VBA-програмі для якихось внутрішніх спеціальних цілей.

Visual Basic для додатків володіє обширним інструментарієм для файлових операцій. Оператори і функції мови дозволяють працювати з дисковими пристроями (приводами), каталогами і власне файлами. Відносно файлів визначений ряд операцій (відкриття і закриття, читання і запис, позиціонування і ін.), проте якщо йдеться про конкретний файл, багато що залежить від його формату і режиму доступу, з яким відкритий файл.

Visual Basic (і не тільки Visual Basic) розрізняє три види файлів (або, що практично те ж саме, три режими доступу до файлу):

**Файли послідовного доступу** - Всі текстові (ASCII-, ANSI-файли) складаються з послідовності рядків символів, при цьому кожен рядок може мати довільну довжину і доступ до цих рядків може бути тільки послідовним. Якщо, наприклад, програмі потрібний 101-й рядок текстового файлу, їй доведеться прочитати спочатку перші 100 рядків, причому абсолютно невідомо, в що це виллється і чи є у файлі 101-строка, мабуть, він складається з одного величезного рядка.

**Файли довільного доступу** - Файл довільного доступу складається із записів фіксованої довжини, тому місцеположення запису у файлі можна обчислити множенням номера запису на її довжину. Завдяки цьому виявляється можливим позиціонуватися на довільний запис. Якщо програмі потрібний 101-й запис у файлі даних, вона може прочитати безпосередньо 101-й запис. Розмір файлу кратний розміру запису, і з цієї причини завжди можна визначити, скільки записів міститься у файлі за його розміром, навіть не відкриваючи його.

**Бінарні (двійкові) файли** - Бінарний файл є окремий випадок файлу довільного доступу – досить прийняти довжину запису рівною одному байту і стає можливим позиціонуватися на довільний байт у файлі. Що означають всі ці байти і як вони організовані – справа самої програми.

Для операцій з файлами, каталогами і дисками в VBA існує набір операторів, функціонально відповідний командам операційної системи MS DOS.

**Оператор ChDrive** робить дисковий пристрій поточним (вибирає його). Як параметр повинен виступати текстовий рядок, першим символом якого повинні бути літера – ім'я диска. Наприклад, оператор

```
ChDrive "D"
```

зробить поточним диск D. Звернете увагу на той факт, що замість рядка "D" можна було б вказати, наприклад, "D: \TEMP" з таким же успіхом – оператор ChDrive розпізнає тільки першу літеру рядка.

За допомогою **оператора MkDir** можна створити каталог (підкаталог) із заданим ім'ям. Наприклад, створити в кореневому каталозі диска C підкаталог з ім'ям WORK можна за допомогою оператора:

```
MkDir "C:\WORK"
```

Щоб змінити поточний каталог на іншій, тобто зробити поточним інший каталог, потрібно скористатися **оператором ChDir**. Дисковий пристрій, на якому розташовується каталог, повинен бути поточним. Якщо це не так, то спочатку слід використовувати оператора ChDrive.

От як, наприклад, можна використовувати рядок з повним ім'ям каталога (тобто, з ім'ям, що включає ім'я диска) як параметр для операторів ChDir і ChDrive одночасно:

```
Dim MyPath As String  
MyPath = "C:\WORK"  
ChDrive MyPath  
ChDir MyPath
```

При цьому оператор ChDrive "зрозуміє" тільки літеру "C" і зробить поточним диск C, а оператор ChDir "зрозуміє" решту всіх літер окрім підрядка "C:" і зробить поточним каталог WORK.



Видалити каталог можна за допомогою *оператора Rmdir*, якому як параметр необхідно вказати повне ім'я каталога, підмета видаленню. Ім'я повинне включати ім'я диска, а "приречений" каталог повинен бути порожній. Спроба видалити непорожній каталог приведе до помилки виконання.

```
Rmdir "C:\WORK"
```

*Функція CurDir* дозволяє дізнатися ім'я поточного каталога. Як параметр можна вказати ім'я диска, за відсутності параметра буде на увазі поточний диск. Функція повертає рядок повного імені поточного каталога.

Наприклад, якщо програмі потрібно змінити каталог, а потім повернутися до початкового каталога, то вона повинна "запам'ятати" ім'я поточного каталога. Ось, як можна використовувати при цьому функцію CurDir:

```
Dim MyPath As String
Dim TargetPath As String
MyPath = CurDir()
MsgBox "Шлях до каталога "MyPath" буде збережений"
TargetPath = "C:\WORK"
ChDrive TargetPath
ChDir TargetPath
MsgBox "Повертаємося до каталога " MyPath
ChDrive MyPath
ChDir MyPath
```

Збережене в рядковій змінній MyPath ім'я поточного каталога використовується потім для того, щоб повернутися до початкового каталога.

Для того, щоб прочитати імена файлів, що містяться в каталозі, використовують *функцію Dir*, яка по своїй дії аналогічна однойменній команді ОС MS DOS.

Як параметри цієї функції необхідно вказати ім'я каталога і (необов'язковий параметр) число, задає атрибути файлів (це число виходить в результаті підсумовування зумовлених констант Visual Basic). В ім'я каталога можна додати маску імен файлів, що формується за тими ж правилами, що і маска для команд MS DOS: символ "?" заміщає будь-який символ, а символ "\*" заміщає будь-яке число будь-яких символів.

В результаті функція поверне *перше* ім'я файлу, що задовольняє заданим умовам, якщо, звичайно, хоч би один такий файл існує. Для того, щоб отримати решту всіх імен, слід продовжувати виклики функції Dir вже *без параметрів* до тих пір, поки вона не поверне порожню рядок.

Наприклад, програмі потрібно отримати імена всіх файлів з розширенням TXT в каталозі WORK диска C:.

```
Оголосимо рядкову змінну:
Dim MyFileName As String
Перший виклик функції Dir з вказівкою каталога:
MyFileName = Dir("C:\WORK\*.TXT")
MsgBox MyFileName
```

Далі викликаємо функцію Dir *без параметрів* до тих пір, поки вона не поверне порожню рядок:

```

Do While MyFileName <> " "
    MyFileName = Dir
    MsgBox MyFileName
Loop

```

Для того, щоб **перейменувати** файл або каталог, використовують оператора Name..As... Як параметри при цьому вказують ім'я файлу або каталога, тобто старе його ім'я, яке ще залишається дійсним, і нове ім'я. Оператор Name, подібно до відповідної команди MS DOS, дозволяє використовувати маскуючі (що заміщають) символи в іменах для групових операцій з файлами. Зрозуміло, що часто зміни в імені файлу автоматично спричиняють за собою його переміщення в інший каталог.

```

Name "C:\WORK\MyText.txt" As "C:\ARCHIVE\MyTextAr.txt"
Name "C:\WORK" As "C:\TMP\OLDFILES"

```

**Копіювання** файлів здійснюється за допомогою оператора FileCopy. Його параметри – це ім'я файлу, підмета копіюванню, і ім'я нового файлу-копії. Використання маскуючих символів "?" і "\*" тут неприпустимо. Наприклад:

```

FileCopy "C:\WORK\MyText.txt", "C:\WORK\MyBackUp.txt"

```

Для **видалення** файлів використовують оператора Kill. Цей оператор дозволяє використовувати маскуючі символи, але, на відміну від команди Del, видаляє групу файлів без попередження. Наприклад, наступний оператор видалить всі текстові файли в каталозі C:\WORK:

```

kill "C:\WORK\*.txt"

```

Перш ніж **дістати доступ** даним, що містяться у файлі в будь-якому режимі, файл необхідно спочатку *відкрити*. Це завдання вирішується за допомогою оператора Open. Синтаксис цього оператора виглядає таким чином:

```

Open Імяфайла For Режимдоступа As Номерфайла

```

**Читання** даних з відкритого файлу послідовного доступу здійснюється за допомогою оператора Input або, якщо йдеться про текстовий файл, Line Input . Необхідно при цьому вказати номер, з яким відкритий файл, а також змінну, в яку повинні бути поміщені введені з файлу дані. От як, наприклад, можна прочитати перший рядок з файлу C:\WORK\MyText.txt в рядкову змінну MyTempString:

```

Dim MyFileNum As Integer
Dim MyTempString As String
...
MyFileNum = FreeFile()
Open "C:\WORK\MyText.txt" For Input As MyFileNum
...
Line Input MyFileNum, MyTempString

```

Наступний рядок тексту буде прочитаний з файлу в змінну при наступному виконанні оператора Line Input. Звичайне читання всього файлу забезпечують за допомогою циклічної конструкції Do Until, вказуючи як умову ознаку кінця файлу EOF. Наприклад, наступний код прочитає в змінну MyTempString послідовно всі рядки з текстового файлу, причому на кожному проході циклу в змінній міститиметься тільки *одна*, черговий рядок тексту:

```
Do Until EOF(MyFileNum)
    Line Input MyFileNum, MyTempString
Loop
```

Подібно до того, як введення даних в послідовний файл може здійснюватися за допомогою двох операторів – Input і Line Input, для **виводу** у файл послідовного доступу використовують операторів Write і Print. При цьому Write складає "пару" для Input (послідовне виведення довільних даних), а для роботи з текстовими файлами використовують Line Input і Print.

Файл необхідно заздалегідь відкрити для виводу з використанням ключового слова Output:

```
MyFileNum = FreeFile()
Open "C:\WORK\MyFile.txt" For Output As MyFileNum
```

У разі використання оператора Write досить вказати номер файлу і змінну, в якій містяться дані для висновку:

```
Write MyFileNum, MyVar
```

Причому змінна MyVar може відноситися до будь-якого простого типу. Якщо це змінна типа Boolean, то у файл буде виведено #True# або #False#, якщо це змінна типа Date, то виведеться текстове значення дати в короткому форматі, і так далі. У разі рядкових значень краще використовувати оператора Print:

```
Print MyFileNum, "Змінна MyVar ="; MyVar
```

Можна поміщати в рядок висновку декілька значень (змінних або рядкових констант), розділених знайомий ";".

Після того, як всі необхідні дії з відкритим файлом завершені, його необхідно **закрити**. Щоб закрити певний файл, використовують оператора Close з вказівкою номера файлу:

```
Close MyFileNum
```

За допомогою оператора Reset (без параметрів) закриваються *всі* відкриті файли:

```
Reset
```

При виконанні **дій з файлом довільного доступу** зазвичай використовують поняття *поточного запису*, тобто запису, до якого проводилося звернення востаннє, або *першому запису* файлу, якщо звернень до записів ще не було. Якщо при подальшому зверненні номер необхідного запису не буде вказаний, то за умовчанням поточним стане наступний запис, тобто покажчик поточного запису за умовчанням переміщається послідовно до останнього запису у файлі.

Після того, як файл відкритий в режимі довільного доступу, можна помістити позицію введення-висновку на будь-який запис файлу – після цього операція запису або читання виконуватиметься саме для вказаного запису. Щоб зробити це, використовують оператора Seek:

```
Open "C:\WORK\Sotrud.dat" For Random Access
Read Write As MyFileNum
Len=Len(MySotrudnik)
Seek MyFileNum, 10
```

Два параметри, наступні після Seek, – це номер, привласнений файлу при його відкритті і порядковий номер запису, на який необхідно позиціонуватися.

В результаті позиція введення-висновку опиниться на 10-му записі файлу. Якщо значення номера запису перевищує кількість записів у файлі, то позиція введення-висновку буде поміщена в кінець файлу.

Щоб дізнатися, який запис є в даний момент поточним, використовують функцію `Loc`. Після виконання приведених вище операцій вираз `Loc (MyFileNum)` поверне значення 10.

Запис у файли довільного доступу проводиться за допомогою оператора `Put`. При цьому необхідно вказати номер файлу, номер запису, в який проводиться запис (цей параметр необов'язковий, він просто неявно використовує функцію `Loc`) і ім'я змінної, в якій містяться ті, що підлягають запису дані.

```
Open "C:\WORK\Sotrud.dat" For Random Access
Read Write As MyFileNum
Len=Len(MySotrudnik)
Put MyFileNum, 10, MySotrudnik
```

В результаті вміст змінної `MySotrudnik` буде записаний в 10-й запис файлу `C:\WORK\Sotrud.dat`.

Протилежна дія, тобто читання з файлу довільного доступу, здійснюється оператором `Get`. Тут теж необхідно вказати номер файлу, номер запису, в який проводиться запис (цей параметр, так само, як і у разі `Put`, необов'язковий) і ім'я змінної, в яку слід прочитати дані з файлу:

```
Get MyFileNum, 10, MySotrudnik
```

Після виконання операторів `Put` або `Get` покажчик поточного запису переміщається вперед, на наступний запис. Якщо використовувати `Put` або `Get` без вказівки номера запису, то поточним записом, тобто записом, з яким повинна проводитися операція запису або читання, кожного разу ставатиме наступний запис. І так продовжуватиметься до тих пір, поки функція `EOF` з параметром – номером даного файлу, не поверне значення `True`, що означатиме кінець файлу.

Якщо програма працює в розрахованому на багато користувачів середовищі (наприклад, звертається до мережевого файлу), то при операціях з файлами довільного доступу слід блокувати запис, до якого проводиться звернення. Це необхідно зробити з тієї причини, що два користувачі можуть одночасно змінити дані в одному і тому ж записі, і в цьому випадку стан файлу виявиться невизначеним.

Для блокування записів використовують оператора `Lock`:

```
Open "C:\WORK\Sotrud.dat" For Random Access Read
Write As MyFileNum
Len=Len(MySotrudnik)
...
Lock MyFileNum, 10 To 11
Put MyFileNum, 10, MySotrudnik1
Put MyFileNum, MySotrudnik2
Unlock MyFileNum, 10 To 11
```

Оператор `Lock` блокує, тобто переводить в режим виняткового (монопольного) доступу запису з 10-ою по 11-у. Блоковані записи стають недоступними

для модифікації іншим додаткам. Потім в 10-у і в 11-у запису виводяться дані з двох змінних типу

```
Sotrudnik.
```

Оператор Unlock призначений для розблокування файлу, який був заблокований оператором Lock. У попередньому прикладі останній рядок коду знімає блокування з 10 і 11 записів файлу MyFileNum.

### 10.3.3. Робота із значеннями дати і часу

Операції із значеннями дати і часу володіють певною специфікою. У форматах представлення дати і часу дуже багато що залежить від версії Windows, версії Office і локалізації того і іншого. Зрозуміло, що формати представлення таких значень відрізняються в локалізованих росіянах, паневропейських і американських версіях. Крім того, на представлення значень дати і часу впливають призначені для користувача настройки конкретного комп'ютера, тобто параметри, задані за допомогою діалогового вікна Властивості регіональних настройок (Regional Settings Properties) панелі управління Windows. Для його відображення потрібно виконати команду Настройка/ Панель управління (Settings Control Panel) меню Пуск (Start).

Значення дати і часу представляються в додатках Office у формі так званого коду "дата-час" (званому також "датою\_в\_числовом\_формате"). Код "дата-час" є числом, ціла частина якого кодує дату (день), а дріб – час. Таке уявлення дозволяє дуже просто виконувати будь-які арифметичні операції з будь-якими тимчасовими проміжками.

Проміжку часу, рівному 24 годинам відповідає число 1 (воно ж відповідає одному дню в значеннях дати).

Полудню (12:00) відповідає значення 1/2, шести годинникам вечора (18:00) – 3/4 і т.д. Звідси легко отримати дробові числа, відповідні елементарним проміжкам часу: 1 година = 1/24, 1 хвилина = 1/1440 і 1 секунда = 1/8640.

Значення дати є цілим числом, що позначає кількість днів, відлічених від так званої *першої дати*. У різних системах дат використовуються різні перші дати, тобто дати, якому відповідає код "дата-час", рівний 1. За умовчанням додатки Office використовують "Систему дат 1900", окрім цього, Office може працювати в "Системі дат 1904", іменованою також "Системою дат 21 століття".

У середовищі VBA існує невелика відмінність для "Системи дат 1900" – як перша дата VBA використовує 31 грудня 1899 року, тоді як компоненти MS Office – 1 січня 1900 року. Для дат пізніше за 29 лютий 1900 року вищеописана відмінність не існує.

Перетворити дату в числовий формат, тобто в код "дата-час", а точніше, значення типу Date, можна за допомогою функції DateSerial. Як параметри функції необхідно вказати три числа: рік, місяць і день місяця. Число місяців може перевищувати 12, а число днів може перевищувати 31 – всі необхідні поправки будуть внесені автоматично. Наприклад:

```
Dim MyDate As Date  
MyDate = DateSerial(5, 10, 11)
```

В результаті в змінній MyDate міститиметься дата "11.10.2005". Наступний рядок коду

```
MyDate = DateSerial(1996, 15, 6)
```

привласнює змінною MyDate дату "06.03.1997". Наступний рядок коду

```
MyDate = DateSerial(1996, 15, 45)
```

привласнює змінною MyDate дату "14.04.1997". Наступний рядок коду

```
MyDate = DateSerial(2035, 2, 3)
```

привласнює змінною MyDate дату "03.02.2035".

Значення дати у вигляді текстового рядка також можна перетворити в значення типу Date. Таке перетворення виконує функція DateValue. Ця функція здатна розпізнавати текстові дати, записані самими різними способами:

```
Dim MyDate As Date
```

```
...
```

```
MyDate = DateValue("11.08.2005")
```

```
MyDate = DateValue("17 Апр 2001")
```

```
MyDate = DateValue("15 березня 99")
```

```
MyDate = DateValue("26/01/2000")
```

Аналогічний набір операцій □оступний відносно значень часу. Перетворення часі доби, заданому у вигляді числа годинника, хвилин і секунд, в значення часу (представлене, як і значення дати, типом Date) виконується функцією TimeSerial:

```
Dim MyTime As Date
```

```
...
```

```
MyTime = TimeSerial(12, 8, 5)
```

В результаті в змінній MyTime міститиметься значення часу "12:08:05". Наступний рядок коду

```
MyTime = TimeSerial(14, 80, 6)
```

привласнює змінною MyTime значення часу "15:20:06". Наступний рядок коду

```
MyTime = TimeSerial(8, 85, 95)
```

привласнює змінною MyTime значення часу "9:26:35". Наступний рядок коду

```
MyTime = TimeSerial(36, 85, 95)
```

привласнює змінною MyTime значення дати-часу "31.12.1899 13:26:35".

Останній приклад ілюструє універсальність і взаємозамінність часу і дати в значеннях дати-часу. Якщо число годинника перевищує одну добу, то у коду "дата-час" з'являється ціла частина числа, що позначає дату. Число 1, як наголошувалося, в "Системі дат 1900" відповідає даті "31.12.1899".

Для перетворення текстових значень часу в код "дата-час" використовують функцію TimeValue. Подібно до функції DateValue, ця функція здатна розпізнавати значення часу, записані різними способами, наприклад:

```
MyTime As Date
```

```
MyTime = TimeValue("12:08:05")
```

В результаті в змінній MyTime міститиметься значення часу "12:08:05". Наступний рядок коду

```
MyTime = TimeValue("3:20:06pm")
```

привласнює змінною MyTime значення часу "15:20:06". Наступний рядок коду

MyTime = TimeValue("9:26:35 AM")  
 привласнює змінною MyTime значення часу "9:26:35".

**Системна дата**, тобто дата "сьогодні" по вбудованому годиннику персонального комп'ютера, доступна для VBA-програм за допомогою функції Date. Ця функція повертає значення типу Date. Наприклад, наступний код обчислює значення дати, **отстоящей** на 100 днів від сьогоднішньої дати:

```
Dim MyDate As Date
...
MyDate = Date + 100 MsgBox MyDate
```

Привласненням значення цій функції можна змінити поточну системну дату. Наприклад, перевести комп'ютерний годинник на 1 день вперед можна за допомогою оператора:

```
Date = Date + 1
```

Функція Time повертає поточне значення системного часу. Наприклад, обчислити значення часу, який буде через 3 години від теперішнього моменту можна за допомогою наступного коду:

```
Dim MyTime As Date
...
MyTime = Time + (3/24) MsgBox MyTime
```

Так само, як і функція Date, функція Time дозволяє змінити системний час, привласненням їй нового значення час.

Нарешті, значення системного часу і дати одночасно повертає функція Now. Функція повертає значення в кодї "дата-час". Наприклад, оператор

```
MsgBox Now
```

виведе на екран повідомлення, що містить в собі одночасне значення дати і часу

Для обчислень із значеннями дати в Visual Basic передбачені спеціальні функції. Взагалі, якщо йдеться про значення дати або часу, перетворених в код "дата-час", то будь-які обчислення зводяться до простих арифметичних дій. Наприклад, просте підсумовування значень типу Date означає підсумовування значень дати і часу. Проте цим не вичерпуються всі можливі потреби в календарних розрахунках, що виникають перед VBA-програмістом.

**Функція DateAdd** дозволяє обчислювати дати і час, що отримуються додаванням до початкового значення дати-часу заданої кількості одиничних інтервалів: років, місяців і т. д., аж до секунд. Другий і третій параметри цієї функції визначають кількість інтервалів, що додаються, і початкове значення. У табл. 10.5 приведені позначення першого параметра, яким задається вид інтервалу.

Таблиця 10.5

Значення першого параметра функції DATEADD

Позначення	Інтервал	Позначення	Інтервал
УУУУ	Рік	<b>D</b>	День місяця
<b>Q</b>	Квартал	<b>W</b>	День тижня
<b>M</b>	Місяць	<b>WW</b>	Тиждень
<b>Y</b>	День року	<b>H</b>	Годинник

Позначення	Інтервал
N	Хвилини

Позначення	Інтервал
S	Секунди

От як, наприклад, можна обчислити значення дати-часу, **отстоящие** на різні інтервали від дати "01/01/2000":

```
Dim MyDate As Date
MyDate = DateAdd("yyyy", 3, "01.01.2000")
```

В результаті змінна MyDate міститиме значення дати "01/01/2003". Наступний рядок коду

```
MyDate = DateAdd("Q", 3, "01.01.2000")
```

привласнює змінною MyDate значення дати "01/10/2000". Наступний рядок коду

```
MyDate = DateAdd("m", 3, "01.01.2000")
```

привласнює змінною MyDate значення дати "01/04/2003". Наступний рядок коду

```
MyDate = DateAdd("ww", 3, "01.01.2000")
```

привласнює змінною MyDate значення дати "22/01/2003". Наступний рядок коду

```
MyDate = DateAdd("H", 3, "01.01.2000")
```

привласнює змінною MyDate значення дати-часу "01/01/2003 3:00:00".

Схожим чином влаштована і діє функція **DateDiff**. Ця функція використовується для вирішення зворотного завдання – для обчислення числа одиничних інтервалів між двома датами. Як параметри їй необхідно задати вид інтервалу (див. табл. 10.5) і дві дати.

Ось приклади обчислень за допомогою функції DateDiff:

```
Dim MyNum As Long
MyNum = DateDiff("yyyy", "01.01.2000", "31.12.2001")
```

В результаті змінна MyNum міститиме число 1. Наступний рядок коду

```
MyNum = DateDiff("Q", "01.01.2000", "31.05.2001")
```

привласнює змінною MyNum число 5. Наступний рядок коду

```
MyNum = DateDiff("m", "01.01.2000", "28.02.2000")
```

привласнює змінною MyNum число 1.

Для обчислення числа тижнів функції DateDiff можна указувати ще один параметр – константу Visual Basic, задаючи спосіб визначення меж тижня: константа vbMonday, наприклад, відповідає тижню, що починається з понеділка, а vbSunday – тижню, що починається з неділі. Наприклад:

```
MyNum = DateDiff("ww", "01.01.2000", "30.07.2000", vbSunday)
```

В результаті змінна MyNum міститиме число 31. Наступний рядок коду

```
MyNum = DateDiff("ww", "01.01.2000", "30.07.2000", vbMonday)
```

привласнює змінною MyNum число 30.

Функцію DatePart використовують, для того, щоб визначити, до якого одиничного інтервалу (року, кварталу, місяцю і т.д.) відноситься задана дата. Як



параметри необхідно вказати вид інтервалу (див. табл. П.7) і власне дату, наприклад:

```
Dim MyNum As Long  
MyNum = DatePart ("yyyy", "01.01.2001")
```

В результаті змінна MyNum міститиме число 2001. Наступний рядок коду

```
MyNum = DatePart ("Q", "01.01.2001")
```

привласнює змінною MyNum число 1. Наступний рядок коду

```
MyNum = DatePart ("m", "01.01.2001")
```

привласнює змінною MyNum число 1.

Точно так, як і у випадку з DateDiff, за допомогою додаткового параметра можна вказати перший день тижня, що може зробити вплив на розрахунки, пов'язані з тижнями:

```
MyNum = DatePart ("ww", "21.01.2001", vbSunday)
```

В результаті змінна MyNum міститиме число 4. Наступний рядок коду

```
MyNum = DatePart ("ww", "21.01.2001", vbMonday)
```

привласнює змінною MyNum число 3.

У Visual Basic передбачений ряд традиційних функцій, що повертають номер певного інтервалу в даті-параметрі. Задавати дату можна довільним чином – ці функції "розуміють" як дати у вигляді тексту, так і значення типу Date.

Функція Year повертає ціле число, відповідне року в заданій даті, наприклад:

```
Dim MyYear As Integer  
MyYear = Year ("26.01.2000")
```

В результаті змінна MyYear міститиме число 2000.

Функція Month повертає ціле число, відповідне місяцю в заданій даті, наприклад:

```
Dim MyMonth As Integer  
MyYear = Month ("26.01.2000")
```

В результаті змінна MyMonth міститиме число 1.

Функція Day повертає ціле число, відповідне дню місяця в заданій даті, наприклад:

```
Dim MyDay As Integer  
MyYear = Day ("26.01.2000")
```

В результаті змінна MyDay міститиме число 26.

Функція WeekDay повертає номер дня тижня в заданій даті. При цьому за допомогою необов'язкового другого параметра можна задати перший день тижня. От як, наприклад, функція WeekDay визначає день 01/01/2001 (понеділок):

```
Dim MyDyOfWeek As Integer  
MyDyOfWeek = WeekDay ("01.01.2001")
```

В результаті змінна MyDyOfWeek міститиме число 2. Наступний рядок коду

```
MyDyOfWeek = WeekDay ("01.01.2001", vbSunday)
```

привласнює змінною MyDyOfWeek число 2. Наступний рядок коду

```
MyDyOfWeek = WeekDay ("01.01.2001", vbMonday)
```

привласнює змінною MyDyOfWeek число 1.

Як легко переконатися, за умовчанням, як перший день тижня використовується неділя.

Схожий набір функцій передбачений для набуття значень кількості тимчасових інтервалів у вигляді цілого числа. Ці функції "розуміють" будь-які допустимі способи завдання часу – від значень типа Date, що повертаються функціями дати-часу (наприклад, Now або Time) до текстових позначень часу.

Функція Hour повертає значення типа Integer, відповідне годиннику в значенні часу, переданому функції як параметр. Наприклад:

```
MsgBox "Зараз " + Str(Hour(Now))+ "-а година"
```

Функція Minute повертає значення типа Integer, відповідне хвилинам в значенні часу, переданому функції як параметр. Наприклад:

```
MsgBox "Зараз " + Str(Minute(Now)) + "-а хвилина"
```

Функція Second повертає значення типа Integer, відповідне хвилинам в значенні часу, переданому функції як параметр. Наприклад:

```
MsgBox "Зараз " + Str(Second(Now))+ "-а секунда"
```

Нарешті, функція Timer повертає значення типа Single, відповідне кількості секунд, що пройшли з останньої півночі. Функція повертає значення з точністю до сотих долей секунди. Наприклад:

```
MsgBox "Поточній добі виповнилося " + Str(Timer) + " секунд"
```

#### 10.3.4. Робота з рядками

У діях з даними особливе місце завжди займає робота з рядковими значеннями. За багато років розвитку мов програмування виробився "джентльменський набір" функцій для виконання різних операцій з символьними рядками. Всі основні функції цього набору є і в Visual Basic.

##### 10.3.4.1. Порівняння рядків

Для порівняння рядків використовують функцію StrComp. Ця функція повертає ціле число із знаком, що характеризує результат порівняння двох рядків, переданих їй як параметри. Третім параметром функції є константа, що визначає режим порівняння. Константа vbTextCompare відповідає текстовому порівнянню рядків, а константа vbBinaryCompare – двійковому. У останньому випадку приймаються до уваги відмінності між прописними і рядковими буквами.

Якщо рядки співпадають, функція повертає значення 0.

Якщо перший рядок більше другою, функція повертає значення 1.

Якщо другий рядок більше першою, функція повертає значення -1.

Наприклад:

```
Dim MyResult As Integer
```

```
MyResult = StrComp("Іванов", "иванов", vbBinaryCompare)
```

В результаті змінна MyResult міститиме число -1. Наступний рядок коду

```
MyResult = StrComp("Іванов", "иванов", vbTextCompare)
```

привласнює змінною MyResult число 0. Наступний рядок коду

```
MyResult = StrComp("иванов", "Іванов", vbBinaryCompare)
```

привласнює змінною MyResult число 1. Наступний рядок коду

```
MyResult = StrComp("ИВАНОВ", "ИВАНОВ", vbBinaryCompare)
```

привласнює змінною MyResult число 0.

#### 10.3.4.2. Виділення підрядка

Для того, щоб витягнути з рядка підрядок, використовується спеціальний набір функцій.

Функція Left повертає підрядок, що складається із заданого числа символів початкового рядка, починаючи з першого. (Якщо рядок містить двійкові дані, то слід використовувати функцію LEFTB). Наприклад:

```
Dim MyString As String  
MyString = Left("АБВГДЕ", 3)
```

В результаті змінна MyString міститиме рядок "АБВ".

Функція Right повертає підрядок, що складається із заданого числа символів початкового рядка, закінчуючи останнім. (Якщо рядок містить двійкові дані, то слід використовувати функцію RIGHTB). Наприклад:

```
Dim MyString As String  
MyString = Right("АБВГДЕ", 3)
```

В результаті змінна MyString міститиме рядок "ДЕ".

Функція Mid повертає підрядок, що складається із заданого числа символів початкового рядка, починаючи із заданої позиції. (Якщо рядок містить двійкові дані, то слід використовувати функцію MIDB). Наприклад:

```
Dim MyString As String  
MyString = Mid("АБВГДЕ", 2, 3)
```

В результаті змінна MyString міститиме рядок "БВГ". Наступний рядок коду

```
MyString = Mid("АБВГДЕ", 5, 1)
```

привласнює змінною MyString рядок "Д".

#### 10.3.4.3. Пошук в рядку

Для пошуку в рядку використовується функція InStr. Ця функція шукає входження підрядка, заданого другим параметром ("строка2"), в рядку, заданому першим параметром ("строка1"). Якщо "строка2" знайшлася всередині "строки1", то функція повертає позицію (порядковий номер символу) "строки2" в "рядку 1". У разі безуспішності пошуку функція повертає значення 0.

Якщо рядок містить двійкові дані, то аналогічну задачу можна вирішити за допомогою функції InStrB.

Наприклад, вираз

```
InStr("АБВГДЕ", "Д")
```

поверне значення 5, а вираз

```
InStr("АБВГДЕ", "Ж")
```

поверне 0, оскільки символу "Ж" в рядку "АБВГДЕ" немає. Вираз

```
InStr("Іванов Іван Іванович", "Іван")
```

поверне значення 1, оскільки перший рядок починається з символів "Іван", а ось вираз

```
InStr("Іванов Іван Іванович" " Іван")
```

поверне 7: хоча поєднання символів " Іван" зустрічається в першому рядку *двічі*, функція повертає позицію *першого* входження підрядка.

Як перший необов'язковий параметр можна вказати порядковий номер символу, з якого слід починати пошук. Наприклад, вираз

```
InStr(8, "Іванов Іван Іванович" " Іван")
```

поверне число 12, оскільки перше входження рядка " Іван" буде пропущений (пошук починається з 8-го символу).

Ось яким чином, наприклад, функцію InStr використовують для того, щоб розділити рядкове значення вигляду "прізвище, ім'я" на два окремі рядки – "ім'я" і "прізвище", користуючись положенням коми, як орієнтиром для розбиття рядка:

```
Dim MyFullName As String
Dim MyLastName As String
Dim MyFirstName As String
...
MyFullName = "Іванов, Іван"
...
MyLastName = Left(MyFullName
InStr(MyFullName ",") - 1)
MsgBox MyLastName
MyFirstName = Right(MyFullName
InStr(MyFullName ",") - 3)
MsgBox MyFirstName
```

У змінну MyLastName потрапить підрядок "Іванов", а в змінну MyFirstName | підрядок "Іван".

#### 10.3.4.4. Перетворення регістра

Для перетворення регістра символів в рядку використовують функції LCase і UCase.

Функція LCase перетворить строку-параметр в нижній регістр (заголовні і рядкові букви стають рядковими), а функція UCase навпаки, перетворить строку-параметр у верхній регістр (заголовні і рядкові букви стають заголовними). Наприклад:

```
Dim MyString As String
MyString = LCase("Довільний Текст")
```

У змінній MyString міститиметься рядок "довільний текст". А це вираз

```
MyString = UCase("Довільний Текст")
```

привласнить змінною MyString рядок "ДОВІЛЬНИЙ ТЕКСТ".

#### 10.3.4.5. Форматування рядка

Для того, щоб вставити в рядок задане число пропусків, використовують функцію Space. Річ не лише в тому, що написати, наприклад, "Space (30)" прос-

тіше, ніж відлічити 30 пропусків. При форматуванні рядкових значень кількість пропусків, що вставляються, може бути величиною змінної, що підлягає обчисленню, і тут вже без функції Space не обійтися. Наприклад:

```
MyString = "Іван" + Space(10)+ "Іванов"
```

В результаті змінна MyString міститиме рядок "Іван Іванов".

Більш універсальним інструментом для вирішення подібних завдань є функція String. Вона повертає рядкове значення, що складається із заданого числа заданих символів, наприклад:

```
MyString = "Іван" + String(10, "-")+ "Іванов"
```

У змінній MyString міститиметься рядок "Іван Іванов".

Функція Len повертає довжину рядка в символах. Наприклад, вираз

```
Len("Іван")
```

поверне значення 4.

Функція Len універсальна, якщо як параметр вказати не рядкову змінну, функція поверне розмір змінної в байтах.

Подивимося, як використовуються ці функції для форматування рядків. Припустимо, необхідно підготувати рядок для висновку на екран або в текстовий файл, причому кожен рядок містить два слова, які необхідно розділити проміжком з ряду крапок, але обидва слова повинні займати фіксовані позиції в рядку. Скажімо, йдеться про таблицю, і обидві колонки повинні мати вирівнювання по лівому краю.

```
Dim MyString As String  
Dim Str1, Str2 As String  
Str1 = "Іван" Str2 = "Іванов"
```

Щоб сформувавши результуючий рядок з рядків Str1 і Str2, використовуємо, по-перше, функцію Space, зробивши відступ першого слова рівним 10:

```
MyString = Space(10) + Str1
```

Зрозуміло, що перед тим, як додати до рядка MyString слово змінній Str2, необхідно вставити якусь кількість крапок. Як обчислити цю кількість? Дуже просто – вираз

```
Len(Str1)
```

поверне довжину першого слова, а різниця

```
30 - Len(Str1)
```

забезпечить таке число символів, яке в сумі з довжиною першого слова завжди дорівнюватиме 30. Таким чином, між першим і другим словом потрібно вставити вираз

```
String(30 - Len(Str1), ".")
```

і отримаємо дві вирівняні по лівому краю колонки незалежно від довжини слів в цих колонках:

```
MyString = Space(10)+ Str1 + String(30 - Len(Str1),  
".")+ Str2
```

Іноді пропуски доводиться не вставляти, а навпаки – видаляти. Наприклад, при порівнянні рядкових значень їх часто доводиться приводити до стандартного вигляду, видаляючи провідні і замикаючі пропуски, а також перетворюючи до одного і тому ж регістра.

Для видалення провідних пропусків використовують функцію LTrim. Вона повертає строку-параметр без провідних пропусків. Замикаючі пропуски аналогічним чином видаляє функція RTrim, а функція Trim видаляє одночасно, і ведучі, і замикаючі пропуски. Наприклад:

```
Dim MyString As String  
MyString = LTrim("Іванов")
```

В результаті в змінній MyString міститиметься рядок "Іванов". А це вираз

```
MyString = RTrim("Іванов")
```

привласнює змінною MyString рядок " Іванов". Наступний рядок

```
MyString = Trim("Іванов")
```

привласнює змінною MyString рядок "Іванов".

Функція Format є, напевно, найбільш могутнім (і, в той же час, найбільш складний) інструментом з арсеналу рядкових функцій Visual Basic.

Як параметри функції Format використовують не тільки рядкові значення, але і значення інших типів, проте повертає вона завжди рядок, причому рядок, що відформатував заданим чином. Різновид форматування задається другим параметром функції. Для форматування, як числових значень, так і значень дати-часу можна використовувати зумовлені формати VBA, а також призначені для користувача форматуючі рядки. Якщо формат не визначений, то функція поверне просто рядок.

#### 10.3.4.6. Перетворення рядкових і символічних значень

Символи в рядкових значеннях мають внутрішнє уявлення у вигляді кодів, –попросту, цілих чисел, кожне з яких відповідає певному символу. Іноді потрібно переходити від кодової "іпостасі" символу до знакової і навпаки.

Для отримання коду символу використовують функцію Asc – ця функція отримала свою назву від коду ASCII, хоча в даний час використовуються також і інші коди.

Функція повертає код першого символу рядка-параметра. Якщо рядок-параметр містить двійкові дані, то слід використовувати функцію ASCB (повертає перший байт рядка) або ASCW (повертає шестнадцятиразрядне слово). Для символів Unicode слід використовувати варіант Asc cW.

Зворотне перетворення, тобто перетворення цілого числа, що представляє код символу, в символічне значення, здійснює функція Chr. Ця функція повертає рядок, що складається з одного символу, код якого заданий параметром. Якщо рядок повинен містити двійкові дані, то слід скористатися функцією CHRВ. Для символів в кодї Unicode, слід скористатися функцією CHRW (якщо система не підтримує Unicode, то функція діє аналогічно Chr).

Вираз

```
Asc ("А")
```

так само, як і вираз

```
Asc ("ABCDEF")
```

поверне цілочисельне (Integer) значення 65 – код латинської букви "А" в стандарті ASCII. Вираз

```
Chr (65)
```

"

поверне рядкове значення "А".

#### 10.4. Основні прийоми роботи з редактором Visual Basic

Багато людей, що щодня працюють в Word або Excel, вважають, що VBA – це "щось для програмістів". Марно вони так думають. У робочому середовищі MS Office є безліч складних інструментів, які, проте, звичайні користувачі застосовують, не замислюючись, і не пускаючись в міркування про те, наскільки це складно. Наприклад, рядовий користувач Word, який просто вводить текст і форматує його у міру свого розуміння, жахнеться, якщо дізнається, наскільки складним може бути форматування звичайного текстового абзацу, і які хитроумні механізми можна в цьому випадку задіювати. Але вся ця *потенційна* складність не заважає йому просто ввести свій текст і віддрукувати документ Word цілком пристойного вигляду, не осягаючи всі "глибини" і тонкощі, які тут можливі. Так само справа йде і з VBA – програмування на VBA в деяких випадках може бути *дуже* складним завданням. Але якщо спочатку не дуже ускладнювати завдання, і не захоплюватися збагненням "глибин", то нічого особливо важкого в цьому процесі немає.

##### 10.4.1. Запуск редактора Visual Basic

Отже, перший крок полягає в тому, щоб відкрити вікно редактора Visual Basic. Для цього існує спеціальна команда в меню *Йабаеп*, проте не будемо поки її використовувати. Для того, щоб найкоротшим шляхом потрапити саме в ту точку редактора Visual Basic, яка потрібна в даному прикладі, потрібно розташувати покажчик миші на ярлику з назвою робочого листа в нижній частині вікна Excel і клацнути *правою* кнопкою миші. Поточним повинен бути саме той лист робочої книги, для якого створюється макрос. В результаті відкриється контекстне меню (Рис. 10.1) даного робочого листа, в якому потрібно виконати команду **Початковий текст (View Code)**. У всіх листів робочої книги контекстне меню виглядає однаково, проте команда **Початковий текст** в меню різних листів приводить до відображення різних вікон в редакторі Visual Basic.

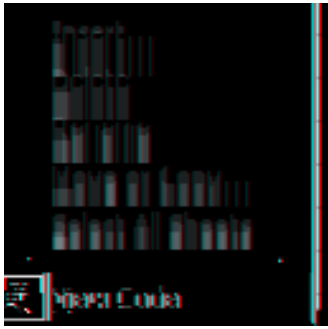


Рис. 10.1. Создание макроса при помощи контекстного меню рабочего листа Excel



Рис. 10.2. Окно кода рабочего листа, при открытии, автоматически создает заготовку текста стандартной процедуры

#### 10.4.2. Створення процедури VBA

В даному випадку, на екрані відобразиться вікно, показане на Рис. 10.2. Не варто лякатися хитромудрої структури редактора Visual Basic – поки можна не звертати уваги на все інше, окрім вікна в центрі екрану, службовця для введення початкового тексту процедур (програм VBA) вибраного робочого листа. Автоматично в цьому вікні створюється заготовка процедури, що містить її заголовок і закінчення. Але в цьому випадку потрібний *не цей* макрос. У верхній частині вікна коду знаходяться два списки, що визначають конкретну процедуру (макрос).

Розверніть лівий список і виберіть в нім пункт **Worksheet**. Потім, в правому списку виберіть пункт **BeforeDoubleClick**, як показано на Рис. 10.3.

В результаті у вікні коду буде створений заголовок потрібної процедури **Worksheet\_BeforeDoubleClick**. Перший і останній рядки процедури створені автоматично, і тепер потрібно ввести початковий текст процедури (лістинг 1.1). Або, як його ще називають, *програмний код*.

##### Лістинг 1.1

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target
As Excel.Range, Cancel As Boolean)
    With ActiveCell
        If .Column = 1 Then
            If .Interior.Pattern = xlPatternNone Then
                .Interior.Patterφ = xlPatternGray25
            Else
                .Interior.Pattern = xlPatternNone
            End If
            Cancel = True
        End If
    End With
End Sub
```



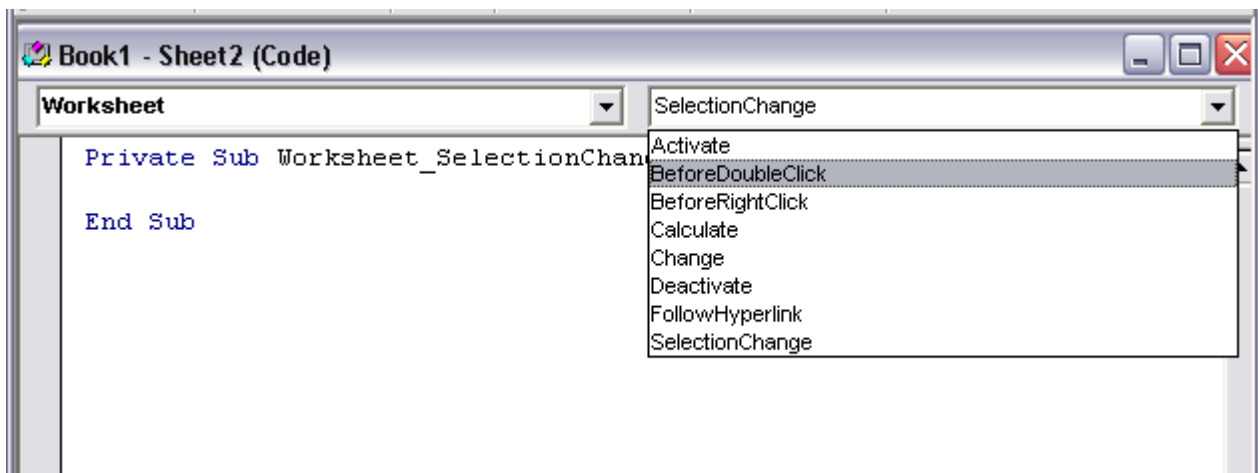


Рис. 10.3. За допомогою списків **Object** і **Procedure** можна створити заго-  
тівку стандартної процедури

На Рис. 10.4 показано, як в результаті належного виглядати вікно коду.

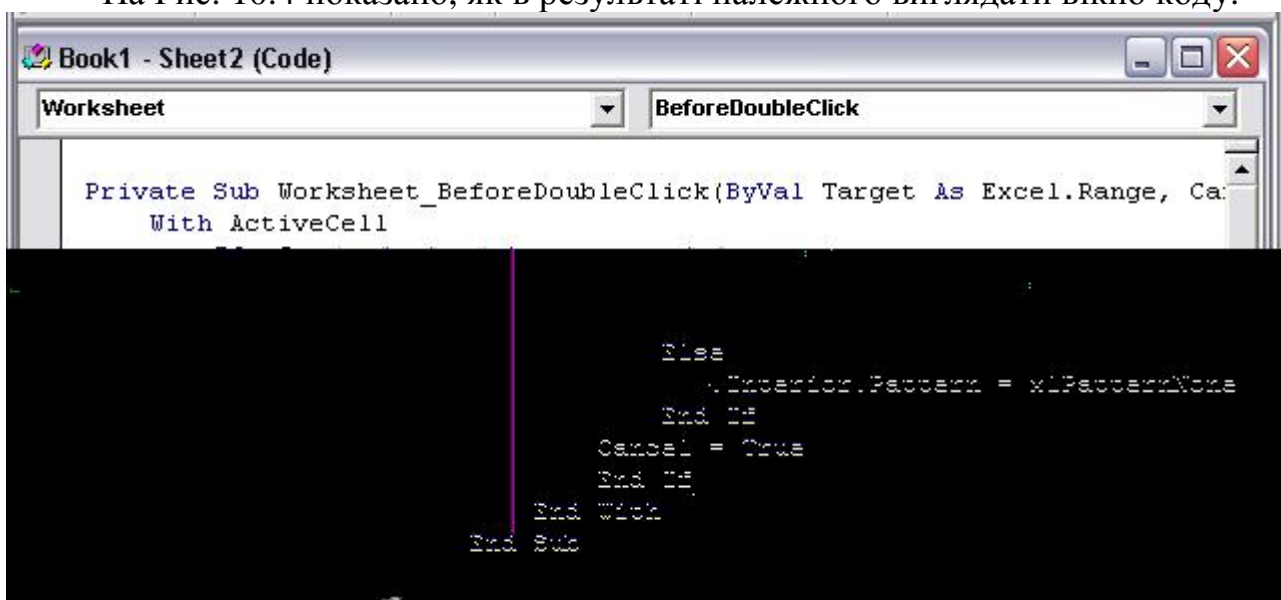


Рис. 10.4. Процедура Worksheet\_BeforeDoubleClick

### 10.4.3. Введення початкового тексту процедури

Редактор Visual Basic аналізує текст, що вводиться користувачем, на предмет помилок. У міру введення якого-небудь слова редактор пропонує найближчий відповідний варіант синтаксично правильного виразу мови. Наприклад, якщо почати введення слова Pattern, редактор, поряд з курсором введення, відобразить список, в якому буде знайдений і виділений найближчий відповідний варіант до вже введеної частини слова. В цьому випадку можна продовжувати вводити решту букв цього слова, не звертаючи уваги на поведінку вікна, або ж ввести відразу символ, наступний за цим словом, наприклад "=", тоді залишок слова Pattern буде введений автоматично (Рис. 10.5).

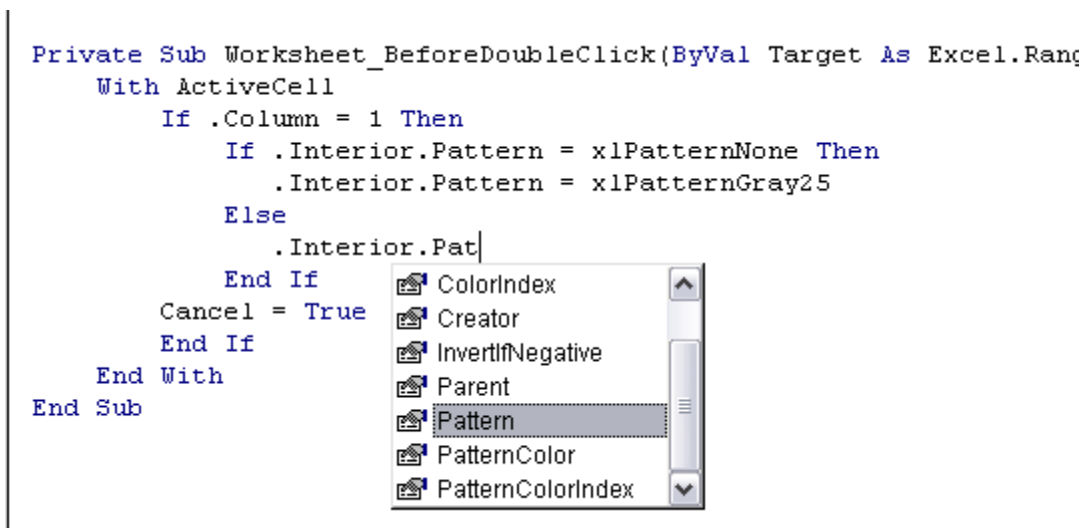


Рис. 10.5. Редактор Visual Basic пропонує свою допомогу – поряд з курсором введення з'являється список імен, які допустимі в даному виразі

У наступному рядку програмного коду допоміжне вікно з'явиться двічі – перед словом Interior і після нього. При *друкарських помилках* або навіть просто спробах перекласти курсор введення з нескінченного рядка редактор може розкрити вікно з попередженням про помилку – в цьому випадку треба клацнути на кнопці ОК цього вікна і виправити помилку. Помилково набране слово буде виділено червоним кольором.

Нарешті, початковий текст макросу введений, і можна випробувати його в справі. Для повернення у вікно Excel можна скористатися кнопкою на панелі завдань або спеціальною кнопкою із значком Excel, розташованою в лівому верхньому кутку вікна редактора Visual Basic (Рис. 10.6).

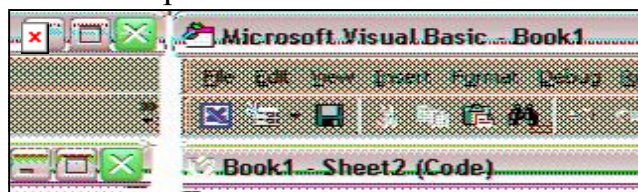


Рис. 10.6. Повернення в Excel

Тепер спробуємо виконати подвійне клацання на одній з клітинок стовпця "А"... Ура, працює! Клітинка заштрихована. Ще одне подвійне клацання і штрихування зникло. Тепер можемо легко і просто відзначати потрібні позиції документа.

#### 10.4.4. Аналіз стандартної процедури обробки події – подвійного клацання мишею на клітинку листа Excel

Подивимося тепер на текст процедури VBA, яка була створена в попередніх розділах (лістинг 1.1). Користувачеві, що не мав раніше справ з VBA, на перший погляд, код може показатися суцільною "китайською грамотою", проте на практиці все не так складно, як може показатися. Давайте розберемося.

У двох списках були вибрані пункти Worksheet і BeforeDoubleClick – в результаті процедура (Sub) стала називатися Worksheet\_BeforeDoubleClick. Тут Worksheet – це стандартний об'єкт, для якого створювалася дана процедура – робочий лист. BeforeDoubleClick – це стандартна *подія* робочого листа, подія ця відбувається при кожному подвійному клацанні на листі. Якби в правому спис-

ку було вибрано іншу подію, наприклад, Activate, то код процедури виконувався б при активації листа, тобто в мить, коли лист стає поточним. Але в нашому випадку код виконуватиметься при подвійних клацаннях миші на будь-якій клітинці даного листа.

Що ж, власне, *робить* код процедури?

```
With ActiveCell
```

```
...
```

```
...
```

```
End With
```

Перший і останній рядки визначають, що укладений між ними, код відноситься до об'єкту ActiveCell, тобто до поточної клітинки. Саме вона нас і цікавить. Оскільки будь-яка клітинка, на якій виконано подвійне клацання миші, автоматично стає поточною.

Оператор

```
If .Column = 1 Then
```

```
...
```

```
End If
```

означає "якщо-то" і обмежує виконання процедури межами стовпця A. Column – це *властивість* будь-якої клітинки і вказує вона на номер стовпця, до якого ця клітинка відноситься. Стовпець "A" розташований в даній таблиці першим і якщо подвійне клацання виконане на клітинці іншого стовпця, то .Column виявиться *не* рівним 1, і подальший код не буде виконаний.

Наступний, самий внутрішній оператор, виглядає складнішим. If-Then-Else означає "якщо-то-інакше" і у вольному перекладі звичайною мовою оператори

```
If .Interior.Pattern = xlPatternNone Then
```

```
    .Interior.Pattern = xlPatternGray25
```

```
Else
```

```
    .Interior.Pattern = xlPatternNone
```

```
End If
```

означають: якщо клітинка не заштрихована, то задати їй колір 25%-й сірий, інакше, тобто, якщо вона вже заштрихована кольором 25%-м сірим, відмінити штрихування клітинки.

I, нарешті, маленький рядок

```
Cancel = True
```

грає тут допоміжну роль. Змінна Cancel є *параметром* процедури, і сенс цього параметра такий: привласнивши йому логічне значення *True* (*Істина*), користувач як би повідомляється робочому листу Excel, що подвійне клацання відмінюється, тобто його не було. Тому Excel не перейде по подвійному клацанню до редагування клітинки, як він це зазвичай робить.

## 10.5. Індивідуальне завдання №11.

### Робота з VBA в електронних таблицях Excel

За списком групи обрати свій варіант програмування додаткових функцій електронної таблиці Excel.

№ п/п	Завдання
1.	Побудувати кнопку звертання до функції „Пошук рішення”
2.	Створити програму автоматичного розрахунку кошторису
3.	Автоматично визначити тип клітинки і надрукувати текст, який показує цей тип
4.	Написати програму транспонування матриці довільного розміру
5.	Створити автономну програму обліку числа клієнтів на фірмі
6.	Визначити спосіб розрахунку експоненційних чисел
7.	Побудувати форму введення анкетних даних співробітників
8.	Побудувати форму автоматичного формування договору
9.	Побудувати форму введення даних про устаткування
10.	Створити програму автоматичного розрахунку амортизації устаткування
11.	Створити меню, що випадає, яке містить дані про спеціальності
12.	Побудувати форму введення даних про матеріали
13.	Створити меню, що випадає, яке містить дані про свійських тварин
14.	Побудувати форму введення договору
15.	Створити програму автоматичного розрахунку коефіцієнтів лінійної регресії
16.	Створити меню, що випадає, яке містить дані про зміст цього завдання
17.	Побудувати форму введення накладної на матеріали
18.	Створити меню, що випадає, яке містить дані про список газет
19.	Створити програму автоматичного розрахунку кошторису витрат на подорож
20.	Створити меню, що випадає, яке містить дані про перелік крамниць
21.	Створити програму автоматичного розрахунку норми витрат матеріалів на одиницю продукції
22.	Побудувати форму введення балансу
23.	Створити меню, що випадає, яке містить дані про перелік готелів
24.	Створити програму автоматичного розрахунку накладних витрат
25.	Створити меню, що випадає, яке містить дані про співробітників
26.	Побудувати форму введення довіреності
27.	Створити меню, що випадає, яке містить дані про устаткування
28.	Створити програму автоматичного розрахунку собівартості продукції
29.	Створити меню, що випадає, яке містить дані про матеріали

№ п/п	Завдання
30.	Побудувати форму введення накладної на надходження коштів

### Контрольні запитання

1. Що таке VBA?
2. Як застосовуються константи?
3. Як застосовуються змінні?
4. Поняття масиву.
5. Поняття об'єктної змінної.
6. Перелічте основні типи даних.
7. Порівняння Integer та Long-змінної.
8. Порівняння Single та Double-змінної.
9. Boolean та Currency змінні.
10. Відмінності змінних Date та String.
11. Арифметичні операції в VBA.
12. Логічні операції в VBA.
13. Операції порівняння в VBA.
14. Як здійснити перехід по мітці GoTo.
15. Синтаксис операторів циклу.
16. Функції для роботи з інтерфейсом користувача.
17. Вбудовані функції взаємодії з операційною системою.
18. Які існують оператори для читання з відкритого файлу?
19. Застосування операторів часу.
20. Які функції роботи з рядками?
21. Функції виділення підрядка.
22. Особливості пошуку у рядку.
23. Форматування рядка операторами VBA.
24. Запуск редактора VBA у середовищі MS Office.
25. Порядок створення процедури.
26. Особливості введення тексту процедури.
27. Перелічте можливі проблеми роботи з редактором VBA.

*Наведено основні типи чисел та оператори, які їх обробляють, процедури доступу до файлів та порядок застосування редактора VBA.*

# 11. ЕКСПЕРТНІ ТА НАВЧАЛЬНІ СИСТЕМИ

*Вивчивши матеріали цього розділу студент має опанувати основні поняття про інтелектуальні інформаційні системи.*

## 11.1. Класифікація інтелектуальних інформаційних систем

Інтелектуальна інформаційна система (ІС) заснована на концепції використання бази знань для генерації алгоритмів розв'язування задач різних класів в залежності від конкретних інформаційних потреб користувачів.

Для ІС характерні наступні ознаки:

- розвинені комунікативні здібності;
- вміння розв'язувати слабо формалізовані задачі;
- здатність до самонавчання;
- адаптивність.

Кожній з перерахованих ознак умовно відповідає свій клас ІС. Різні системи можуть володіти одним або декількома ознаками інтелектуальності з різним ступенем прояву.

Засоби ІІ можуть використовуватися для реалізації різних функцій, виконуваних ІС. Існує класифікація ІС, ознаками якої є наступні інтелектуальні функції:

- *комунікативні здатності* — спосіб взаємодії кінцевого користувача з системою;
- *вирішення складних задач, що погано формалізуються*, які вимагають побудови оригінального алгоритму рішення залежно від конкретної ситуації, що характеризується, невизначеністю і динамічністю початкових даних і знань;
- *здатність до самонавчання* — уміння системи автоматично витягувати знання з накопиченого досвіду і застосовувати їх для вирішення задач;
- *адаптивна* — здібність системи до розвитку відповідно до об'єктивних змін області знань.

### 11.1.1. Системи з інтелектуальним інтерфейсом

Застосування ІІ для посилення комунікативних здібностей інформаційних систем привело до появи систем з інтелектуальним інтерфейсом, серед яких можна виділити наступні типи.

1. *Інтелектуальні бази даних*. Дозволяють на відміну від традиційних БД забезпечувати вибірку необхідної інформації, не присутньої в явному вигляді, а що виводиться з сукупності даних, що зберігаються.

2. *Природно-мовний інтерфейс*. Застосовується для доступу до інтелектуальних баз даних, контекстного пошуку документальної текстової інформації, голосового введення команд в системах управління, машинного перекладу з

іноземних мов. Для реалізації *природно-мовного* інтерфейсу необхідно розв'язати проблеми морфологічного, синтаксичного і семантичного аналізу, а також задачу синтезу висловів на природній мові. При морфологічному аналізі здійснюються розпізнавання і перевірка правильності написання слів в словнику. Синтаксичний контроль припускає розкладання вхідних повідомлень на окремі компоненти, перевірку відповідності граматичним правилам внутрішнього представлення знань і виявлення частин, яких бракує. Семантичний аналіз забезпечує встановлення смислової правильності синтаксичних конструкцій. На відміну від аналізу синтез висловів полягає в перетворенні цифрового представлення інформації в уявлення на природній мові.

3. *Гіпертекстові системи.* Використовуються для реалізації пошуку за ключовими словами в базах даних з текстовою інформацією. Для повнішого віддзеркалення різних смислових відносин термінів, потрібна складна, семантична організація ключових слів. Рішення цих задач здійснюється за допомогою інтелектуальних гіпертекстових систем, в яких механізм пошуку спочатку працює з базою знань ключових слів, а потім — з самим текстом. Аналогічним чином проводиться пошук мультимедійної інформації, що включає окрім тексту графічну інформацію, аудіо- і відеообрази.

4. *Системи контекстної допомоги.* Відносяться до класу систем розповсюдження знань. Такі системи є, як правило, додатками до документації. Системи контекстної допомоги — окремий випадок гіпертекстових і природно-мовних систем. У них користувач описує проблему, а система на основі додаткового діалогу конкретизує її і виконує пошук рекомендацій, що відносяться до ситуації. У звичних гіпертекстових системах, навпаки, комп'ютерні додатки навіають користувачу схему пошуку необхідної інформації.

5. *Системи когнітивної графіки.* Орієнтовані на спілкування з користувачем ПС за допомогою графічних образів, які генеруються відповідно до змін параметрів модельованих або спостережуваних процесів. Когнітивна графіка дозволяє в наочному і виразному вигляді представити безліч параметрів, що характеризують явище, що вивчається, звільняє користувача від аналізу тривіальних ситуацій, сприяє швидкому освоєнню програмних засобів і підвищенню конкурентоспроможності розробляються ПС. Застосування когнітивної графіки особливо актуальне в системах моніторингу і оперативного управління, в почальних і тренажерних системах, в оперативних системах ухвалення рішень, що працюють в режимі реального часу.

### 11.1.2. Експертні системи

Експертні системи як самостійний напрям в штучному інтелекті сформувався в кінці 1970-х рр. Історія ЕС почалася з повідомлення японського комітету з розробки ЕОМ п'ятого покоління, в якому основна увага надавалася розвитку «інтелектуальних здібностей» комп'ютерів з тим, щоб вони могли оперувати не тільки даними, але і знаннями, як це роблять фахівці (експерти) при виробленні висновків. Група по експертних системах при Комітеті British Computer Society визначила ЕС як «втільнення в ЕОМ компоненти досвіду експерта, заснованої на знаннях, в такій формі, що машина може дати інтелектуальну раду або ухвали-

вши рішення щодо оброблюваної функції». Однією з важливих властивостей ЕС є здатність пояснити хід своїх міркувань зрозумілим для користувача чином.

Область дослідження ЕС називають «інженерією знань». Цей термін був введений Е. Фейгенбаумом і в його трактуванні означає той, що «привнесе принципи й інструментарій з області штучного інтелекту в рішення важких прикладних проблем, які вимагають знань експертів». Іншими словами, ЕС застосовуються для вирішення неформалізованих проблем, до яких відносять задачі, що володіють однією (або декількома) з наступних характеристик:

- задачі не можуть бути представлені в числовій формі;
- початкові дані і знання про наочну область володіють неоднозначністю, неточністю, суперечністю;
- цілі не можна виразити за допомогою чітко певної цільової функції;
- не існує однозначного алгоритмічного рішення задачі;
- алгоритмічне рішення існує, але його не можна використовувати унаслідок великої розмірності простору рішень і обмежень на ресурси (часу, пам'яті).

Головна відмінність ЕС і систем штучного інтелекту від систем обробки даних полягає у тому, що в них використовується символічний, а не числовий спосіб представлення даних, а як методи обробки інформації застосовуються процедури логічного висновку і евристичного пошуку рішень.

ЕС охоплюють самі різні наочні області, серед яких лідирують бізнес, виробництво, медицина, проектування і системи управління.

У багатьох випадках ЕС є інструментом, що підсилює інтелектуальні здібності експерта. Крім того, ЕС може виступати в ролі:

- консультанта для недосвідчених або непрофесійних користувачів;
- асистента експерта-людини в процесах аналізу варіантів рішень;
- партнера експерта в процесі рішення задач, що вимагають залучення знань з різних наочних областей.

Для класифікації ЕС використовуються наступні ознаки:

- спосіб формування рішення;
- спосіб обліку тимчасової ознаки;
- вид використовуваних даних і знань;
- число використовуваних джерел знань.

За способом формування рішення ЕС можна розділити на ті, що аналізують і синтезують. У системах першого типу здійснюється вибір рішення з безлічі відомих рішень на основі аналізу знань, в системах другого типу рішення синтезується з окремих фрагментів знань.

Залежно від способу обліку тимчасової ознаки ЕС ділять на статичні і динамічні. Статичні ЕС призначені для вирішення задач з незмінними в процесі рішення даними і знаннями, а динамічні ЕС допускають такі зміни.

По видах використовуваних даних і знань розрізняють ЕС з детермінованими і невизначеними знаннями. Під невизначеністю знань і даних розуміються їх неповнота, ненадійність, нечіткість.

ЕС можуть створюватися з використанням одного або декількох джерел знань.



Відповідно до перерахованих ознак можна виділити чотири основні класи ЕС: класифікуючі, довизначальні, трансформуючі і мультиагентні (табл. 11.1).

Таблиця 11.1

	Аналіз	Синтез	
Детермінованість знань	Класифікуючі	Трансформуючі	Одне джерело знань
Невизначеність знань	Довизначальні	Мультиагентні	Декілька джерел знань
	Статика	Динаміка	

Рис. 11.1 Основні класи експертних систем

*Класифікуючі* ЕС вирішують задачі розпізнавання ситуацій. Основним методом формування рішень в таких системах є дедуктивний логічний висновок.

*Довизначальні* ЕС використовуються для вирішення задач з не повністю певними даними і знаннями. У таких ЕС виникають задачі інтерпретації нечітких знань і вибору альтернативних напрямів пошуку в просторі можливих рішень. Як методи обробки невизначених знань можуть використовуватися баєсівський підхід вірогідності, коефіцієнти упевненості, нечітка логіка.

*Трансформуючі* ЕС відносяться до синтезуючих динамічних експертних систем, в яких передбачається перетворення знань в процесі рішення задач, що повторюється. У ЕС даного класу використовуються різні способи обробки знань:

- генерація і перевірка гіпотез;
- логіка припущень і умовчань (коли за неповними даними формуються уявлення про об'єкти певного класу, які згодом адаптуються до конкретних умов ситуацій, що змінюються);
- використання метазнань (загальніших закономірностей) для усунення невизначеностей в ситуаціях.

*Мультиагентні* системи - це динамічні ЕС, засновані на інтеграції декількох різнорідних джерел знань. Ці джерела обмінюються між собою одержуваними результатами в ході рішення задач. Системи даного класу мають наступну нагоду:

- реалізація альтернативних міркувань на основі використання різних джерел знань і механізму усунення суперечностей;
- розподілене рішення проблем, які розділяються на паралельно вирішувані підзадачі з самостійними джерелами знань;
- застосування різних стратегій висновку висновків залежно від типу вирішуваної проблеми;
- обробка великих масивів інформації з баз даних;
- використання математичних моделей і зовнішніх процедур для імітації розвитку ситуацій.

### 11.1.3. Самонавчальні системи

Самонавчальні інтелектуальні системи засновані на методах автоматичної класифікації ситуацій з реальної практики, або на методах навчання на прикладах. Приклади реальних ситуацій складають так звану повчальну вибірку, яка формується протягом певного історичного періоду. Елементи повчальної вибірки описуються безліччю класифікаційних ознак.

Стратегія «навчання з вчителем» припускає завдання фахівцем для кожного прикладу значень ознак, що показують його приналежність до певного класу ситуацій. При навчанні «без вчителя» система повинна самостійно "виділяти класи ситуацій по ступеню близькості, значень класифікаційних ознак.

В процесі навчання проводиться автоматична побудова узагальнених правил або функцій, що описують приналежність ситуацій до класів, якими система згодом користуватиметься при інтерпретації незнайомих ситуацій. З узагальнених правил, у свою чергу, автоматично формується база знань, яка періодично коректується у міру накопичення інформації про аналізовані ситуації.

Побудовані відповідно до цих принципів самонавчальні системи мають наступні недоліки:

- відносно низьку адекватність баз знань, які виникають у реальних проблемах через неповноту та/або зашумленість навчальної вибірки;
- низький ступінь з'ясовної одержаних результатів;
- поверхневий опис проблемної області і вузьку спрямованість застосування через обмеження в розмірності простору ознак.

*Індуктивні системи* дозволяють узагальнювати приклади на основі принципу індукції «від приватного до загального». Процедура узагальнення зводиться до класифікації прикладів по значущих ознаках. Алгоритм класифікації прикладів включає наступні основні кроки.

1. Вибір класифікаційної ознаки з безлічі заданих.
2. Розбиття безлічі прикладів на підмножини по значенню вибраної ознаки.
3. Перевірка приналежності кожної підмножини прикладів одному з класів.
4. Перевірка закінчення процесу класифікації. Якщо якась підмножина прикладів належить одному підкласу, тобто у всіх прикладів цієї підмножини співпадає значення класифікаційної ознаки, то процес класифікації закінчується.
5. Для підмножин прикладів з не співпадаючими значеннями класифікаційних ознак процес розпізнавання продовжується, починаючи з першого кроку. При цьому кожна підмножина прикладів стає множиною, що класифікується. Нейронні мережі є класичним прикладом технології, заснованої на прикладах. Нейронні мережі – узагальнена назва групи математичних алгоритмів, що володіють здатністю навчатися на прикладах; «взнаючи» згодом риси зустрінутих зразків і ситуацій. Завдяки цій здатності нейронні мережі використовуються при рішенні задач обробки сигналів і зображень, розпізнавання образів, а також для прогнозування.

Нейронна мережа — це кібернетична модель нервової системи, яка є сукупністю великого числа порівняно простих елементів - нейронів, топологія з'єднання яких залежить від типу мережі. Щоб створити нейронну мережу для вирішення якої-небудь конкретної задачі, слід вибрати спосіб з'єднання нейронів один з одним і підібрати значення параметрів з'єднань поміж нейронами.

У системах, заснованих на прецедентах, БЗ містить описи конкретних ситуацій (прецеденти). Пошук рішення здійснюється на основі аналогій і включає наступні етапи:

- отримання інформації про поточну проблему;
- зіставлення одержаної інформації із значеннями ознак прецедентів з бази знань;
- вибір прецеденту з бази знань, найближчого до даної проблеми;
- адаптація вибраного прецеденту до поточної проблеми;
- перевірка коректності кожного одержаного рішення;
- занесення детальної інформації про одержане рішення в БЗ.

Прецеденти описуються безліччю ознак, по яких будуються індекси швидкого пошуку. Проте в системах, заснованих на прецедентах, на відміну від індуктивних систем допускається нечіткий пошук з отриманням безлічі допустимих альтернатив, кожна з яких оцінюється деяким коефіцієнтом упевненості. Найефективніші рішення адаптуються до реальних ситуацій за допомогою спеціальних алгоритмів.

Системи, засновані на прецедентах, застосовуються для розповсюдження знань і в системах контекстної допомоги.

*Інформаційні сховища* відрізняються від інтелектуальних баз даних, тим, що є сховищами значущої інформації, яку регулярно запитують з оперативних баз даних. Сховище даних — це наочно-орієнтоване, інтегроване, прив'язане до часу, незмінні збори даних, вживаних для підтримки процесів ухвалення управлінських рішень. Наочна орієнтація означає, що дані об'єднані в категорії і зберігаються відповідно до тих областей, які вони описують, а не з додатками, які їх використовують. У сховищі дані інтегруються в цілях задоволення вимог підприємства в цілому, а не окремої функції бізнесу. Прихильність даних до часу виражає їх «історичність», тобто атрибут часу завжди явно присутній в структурах сховища даних. Незмінність означає, що, потрапивши одного разу в сховище, дані вже не змінюються на відміну від оперативних систем, де дані присутні тільки в останній версії, тому постійно міняються.

Технології витягання знань з сховищ даних засновані на методах статистичного аналізу і моделювання, орієнтованих на пошук моделей і відносин, прихованих в сукупності даних. Ці моделі можуть надалі використовуватися для оптимізації діяльності підприємства або фірми.

Для витягання значущої інформації з сховищ даних є спеціальні методи (OLAP-аналізу, Data Mining або Knowledge Discovery), засновані на застосуванні методів математичної статистики, нейронних мереж, індуктивних методів побудови дерев рішень і ін.

Технологія OLAP (On-Line Analytical Processing - оперативний аналіз даних) надає користувачу засобу для формування і перевірки гіпотез про власти-

вості даних або відносинах між ними на основі різноманітних запитів до бази даних. Вони застосовуються на ранніх стадіях процесу витягання знань, допомагаючи аналітику сфокусувати увагу на важливих змінних. Засоби Data Mining відрізняються від OLAP тим, що окрім перевірки передбачуваних залежностей вони здатні самостійно (без участі користувача) генерувати гіпотези про закономірності, існуючі в даних, і будувати моделі, що дозволяють кількісно оцінити ступінь взаємного впливу досліджуваних чинників на основі наявної інформації.

#### 11.1.4. Адаптивні інформаційні системи

Потреба в адаптивних інформаційних системах виникає в тих випадках, коли підтримувані ними проблемні області постійно розвиваються. У зв'язку з цим адаптивні системи повинен задовольняти ряд специфічних вимог, а саме:

- адекватно відображати знання проблемної області в кожен момент часу;
- годитися для легкої і швидкої реконструкції при зміні проблемного середовища.

Адаптивні властивості інформаційних систем забезпечуються за рахунок інтелектуалізації їх архітектури. Ядром таких систем є модель проблемної області, що постійно розвивається, підтримувана в спеціальній базі знань – репозиторії. Ядро системи управляє процесами генерації або зміни конфігурації програмного забезпечення.

В процесі розробки адаптивних інформаційних систем застосовується оригінальне або типове проектування. Оригінальне проектування припускає розробку інформаційної системи з «чистого листу» на основі сформульованих вимог. Реалізація цього підходу заснована на використуванні систем автоматизованого проектування, або *CASE-технології*.

При типовому проектуванні здійснюється адаптація типових розробок до особливостей проблемної області. Для реалізації цього підходу застосовуються інструментальні засоби *компонентного (складального) проектування* інформаційних систем (R/3, BAAN IV, Prodis і ін.).

Головна відмінність підходів полягає у тому, що при використуванні *CASE* - технології на основі репозиторія при зміні проблемної області кожного разу виконується *генерація* програмного забезпечення, а при використуванні складальної технології — *конфігурація* програм і лише в окремих випадках їх *переробка*.

#### 11.2. Технології розробки експертних систем

Технологія створення інтелектуального програмного забезпечення істотно відрізняється від розробки традиційних програм з використанням відомих алгоритмічних мов (табл. 11.2).

Розглянемо відпрацьовані на сьогоднішній день елементи технології створення ІС на прикладі розробки експертних систем. Цей вибір обумовлений тим, що ІС одержали вельми широке розповсюдження в багатьох сферах людської діяльності, а технології їх створення мають універсальний характер і не вимагають апаратних реалізацій.

*Експертними системами* називають складні програмні комплекси, що акумулюють знання фахівців в конкретних наочних областях і тиражують цей емпіричний досвід для консультацій менш кваліфікованих користувачів

Таблиця 11.2

**Відмінності систем штучного інтелекту від звичних програмних систем**

Характеристика	Програмування в системах штучного інтелекту	Традиційне програмування
Тип обробки	Символьний	Числовий
Метод	Евристичний пошук	Точний алгоритм
Завдання кроків рішення	Неявне	Явне
Шукане рішення	Задовільне	Оптимальне
Управління і дані	Змішані	Розділені
Знання	Неточні	Точні
Модифікації	Часті	Рідкісні

У найперших ЕС не враховувалося зміна знань, використовуваних в процесі рішення конкретної задачі. Їх назвали *статичними ЕС*. Типова статична ЕС містить наступні основні компоненти:

- базу знань;
- робочу пам'ять, звану також базою даних;
- вирішувач (інтерпретатор);
- систему пояснень;
- компоненти придбання знань;
- інтерфейс з користувачем.

*База знань* ЕС призначена для зберігання довгострокових даних, що описують дану область, і правил, що описують доцільні" перетворення "даних "цієї області.

*База даних* служить для зберігання поточних даних вирішувача задачі.

*Вирішувач* (інтерпретатор) формує послідовність застосування правил і здійснює їх обробку, використовуючи дані з робочої пам'яті і знання з БЗ.

*Система пояснень* показує, яким чином система одержала рішення задачі і які знання при цьому використовувалися. Це полегшує тестування системи і підвищує довіру користувача до одержаного результату.

*Компоненти придбання знань* необхідні для заповнення ЕС знаннями в діалозі з користувачем-експертом, а також для додавання і модифікації закладених в систему знань.

До розробки ЕС притягуються фахівці з різних наочних областей, а саме:

- експерти тієї проблемної області, до якої відносяться задачі, вирішувані ЕС;
- інженери по знаннях, що є фахівцями з розробки ПС;
- програмісти, що здійснюють реалізацію ЕС. *Експерти* поставляють знання в ЕС і оцінюють правильність одержуваних результатів.

*Інженери по знаннях* допомагають експертам виявити і структурувати знання, необхідні для роботи ЕС, виконують роботу за уявленням знань, вибирають методи обробки знань, проводять вибір інструментальних засобів для реалізації ЕС, найпридатніших для вирішення поставлених задач.

*Програмісти* розробляють програмне забезпечення ЕС і здійснюють його сполучення з середовищем, в якій воно використовуватиметься.

Будь-хто ЕС повинен мати, принаймні, два режими роботи. У *режимі придбання знань* експерт наповнює систему знаннями, які згодом дозволять ЕС самотійно (без допомоги експерта) вирішувати певні задачі з конкретної проблемної області. Експерт описує проблемну область у вигляді сукупності даних і правил. Дані визначають об'єкти, їх характеристики і значення, існуючі у області експертизи. Правила визначають, взаємні зв'язки, існуючі між даними, і способи маніпулювання даними, характерні для даного класу задач.

У *режимі консультації* користувач ЕС повідомляє системі конкретні дані про вирішувану задачу і прагне одержати з її допомогою результат. Користувачі-неспіціалісти звертаються до ЕС за результатом, не уміючи одержати його самотійно, користувачі-фахівці використовують ЕС для прискорення і полегшення процесу отримання результату. Слід підкреслити, що термін «користувач» є багатозначним, оскільки використовувати ЕС можуть і експерт, і інженер по знаннях, і програміст. Тому, коли хочуть підкреслити, що йдеться про те, для кого робилася ЕС, використовують термін «кінцевий користувач».

У режимі консультації вхідні дані про задачу поступають в робочу пам'ять. Вирішувач на основі вхідних даних з робочої пам'яті і правил з БЗ формує рішення. На відміну від традиційних програм комп'ютерної обробки даних ЕС при рішенні задачі не тільки виконує наказану послідовність операцій, але і сама формує її.

Існує широкий клас додатків, в яких вимагається враховувати зміни, що відбуваються в навколишньому світі за час виконання додатку. Для вирішення таких задач необхідно застосовувати *динамічні ЕС*, які разом з компонентами статичних систем містять підсистему моделювання зовнішнього світу і підсистему зв'язку із зовнішнім оточенням. Підсистема моделювання зовнішнього світу необхідна для прогнозування, аналізу і адекватної оцінки стану зовнішнього середовища. Зміни оточення вирішуваної задачі вимагають зміни знань, що зберігаються в ЕС, для того, щоб відобразити тимчасову логіку подій, що відбуваються в реальному світі. Компоненту зв'язку із зовнішнім світом актуальна для автономних інтелектуальних систем (роботів), а також для інтелектуальних систем управління. Зв'язок із зовнішнім світом здійснюється через систему датчиків і контролерів.

### 11.2.1. Класифікаційні ознаки експертних систем

У основі класифікації експертних систем (називатимемо їх додатками) лежать наступні параметри:

- тип додатку;
- стадія існування;
- масштаб;
- тип проблемного середовища;
- тип вирішуваної задачі.

**Тип додатку.** Характеризується наступними ознаками.

1. Можливість взаємодії додатку з іншими програмними засобами:

- *ізолюваний додаток* – ЕС, не здатна взаємодіяти з іншими програмними системами (наприклад, з БД, електронними таблицями, пакетами прикладних програм, контролерами, датчиками і т.п.);

- *інтегрований додаток* – ЕС і інші програмні системи, з якими вона взаємодіє в ході роботи. Більшість сучасних ЕС, використовуваних для вирішення практично значущих задач, є інтегрованими.

2. Можливість виконувати додаток на різномірній апаратурі і переносити його на різні платформи:

- *закриті додатки* - виконуються тільки в програмному середовищі даної фірми і можуть бути перенесені на інші платформи тільки шляхом того, що перепрограмує додатку;

- *відкриті додатки* — орієнтовані на виконання в різномірному програмно-апаратному оточенні і можуть бути перенесені на інші платформи без того, що перепрограмує.

3. Архітектура додатку:

- *централізований додаток* — реалізується на базі центральної ЕОМ, з якою пов'язані термінали;

- *розподілений додаток* - звичайно використовується архітектура клієнт-сервер.

**Стадія існування.** Характеризує ступінь завершеності розробки ЕС. Прийнято виділяти наступні стадії:

- *дослідницький прототип* - вирішує представницький клас задач проблемної області, але може бути нестійкий в роботі і не повністю перевірений. За наявності розвинених інструментальних засобів для розробки дослідницького прототипу потрібно приблизно 2-4 місяці. База знань дослідницького прототипу звичайно містить невелике число виконуваних тверджень;

- *діючий прототип* - надійно вирішує будь-які задачі проблемної області, але при вирішенні складних задач може зажадати надмірно багато часу і (або) пам'яті. Доведення системи; від початку розробки, до стадії діючого прототипу вимагає приблизно 6-9 місяців, при цьому кількість виконуваних тверджень в БЗ збільшується в порівнянні з дослідницьким прототипом;

- *промислова система* — забезпечує високу якість рішення всіх задач при мінімумі часу і пам'яті. Звичайно процес перетворення діючого прототипу в промислову систему полягає в розширенні бази знань і її ретельній відладці.

Доведення ЕС від початку розробки до стадії промислової системи із застосуванням розвинених інструментальних засобів вимагає не менше 12-18 місяців;

- *комерційна система* — придатна не тільки для використання розробником, але і для продажу різним споживачам. Доведення системи до комерційної стадії вимагає приблизно 1,5-2 роки. Приведені тут терміни справедливі для ЕС середньої складності.

**Масштаб ЕС.** Характеризує складність вирішуваних задач і пов'язаний з типом використовуваної ЕОМ. По цій ознаці розрізняють:

- *малі ЕС* — призначені для первинного навчання і дослідження можливості застосування технології ЕС для даного класу задач. Системи такого типу можуть бути реалізовані на персональних комп'ютерах;

- *середні ЕС* - охоплюють весь спектр необхідних додатків і звичайно інтегровані з БД, електронними таблицями і т.д. Системи такого масштабу найчастіше реалізуються на робочих станціях;

- *великі ЕС* - мають доступ до великих БД і реалізуються на робочих станціях або на спеціалізованих комп'ютерах;

- *символьні ЕС* — створюються з дослідницькою метою і реалізуються на спеціалізованих комп'ютерах, орієнтованих на обробку символьних даних.

**Тип проблемного середовища** - Це поняття включає опис наочної області (безліч суті, що описує безліч об'єктів, їх характеристик і відносин між об'єктами) і вирішуваних в ній задач. Інакше кажучи, проблемне середовище включає структури даних і вирішувани з ними задачі, представлені у вигляді виконуваних тверджень (правив, процедур, формул і т. п.). У зв'язку з цим проблемне середовище визначається характеристиками відповідної наочної області і характеристиками типів вирішуваних в ній задач.

#### **Характеристики наочної області**

1. Тип наочної області:

- *статичний* - вхідні дані не змінюються за час сеансу роботи додатку, значення інших (не вхідних) даних змінюються тільки самою експертною системою;

- *динамічний* - вхідні дані, що поступають із зовнішніх джерел, змінюються в часі, значення інших даних змінюються ЕС або підсистемою моделювання зовнішнього оточення.

2. Спосіб опису суті наочної області:

- сукупність атрибутів і їх значень (фіксований склад суті);

- сукупність класів (об'єктів) і їх екземплярів (змінний склад суті).

3. Спосіб організації суті в БЗ:

- неструктурована БЗ;

- структуризація суті в БЗ за різними ієрархіями («приватне - загальне», «частина - ціле», «рід - вигляд»), що забезпечує спадкоємство властивостей суті.

Структуризація БЗ сприяє:

- обмеженню круга суті, яка повинна розглядатися механізмом висновку, і скороченню кількості перебраних варіантів в процесі вибору рішення;



- забезпеченню *спадкоємства властивостей суті*, тобто передачі властивостей розташованих вище в ієрархії суті розташованим нижче, що значно спрощує процес придбання і використання знань.

### ***Характеристики задач***

#### 1. Тип вирішуваних задач:

- задачі аналізу або синтезу. У *задачі аналізу* задана модель суті і вимагається визначити невідомі характеристики моделі. У *задачі синтезу* задаються умови, яким повинні задовольняти характеристики «невідомої» моделі суті, і вимагається побудувати модель цієї суті. Рішення задачі синтезу звичайно включає задачу аналізу як складову частину;

- статичні або динамічні задачі. Якщо задачі, вирішувані ЕС, явно не враховують чинник часу та/або не змінюють в процесі свого рішення знання про навколишній світ, то говорять, що ЕС вирішує *статичні задачі*, інакше говорять про рішення *динамічних задач*. Враховуючи значущість часу в динамічних проблемних середовищах, багато фахівців називають їх додатками, що працюють у реальному часі. Звичайно виділяють наступні системи реального часу: псевдореального часу, «*м'якого*» реального часу і «*жорсткого*» реального часу. Системи псевдореального часу, як випливає з назви, не є системами реального часу, проте вони, на відміну від статичних систем, одержують і обробляють дані, що поступають із зовнішніх джерел. Системи псевдореального часу вирішують задачу швидше, ніж відбуваються значущі зміни інформації про навколишній світ.

#### 2. Спільність виконуваних тверджень:

- приватні виконувані твердження, що містять посилання на конкретну суть (об'єкти);

- загальні виконувані твердження, що відносяться до будь-якої суті заданого типу (незалежно від їх числа і імені). Використовування загальних тверджень дозволяє значно лаконічне представляти знання. Проте оскільки загальні твердження не містять явних посилань на конкретну суть, для їх використання кожного разу вимагається визначати ту суть, до яких вони повинні застосовуватися.

Не всі поєднання перерахованих вище параметрів, що характеризують проблемне середовище, зустрічаються на практиці. Найбільш поширені наступні типи проблемних середовищ:

- статична наочна область:

представлення суті у вигляді сукупності атрибутів і їх значень, незмінний склад суті, БЗ не структурована, розв'язуються статичні задачі аналізу, використовуються тільки приватні виконувані твердження;

представлення суті об'єктами, змінний склад суті, БЗ структурована, розв'язуються статичні задачі аналізу і синтезу, використовуються загальні і приватні виконувані твердження;

- динамічна наочна область:

– уявлення, суті сукупністю атрибутів, і їх значень, незмінний склад суті, БЗ не структурована, розв'язуються динамічні задачі аналізу, використовуються приватні виконувані твердження;

– представлення суті у вигляді об'єктів, змінний склад суті, БЗ структурована, розв'язуються динамічні задачі аналізу і синтезу, використовуються загальні і приватні виконувани твердження.

**Тип вирішуваної задачі.** По цій ознаці розрізняють наступні задачі:

- *інтерпретація даних* — процес визначення значення даних, результати якого повинні бути узгодженими і коректними. Експертні системи, як правило, проводять багатоваріантний аналіз даних;

- *діагностика* — процес співвідношення об'єкту з деяким класом об'єктів так/або виявлення несправностей в системі (відхилень параметрів системи від нормативних значень);

- *моніторинг* - безперервна інтерпретація даних в реальному масштабі часу і сигналізація про вихід тих або інших параметрів за допустимі межі;

- *проективання* - створення раніше не існуючого об'єкту і підготовка специфікацій на створення об'єктів з наперед певними властивостями. Ступінь новизни може бути різним і визначається видом знань, закладених в ЕС, і методами їх обробки. Для організації ефективного проектування і реінжинірінга вимагається формувати не тільки самі проектні рішення, але і мотиви їх ухвалення. ЕС, вирішальні задачі проектування, реалізують процедури висновку рішення і пояснення одержаних результатів;

- *прогнозування* — прогноз наслідків деяких подій або явищ на основі аналізу наявних даних. Прогнозуючі ЕС логічно виводять вірогідні слідства із заданих ситуацій. У прогнозуючих ЕС в більшості випадків використовуються динамічні моделі, в яких значення параметрів «підганяються» під задану ситуацію. Слідства, що виводяться з цих моделей, складають основу для прогнозів з оцінками вірогідності;

- *планування* — побудова планів дій об'єктів, здатних виконувати деякі функції. Робота ЕС по плануванню заснована на моделях поведінки реальних об'єктів, які дозволяють проводити логічний висновок наслідків планованої діяльності;

- *навчання* — використання комп'ютера для навчання яким-небудь дисципліні або предмету. Експертні системи навчання виконують такі функції, як діагностика помилок, підказування правильних рішень; акумуляція знань про гіпотетичного «учня» і його характерні помилки; діагностика слабкості в пізнаннях навчених і знаходження відповідних засобів для їх ліквідації. Системи навчання здатні планувати акт спілкування з учнем залежно від успіхів учня для передачі необхідних знань;

- *управління* — функція організованої системи, підтримуюча певний режим її діяльності. Експертні системи даного типу призначені для управління поведінкою складних систем відповідно до заданих специфікацій;

- *підтримка ухвалення рішень* - сукупність процедур, що забезпечує особу, що ухвалює рішення, необхідною інформацією і рекомендаціями, що полегшують процес ухвалення рішення. Такого роду ЕС надають допомога фахівцям у виборі та/або генерації найраціональнішої альтернативи з безлічі можливих при ухваленні відповідальних рішень.

Задачі інтерпретації даних, діагностики, підтримка ухвалення рішень відносяться до задач аналізу, задачі проектування, планування і управління - до задач синтезу. До комбінованого типу задач відносяться навчання, моніторинг і прогнозування.

### 11.2.2. Структура ЕС

При розробці експертної системи прийнято поділяти її на три основних модулі (рис. 11.1):

1. база знань;
2. машина логічного виводу;
3. інтерфейс із користувачем.

*База знань* містить знання, що відносяться до конкретної прикладної області, у тому числі окремі факти, правила, що описують чи відносини явища, а також, можливо, методи, евристики і різні ідеї, що відносяться до рішення задач у цій прикладній області (мета правила).

*Машина логічного виводу* вміє активно використовувати інформацію, що міститься в базі знань.

*Інтелектуальний інтерфейс* із користувачем відповідає за безперебійний обмін інформацією між користувачем і системою; він також дає користувачу можливість спостерігати за процесом рішення задач, що протікають у машині логічного висновку.

Прийнято розглядати машину виводу й інтерфейс як один великий модуль, звичайно названий оболонкою експертної системи, чи, для стислості, просте оболонкою.

Основу кваліфікації експерта, окрім формалізованих знань, складають знання, що важко формалізувати, а саме: евристичні прийоми, припущення, інтуїтивні думання і уміння роботі висновки, які сам експерт може не цілком усвідомлювати. Тому створення експертних систем – тривалій і складній процес. Заповнення бази знань – найбільш трудомісткий етап. Розробку структури і наповнення бази знань здійснює інженер за уявленням знань. Він працює в тісному зв'язку з одним або декількома експертами. Разом сморід докладним чином визначають, якими повинні бути правила висновку, як смороду взаємозв'язані між собою, по можливості усувають суперечності в думаннях експертів, знаходять аналогії, виробляють мета правила (тобто правила, що описують, яким чином інші правила повинні бути використані або модифіковані), будують розгалужені мережі логічних висновків і багато що інше.

### 11.2.3. Технологія проектування і розробки експертних систем

Промислова технологія створення інтелектуальних систем включає наступні етапи:

- дослідження здійснимості проекту;
- розробку загальної концепції системи;
- розробку і тестування серії прототипів;
- розробку і випробування головного зразка;

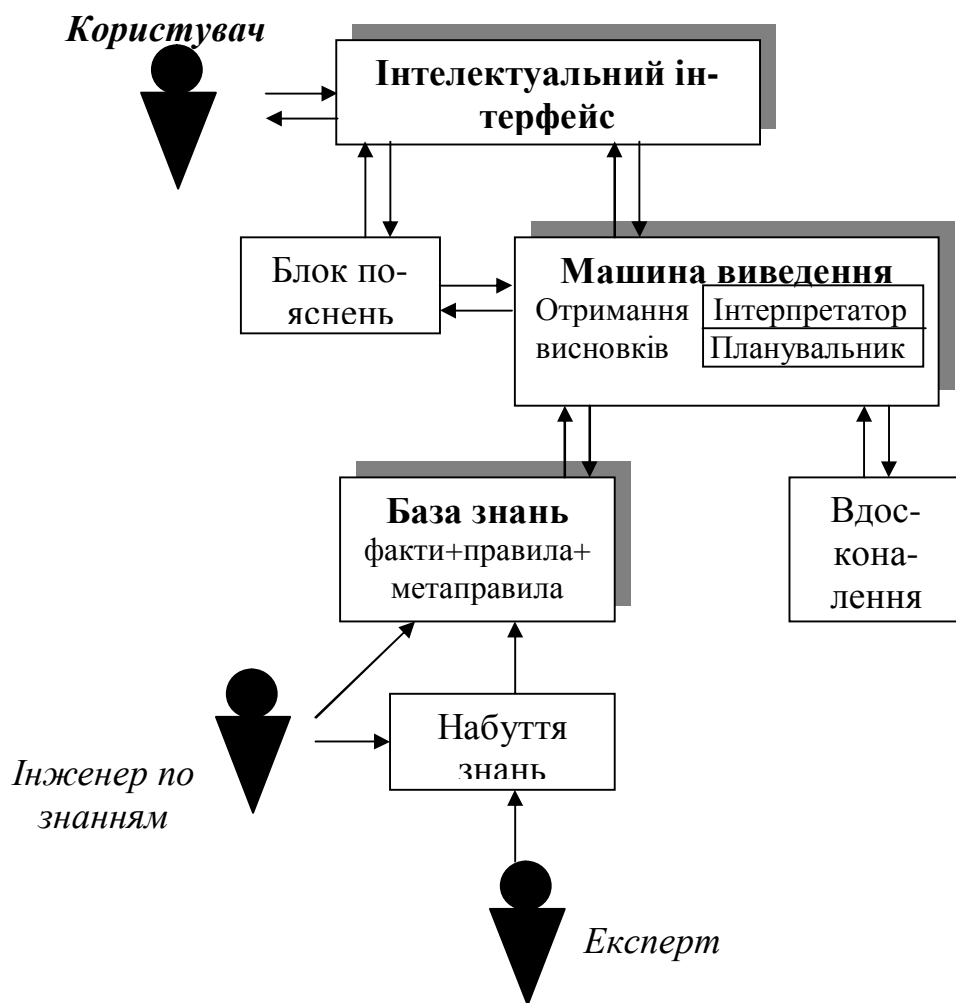


Рис. 11.1. Типова структура ЕС

- розробку і перевірку розширених версій системи;
- прив'язку системи до реального робочого середовища.

Проектування ЕС засноване на трьох головних принципах:

1. Потужність експертної системи обумовлена перш за все потужністю БЗ і можливостями її поповнення і тільки тоді - використовуваними методами (процедурами) обробки інформації.

2. Знання, що дозволяють експерту (або експертній системі) одержати якісні і ефективні рішення задач, є в основному евристичними, емпіричними, невизначеними, правдоподібними.

3. Неформальний характер вирішуваних задач і використовуваних знань робить необхідним забезпечення активного діалогу користувача з ЕС в процесі її роботи.

Перш ніж приступити до розробки ЕС, інженер по знанням повинен розглянути питання, чи слід розробляти ЕС для даного додатку. Позитивне рішення ухвалюється тоді, коли розробка ЕС *можлива, оправдані* і методи інженерії знань *відповідають* вирішуваній задачі.

Щоб розробка ЕС була *можливою* для даного додатку, необхідне виконання, принаймні, наступних вимог:

- існують експерти в даній області, які вирішують задачу значно краще, ніж фахівці, що починають;
- експерти сходяться в оцінці пропонованого рішення, оскільки інакше буде неможливо оцінити якість розробленої ЕС;
- експерти здатні вербалізувати (виразити на природній мові) і пояснити використовувані ними методи, інакше важко розраховувати на те, що знання експертів будуть «витагнуті» і закладені в ЕС;
- рішення задачі вимагає тільки міркувань, а не дій;
- задача не повинна бути дуже важкою (тобто її рішення повинне займати у експерта декілька годин або днів, а не тижнів або років);
- задача хоча і не повинна бути виражена у формальному вигляді, але все таки повинна відноситися до достатньо «зрозумілої» і структурованої області, тобто повинна існувати можливість виділення основних понять, відносин і способів отримання рішення задачі;
- рішення задачі не повинне в значній мірі спиратися на «здоровий глузд» (тобто широкий спектр загальних відомостей про світ і про спосіб його функціонування, які знає і уміє використовувати будь-яка нормальна людина), оскільки подібні знання поки не вдасться в достатній кількості закласти в системи штучного інтелекту.

Додаток *відповідає* методам ЕС, якщо вирішувана задача володіє сукупністю наступних характеристик:

- задача може бути природним чином вирішена за допомогою маніпулювання символами (за допомогою символічних міркувань), а не маніпулювання числами, як прийнято в математичних методах і в традиційному: програмуванні;
- задача повинна мати евристичну, а не алгоритмічну природу, тобто її рішення повинне вимагати застосування евристичних правил. Для задач, які можуть бути гарантований вирішені (при дотриманні заданих обмежень) за допомогою формальних процедур, існують ефективніші підходи, ніж технології ЕС.

При розробці ЕС, як правило, використовується концепція *швидкого прототипу*, суть якої полягає у тому, що розробники не намагаються відразу побудувати кінцевий продукт. На початковому етапі вони створюють прототип (можливо, не єдиний) ЕС, що задовольняє двом суперечливим вимогам:

уміння вирішувати типові задачі конкретного додатку і незначні час і трудомісткість його розробки. При виконанні цих умов стає можливим паралельно вести процес накопичення і відладки знань, здійснюваний експертом, і процес вибору (розробки) програмних засобів, виконуваний інженером по знаннях і програмістами. Для задоволення вказаним вимогам при створенні прототипу використовуються різноманітні інструментальні засоби, які прискорюють процес проектування.

Традиційна технологія реалізації ЕС включає шість основних етапів: ідентифікацію, концептуалізацію, формалізацію, виконання, тестування, досвідчену експлуатацію.

На етапі *ідентифікації* визначаються задачі, що підлягають рішенню, цілі розробки, експерти і типи користувачів.

На етапі *концептуалізації* проводиться змістовний аналіз проблемної області, виявляються використовувані поняття і їх взаємозв'язки, визначаються методи рішення задач.

На етапі *формалізації* вибираються інструментальні засоби і способи представлення всіх видів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність системи зафіксованих понять, методів рішення, засобів уявлення і маніпулювання знаннями даної наочної області.

На етапі *виконання* здійснюється заповнення бази знань. У зв'язку з тим, що основою ЕС є знання, даний етап є одним з найважливіших і самих трудомістких. Процес придбання знань розділяють на витягання знань в діалозі з експертами; організацію знань, яка забезпечує ефективну роботу системи, і представлення знань у вигляді, «зрозумілому» ЕС. Процес придбання знань здійснюється інженером по знаннях на основі аналізу діяльності експерта за рішенням реальних задач.

На етапі *тестування* експерт і інженер по знаннях в інтерактивному режимі з використанням діалогових і пояснювальних засобів перевіряють компетентність ЕС. Процес тестування продовжується до тих пір, поки експерт не вирішить, що система досягла необхідного рівня компетентності.

На етапі *досвідченої експлуатації* перевіряється придатність ЕС для кінцевих користувачів. Одержані результати можуть показати необхідність істотної модифікації ЕС.

Процес створення ЕС не зводиться до строгої послідовності перерахованих вище етапів. В ході розробки доводиться неодноразово повертатися на ранішні етапи і переглядати ухвалені там рішення.

Інструментальні засоби розрізняються залежно від того, яку технологію розробки ЕС вони допускають. Можна виділити, принаймні, чотири підходи до розробки ЕС:

- підхід, що базується на поверхневих знаннях;
- структурний підхід;
- підхід, заснований на глибинних знаннях;
- змішаний підхід, що спирається на використання поверхневих і глибинних знань.

*Поверхневий* підхід застосовується для складних задач, які не можуть бути точно описані. Його суть полягає в отриманні від експертів фрагментів знань, релевантних вирішуваній задачі. При цьому не робиться спроб систематичного або глибинного вивчення області, що зумовлює використання пошуку в просторі станів як універсальний механізм висновку. Звично в ЕС, які використовують даний підхід, як спосіб представлення знань вибираються правила. Умова кожного правила визначає зразок деякої ситуації, в якій правило може бути виконане. Пошук рішення полягає у виконанні тих правил, зразки яких зіставляються з поточними даними. При цьому передбачається, що в процесі пошуку рішення послідовність формованих таким чином ситуацій не обірветься

до отримання рішення, тобто не виникне невідомої ситуації, яка не відповідає жодному правилу. Даний підхід з успіхом застосовується до широкого класу додатків, але виявляється неефективним в тих випадках, коли задача може структуруватися або для її вирішення може використовуватися деяка модель.

*Структурний підхід* до побудови ЕС передбачає структурування знань проблемної області. Його поява обумовлена тим, що для ряду додатків застосування техніки поверхневих знань не забезпечує рішення задачі. Структурний підхід до побудови ЕС багато в чому схожий на структурне програмування. Проте стосовно ЕС йдеться не про те, щоб структурування задачі була доведена до точного алгоритму (як в традиційному програмуванні), а передбачається, що частина задачі розв'язується за допомогою евристичного пошуку. Структурний підхід в різних додатках доцільно поєднувати з поверхневим або глибинним.

При *глибинному підході* компетентність ЕС базується на моделі того проблемного середовища, в якій вона працює; Модель може бути визначена різними способами (декларативно, процедурно). Необхідність використання моделей у ряді додатків викликана прагненням виправити недолік поверхневого підходу, пов'язаний з виникненням ситуацій, не описаних правилами, що зберігаються в БЗ. Експертні системи, розроблені із застосуванням глибинних знань, при виникненні невідомої ситуації здатні самостійно визначити, які дії слід виконати, за допомогою деяких загальних принципів, справедливих для даної області експертизи.

Глибинний підхід вимагає явного опису структури і взаємостосунків між різною суттю проблемної області. У цьому підході необхідно використовувати інструментальні засоби, що володіють можливостями моделювання:

об'єкти з приєднаними процедурами, ієрархічне спадкоємство властивостей, активні знання (програмування, кероване даними), механізм передачі повідомлень об'єктам (об'єктно-орієнтоване програмування) і т.п.

*Змішаний підхід* в загальному випадку може поєднувати поверхневий, структурний і глибинний підходи. Наприклад, поверхневий підхід може застосовуватися для пошуку адекватних знань, які потім використовуються деякою глибинною моделлю.

### **Контрольні запитання**

1. Що таке „вербалізація методики”?
2. Дайте ознаки відмінності інтелектуальної інформаційної системи від інших.
3. Дайте поняття когнітивної графіки.
4. Що таке «інженерія знань»?
5. Які інформаційні системи називаються мультиагентами?
6. Які принципи роботи самонавчальних інформаційних систем?
7. Як працює адаптивна інформаційна система?

*У розділі подано поняття про класифікацію інтелектуальних інформаційних систем та технологію розробки експертних систем*

# 12. ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

*Вивчивши матеріал цього розділу, студент узнає про історію, розвиток та перспективи інформаційних технологій.*

## 12.1. Що таке інформаційна технологія

Технологія - це комплекс наукових та інженерних знань, реалізованих у прийомах праці, наборах матеріальних, технічних, енергетичних, трудових факторів виробництва, засобах їх об'єднання для створення продукту або послуги, що відповідають певним вимогам. Тому технологія нерозривно пов'язана з машинізацією виробничого або невиробничого, насамперед управлінського процесу. Управлінські технології ґрунтуються на застосуванні комп'ютерів і телекомунікаційної техніки.

Відповідно до визначення, прийнятого ЮНЕСКО, інформаційна технологія - це комплекс взаємозалежних, наукових, технологічних, інженерних дисциплін, що вивчають методи ефективної організації праці людей, зайнятих опрацюванням і збереженням інформації; обчислювальну техніку і методи організації і взаємодії з людьми і виробничим устаткуванням, практичні додатки, а також пов'язані з усім цим соціальні, економічні і культурні проблеми. Самі інформаційні технології вимагають складної підготовки, великих початкових витрат і наукомісткої техніки. Їхнє введення повинно починатися зі створення математичного забезпечення, формування інформаційних потоків у системах підготовки спеціалістів.

## 12.2 Етапи розвитку інформаційних технологій

Існує декілька точок зору щодо розвитку інформаційних технологій із використанням комп'ютерів, що визначаються різноманітними ознаками поділу.

Загальним для усіх викладених нижче підходів є те, що з появою персонального комп'ютера почався новий етап розвитку інформаційної технології. Основною ціллю стає задоволення персональних інформаційних потреб людини як для фахової сфери, так і для побутової.

Ознака поділу - вид задач і процесів опрацювання інформації:

1-й етап (60 - 70-і рр.) - опрацювання даних в обчислювальних центрах у режимі колективного користування. Основним напрямком розвитку інформаційної технології була автоматизація операційних рутинних дій людини.

2-й етап (з початку 80-х рр.) - створення інформаційних технологій, спрямованих на розв'язання стратегічних задач.



Ознака поділу - проблеми, які стоять на шляху інформатизації:

1-й етап (до кінця 60-х рр.) характеризується проблемою опрацювання великих обсягів даних в умовах обмежених можливостей апаратних засобів.

2-й етап (до кінця 70-х рр.) пов'язаний з поширенням ЕОМ серії ІВМ/360. Проблема цього етапу - відставання програмного забезпечення від рівня розвитку апаратних засобів.

3-й етап (з початку 80-х рр.) - комп'ютер стає інструментом непрофесійного користувача, а інформаційні системи - засобом підтримки прийняття його рішень. Проблемами цього етапу є максимальне задоволення потреб користувача і створення відповідного інтерфейсу для роботи в комп'ютерному середовищі.

4-й етап (з початку 90-х рр.) - створення сучасної технології міжустановних зв'язків і інформаційних систем. Проблеми цього етапу дуже багаточислені. Найбільше суттєвими з них є:

- укладання угод і встановлення стандартів, протоколів для комп'ютерного зв'язку;
- організація доступу до стратегічної інформації;
- організація захисту і безпеки інформації.

Ознака поділу - перевага, яку надає комп'ютерна технологія:

1-й етап (з початку 60-х рр.) характеризується досить ефективним опрацюванням інформації при виконанні рутинних операцій з орієнтацією на централізоване колективне використання ресурсів обчислювальних центрів. Основним критерієм оцінки ефективності інформаційних систем, які створювались, була різниця між витраченими на розробку і зекономленими в результаті впровадження коштами. Основною проблемою на цьому етапі була психологічна - погана взаємодія користувачів, для яких створювалися інформаційні системи, і розроблювачів через розходження їхніх поглядів і розуміння розв'язуваних проблем. Як наслідок цієї проблеми, створювалися системи, які користувачі погано сприймали і, незважаючи на їх достатньо великі можливості, не використовували повною мірою їх потенціал.

2-й етап (з середини 70-х рр.) пов'язаний з появою персональних комп'ютерів. Змінився підхід до створення інформаційних систем - орієнтація зміщується у бік індивідуального користувача для підтримки прийнятих ним рішень. Користувач зацікавлений у проведеній розробці, налагоджується контакт із розроблювачем, виникає порозуміння між обома групами спеціалістів. На цьому етапі використовується як централізоване опрацювання даних, характерне для першого етапу, так і децентралізоване, що базується на розв'язанні локальних задач і роботі з локальними базами даних на робочому місці користувача.

3-й етап (з початку 90-х рр.) пов'язаний з поняттям аналізу стратегічних переваг у бізнесі і заснований на досягненнях телекомунікаційної технології розподіленого опрацювання інформації. Інформаційні системи мають своєю метою не просто збільшення ефективності опрацювання даних і допомога керівнику. Відповідні інформаційні технології повинні допомогти організації вистояти в конкурентній боротьбі й одержати перевагу.

Ознака поділу - види інструментарію інформаційної технології:

1-й етап (до другої половини XIX ст.) - "ручна" інформаційна технологія інструментарій якої складала: ручка, чорнильниця, книга. Комунікації здійснювалися ручним способом шляхом надсилання через пошту листів, пакетів, депеш. Основною метою інформаційної технології цього періоду було представлення інформації в потрібній формі.

2-й етап (з кінця XIX ст.) - "механічна" технологія, інструментарій якої складала: друкарська машинка, телефон, диктофон, оснащена більш досконалими засобами доставки пошта. Основна мета технології - представлення інформації в потрібній формі більш зручними засобами.

3-й етап (40-60-і рр. XX ст.) - "електрична" технологія, інструментарій якої складала: великі ЕОМ і відповідне програмне забезпечення, електричні друкарські машинки, ксерокси, портативні диктофони. На цьому етапі відбувається зміна мети технології. Акцент в інформаційній технології починає зміщуватись з форми представлення інформації на формування її змісту.

4-й етап (з початку 70-х рр.) - "електронна" технологія, основним інструментарієм якої стають великі ЕОМ і створені на їхній базі автоматизовані системи керування (АСК) і інформаційно-пошукові системи (ІПС), оснащені широким спектром базових і спеціалізованих програмних комплексів. Центр ваги технології ще більш зміщується на формування змістовної сторони інформації для управлінського середовища різноманітних сфер громадського життя, особливо на організацію аналітичної роботи. Безліч об'єктивних і суб'єктивних факторів не дозволили вирішити поставлені перед новою концепцією інформаційної технології задачі. Проте був здобутий досвід формування змістовної сторони управлінської інформації і підготовлена фахова, психологічна і соціальна база для переходу на новий етап розвитку технології.

5-й етап (з середини 80-х рр.) - "комп'ютерна" ("нова") технологія, основним інструментарієм якої є персональний комп'ютер із широким спектром стандартних програмних продуктів різного призначення. На цьому етапі відбувається процес персоналізації АСК, що проявляється у створенні систем підтримки прийняття рішень певними спеціалістами. Подібні системи мають умонтовані елементи аналізу та інтелекту для різних рівнів керування, вони реалізуються на персональному комп'ютері і використовують телекомунікації. У зв'язку з переходом на мікропроцесорну базу суттєвим змінам піддаються і технічні засоби побутового, культурного та інших призначень. Починають широко використовуватися в різноманітних галузях глобальні і локальні комп'ютерні мережі.

Для створення більш цілісного представлення про цей період доцільно познайомитися з приведеною нижче таблицею про зміну поколінь електронно-обчислювальних машин (ЕОМ) і зіставити ці дані з етапами в області обробки і передачі інформації.

*Зміна поколінь ЕОМ*

<b>Покоління</b>	<b>Елементна база</b>	<b>Характеристика ЕОМ</b>
<b>1-і покоління</b> (початок 50-х рр.)	електронні лампи	ЕОМ відрізнялися великими габаритами, великим споживанням енергії, малою швидкістю, низкою надійністю, програмуванням у кодах

<b>Покоління</b>	<b>Елементна база</b>	<b>Характеристика ЕОМ</b>
<b>2-і покоління</b> (з кінця 50-х рр. )	напівпровідникові елементи	Покращилися в порівнянні з ЕОМ попереднього покоління всі технічні характеристики. Для програмування використовуються алгоритмічні мови
<b>3-і покоління</b> (початок 60-х рр.)	інтегральні схеми, багатошаровий друкований монтаж	Різде зниження габаритів ЕОМ, підвищення їхньої надійності, збільшення продуктивності. Доступ з віддалених терміналів
<b>4-і покоління</b> (із середини 70-х рр.)	мікропроцесори, великі інтегральні схеми	Покращилися технічні характеристики. Масовий випуск персональних комп'ютерів. Напрямки розвитку: могутні багатопроцесорні обчислювальні системи з високою виробничістю, створення дешевих мікро-ЕОМ
<b>5-і покоління</b> (із середини 80-х рр.)	Почалася розробка інтелектуальних комп'ютерів, поки не увінчалася успіхом	Упровадження в усі сфери комп'ютерних мереж і їхнє об'єднання, використання розподіленої обробки даних, повсюдне застосування комп'ютерних інформаційних технологій

### **12.3 Складові частини інформаційної технології**

Такі технологічні поняття, що використовуються у виробничій сфері, як норма, норматив, технологічний процес, технологічна операція і т.п., можуть застосовуватися і в інформаційній технології. Перед тим, як розробляти ці поняття в будь-якій технології, у тому числі й в інформаційній, завжди варто починати з визначення мети. Потім варто спробувати провести структурування всіх дій, що призводять до наміченої мети, і вибрати необхідний програмний інструментарій.

Необхідно розуміти, що освоєння інформаційної технології і подальше її використання повинні бути зведені до того, щоб було потрібно спочатку добре оволодіти набором елементарних операцій, кількість яких обмежена. З цієї обмеженої кількості елементарних операцій у різних комбінаціях складається дія, а з дій, також у різних комбінаціях, складаються операції, що визначають той або інший технологічний етап. Сукупність технологічних етапів утворює технологічний процес (технологію). Він може починатися з будь-якого рівня і не включати, наприклад, етапи або операції, а складатися тільки з дій. Для реалізації етапів технологічного процесу можуть використовуватися різні програмні середовища.

Інформаційна технологія, як і будь-яка інша, повинна відповідати таким вимогам:

- забезпечувати високий рівень розчленування всього процесу опрацювання інформації на етапи (фази), операції, дії;
- включати весь набір елементів, необхідних для досягнення поставленої мети;
- мати регулярний характер. Етапи, дії, операції технологічного процесу можуть бути стандартизовані й уніфіковані, що дозволить більш ефективно здійснювати цілеспрямоване керування інформаційними процесами.

## 12.4 Інструментарій інформаційної технології

Реалізація технологічного процесу матеріального виробництва здійснюється за допомогою різноманітних технічних засобів, до яких відносяться: устаткування, верстати, інструменти, конвеєрні лінії і т.п.

За аналогією і для інформаційної технології повинно бути щось подібне. Такими технічними засобами виробництва інформації будуть апаратне, програмне і математичне забезпечення цього процесу. З їхньою допомогою відбувається переробка первинної інформації в інформацію нової якості. Виділимо окремо з цих засобів програмні продукти і назовемо їх інструментарієм, а для більшої чіткості можна його конкретизувати, назвавши програмним інструментарієм інформаційної технології.

Визначимо це поняття:

Інструментарій інформаційної технології - один або декілька взаємозалежних програмних продуктів для певного типу комп'ютера, технологія роботи в яких дозволяє досягти поставленої користувачем мети.

У якості інструментарію можна використовувати такі поширені види програмних продуктів для персонального комп'ютера як текстовий процесор (редактор), настільні видавничі системи, електронні таблиці, системи керування базами даних, електронні записні книжки, електронні календарі, інформаційні системи функціонального призначення (фінансові, бухгалтерські, для маркетингу та ін.), експертні системи і т.д.

Існує кілька поглядів на розвиток інформаційних технологій з використанням комп'ютерів, що визначаються різними ознаками розподілу. Загальним для усіх викладених нижче підходів є те, що з появою персонального комп'ютера почалася новий етап розвитку інформаційної технології. Основною метою стає задоволення персональних інформаційних потреб людини як для професійної сфери, так і для побутової.

<b>Ознака розподілу</b>	<b>Характеристика етапів</b>
Вид задач і процесів обробки інформації.	1 - й етап (60 70-і рр.) обробка даних в обчислювальних центрах у режимі колективного користування. Основним напрямком розвитку інформаційної технології була автоматизація операційних рутинних дій людини. 2- й етап (з 80-х рр.) створення інформаційних технологій, спрямованих на вирішення стратегічних задач.

<b>Ознака розподілу</b>	<b>Характеристика етапів</b>
Проблеми, що коштують на шляху інформатизації суспільства	<p>1-й етап (до кінця 60-х рр.) характеризується проблемою обробки великих обсягів даних в умовах обмежених можливостей апаратних засобів.</p> <p>2 - й етап (до кінця 70-х рр.) зв'язується з поширенням ЕОМ серії ІВМ/360. Проблема цього етапу відставання програмного забезпечення від рівня розвитку апаратних засобів.</p> <p>3 - й етап (з початку 80-х рр.) комп'ютер стає інструментом непрофесійного користувача, а інформаційні системи засобом підтримки прийняття його рішень. Проблеми максимальне задоволення потреб користувача і створення відповідного інтерфейсу роботи в комп'ютерному середовищі.</p> <p>4 - й етап ( з початку 90-х рр.) створення сучасної технології міжорганізаційних зв'язків і інформаційних систем. Проблеми цього етапу досить численні. Найбільш істотними з них є:</p> <ul style="list-style-type: none"> <li>• вироблення угод і встановлення стандартів, протоколів для комп'ютерного зв'язку;</li> <li>• організація доступу до стратегічної інформації; о організація захисту і безпеки інформації.</li> </ul>
Перевага, що приносить комп'ютерна технологія	<p>1- й етап (з початку 60-х рр.) характеризується досить ефективною обробкою інформації при виконанні рутинних операцій з орієнтацією на централізоване колективне використання ресурсів обчислювальних центрів. Основним критерієм оцінки ефективності створюваних інформаційних систем була різниця між витраченими на розробку і зекономленими в результаті впровадження засобами. Основною проблемою на цьому етапі була психологічна погана взаємодія користувачів, для яких створювалися інформаційні системи, і розроблювачів через розходження їхніх поглядів і розуміння розв'язуваних проблем. Як наслідок цієї проблеми, створювалися системи, що користувачі погано сприймали і, незважаючи на їхній досить великі можливості, не використовували повною мірою.</p> <p>2-й етап (із середини 70-х рр.) зв'язаний з появою персональних комп'ютерів. Змінився підхід до створення інформаційних систем орієнтація зміщається убік індивідуального користувача для підтримки прийнятих їм рішень. Користувач зацікавлений у проведеній розробці, налагоджується контакт із розроблювачем, виникає взаєморозуміння обох груп фахівців. На цьому етапі використовується як централізована обробка даних, характерна для першого етапу, так і децентралізована, що базується на рішенні локальних задач і роботі з локальними базами даних на робочому місці користувача.</p> <p>3- й етап ( з початку 90-х рр.) зв'язаний з поняттям аналізу</p>

<b>Ознака розподілу</b>	<b>Характеристика етапів</b>
	<p>стратегічних переваг у бізнесі і заснований на досягненнях телекомунікаційної технології розподіленої обробки інформації. Інформаційні системи мають своєю метою не просте збільшення ефективності обробки даних і допомогу керівникові. Відповідні інформаційні технології повинні допомогти організації вистояти в конкурентній боротьбі й отримати перевагу.</p>
<p>Види інструментарію технології</p>	<p>1 - й етап (до другої половини ХІХ в.) "ручна" інформаційна технологія, інструментарій якої складали: перо, чорнильниця, книга. Комунікації здійснювалися ручним способом шляхом переправлення через пошту листів, пакетів, депеш. Основна мета технології представлення інформації в потрібній формі.</p> <p>2- й етап (з кінця ХІХ в.) "механічна" технологія, інструментарій якої складали: друкарська машинка, телефон, диктофон, оснащена більш зробленими засобами доставки пошта. Основна мета технології представлення інформації в потрібній формі більш зручними засобами.</p> <p>3 - й етап (40 60-і рр. ХХ в.) "електрична" технологія, інструментарій якої складали: великі ЕОМ і відповідне програмне забезпечення, електричні друкарські машинки, ксерокси, портативні диктофони. Змінюється мета технології. Акцент в інформаційній технології починає переміщатися з форми представлення інформації на формування її змісту.</p> <p>4- й етап (з початку 70-х рр.) "електронна" технологія, основним інструментарієм якої стають великі ЕОМ і створювані на їхній базі автоматизовані системи керування (АСУ) і пошуковий - пошукові-інформаційно-пошукові системи (ІПС), оснащені широким спектром базових і спеціалізованих програмних комплексів. Центр ваги технології ще більш зміщається на формування змістовної сторони інформації для управлінського середовища різних сфер громадського життя, особливо на організацію аналітичної роботи. Безліч об'єктивних і суб'єктивних факторів не дозволили вирішити, що коштують перед новою концепцією інформаційної технології поставлені задачі. Однак був придбаний досвід формування змістовної сторони управлінської інформації і підготовлена професійна, психологічна і соціальна база для переходу на новий етап розвитку технології.</p> <p>5 - й етап (із середини 80-х рр.) "комп'ютерна" ("нова") технологія, основним інструментарієм якої є персональний комп'ютер із широким спектром стандартних програмних продуктів різного призначення. На цьому етапі відбувається процес персоналізації АСУ, що виявляється в створенні систем підтримки прийняття рішень визначеними фахівцями. Подібні</p>

Ознака розподілу	Характеристика етапів
	системи мають убудовані елементи аналізу й інтелекту для різних рівнів керування, реалізуються на персональному комп'ютері і використовують телекомунікації. У зв'язку з переходом на мікропроцесорну базу істотним змінам піддаються і технічні засоби побутового, культурного й іншого призначень. Починають широко використовуватися в різних областях глобальні і локальні комп'ютерні мережі.

### 12.5. Інформаційна технологія опрацювання даних

Інформаційна технологія опрацювання даних використовується для розв'язання добре структурованих задач, стосовно яких є необхідні вхідні дані і відомі алгоритми та інші стандартні процедури їх опрацювання. Ця технологія застосовується на рівні операційної (виконавчої) діяльності персоналу невисокої кваліфікації з метою автоматизації деяких рутинних постійно повторюваних операцій управлінської праці. Тому впровадження інформаційних технологій і систем на цьому рівні істотно підвищить продуктивність праці персоналу, звільнить його від рутинних операцій, можливо, навіть призведе до необхідності скорочення чисельності працівників.

На рівні операційної діяльності вирішуються такі задачі:

- опрацювання даних про операції, які здійснює фірма;
- створення періодичних контрольних звітів про стан справ у фірмі;
- одержання відповідей на всілякі поточні запити й оформлення їх у вигляді паперових документів або звітів.

Прикладом може послужити щоденний звіт про надходження і видачу готівки банком, який формується з метою контролю балансу готівки; або ж запит до бази даних по кадрах, який дозволить одержати дані про вимоги, що висуваються до кандидатів на певну посаду.

Існує декілька особливостей, пов'язаних з опрацюванням даних, що відрізняють дану технологію від усіх інших:

- виконання необхідних фірмі задач по опрацюванню даних. Від кожної фірми закон вимагає наявності та збереження даних про свою діяльність, які можна використовувати як засіб забезпечення і підтримки контролю на фірмі. Тому в будь-якій фірмі обов'язково повинна бути інформаційна система опрацювання даних і розроблена відповідна інформаційна технологія;
- вирішення тільки добре структурованих задач, для яких можна розробити алгоритм;
- виконання стандартних процедур опрацювання. Існуючі стандарти визначають типові процедури опрацювання даних і регламентують їхнє дотримання організаціями усіх видів;
- виконання основного обсягу робіт в автоматичному режимі з мінімальною участю людини;

- використання деталізованих даних. Записи про діяльність фірми мають докладний характер, що допускає проведення ревізій. У процесі ревізії діяльність фірми перевіряється хронологічно від початку періоду до його кінця і від кінця до початку;

- акцент на хронологію подій;

- вимога мінімальної допомоги у вирішенні проблем з боку спеціалістів інших рівнів.

Багато даних на рівні операційної діяльності необхідно зберігати для наступного використання або на цьому ж рівні, або на іншому. Для їхнього збереження створюються бази даних.

Стосовно створення звітів (документів) слід зазначити, що у інформаційній технології опрацювання даних необхідно створювати документи для керівництва і працівників фірми, а також для зовнішніх партнерів. При цьому документи можуть створюватися як за вимогою у зв'язку з проведеною фірмою операцією, так і періодично наприкінці кожного місяця, кварталу або року.

## **12.6. Інформаційна технологія керування**

Метою інформаційної технології керування є задоволення інформаційних потреб усіх без винятку співробітників фірми, що мають справу з прийняттям рішень. Вона може бути корисна на будь-якому рівні керування.

Ця технологія орієнтована на роботу в середовищі інформаційної системи керування і використовується при більш поганій структурованості розв'язуваних задач, якщо їх порівнювати з задачами, які розв'язуються за допомогою інформаційної технології опрацювання даних.

Інформаційна технологія керування ідеально підходять для задоволення подібних інформаційних потреб працівників різноманітних функціональних підсистем (підрозділів) або рівнів керування фірмою. Інформація, що поставляється нею, містить відомості про минуле, дійсне і ймовірне майбутнє фірми. Ця інформація має вигляд регулярних або спеціальних управлінських звітів.

Для прийняття рішень на рівні управлінського контролю інформація повинна бути подана в агрегованому вигляді, так, щоб проглядалися тенденції зміни даних, причини відхилень, що виникли, і можливі варіанти вирішення проблеми. На цьому етапі розв'язуються такі задачі опрацювання даних:

- оцінка планованого стану об'єкта керування;
- оцінка відхилень від планованого стану;
- виявлення причин відхилень;
- аналіз можливих рішень і дій.

Інформаційна технологія керування спрямована на створення різноманітних видів звітів. Регулярні звіти створюються відповідно до встановленого графіка, що визначає час їхнього створення, наприклад місячний аналіз продажів компанії.



Спеціальні звіти створюються за вимогою керівників, або коли в компанії відбулося щось незаплановане. І ті, і інші види звітів можуть мати форму підсумкових, порівняльних і надзвичайних звітів.

У підсумкових звітах дані об'єднані в окремі групи, відсортовані і подані у вигляді проміжних і остаточних результатів по окремих полях.

Порівняльні звіти містять дані, отримані з різноманітних джерел або класифіковані по різноманітних ознаках і використовуються для порівняння.

Надзвичайні звіти містять дані виняткового (надзвичайного) характеру.

Використання звітів для підтримки керування є особливо ефективним при реалізації так званого керування по відхиленнях. Керування по відхиленнях припускає, що головним змістом одержуваних спеціалістом даних повинні бути відхилення стану господарської діяльності фірми від деяких встановлених стандартів (наприклад, від її запланованого стану). При використанні на фірмі принципів керування по відхиленнях до звітів, які створюються, висуваються такі вимоги:

- звіт необхідно створювати тільки тоді, коли відхилення відбулося;
- відомості у звіті повинні бути відсортовані за значенням критичного для даного відхилення показника;
- усі відхилення бажано показати разом, щоб спеціаліст міг уловити існуючий між ними зв'язок;
- у звіті необхідно показати, кількісне відхилення від норми.

Вхідна інформація надходить із систем операційного рівня. Вихідна інформація формується у вигляді управлінських звітів у зручному для ухвалення рішення вигляді. Зміст бази даних за допомогою відповідного програмного забезпечення перетворюється в періодичні і спеціальні звіти, що надходять до спеціалістів, які беруть участь у прийнятті рішень в організації. База даних, що використовується для одержання зазначеної інформації, повинна складатися з двох компонентів:

- 1) даних, що накопичуються на основі оцінки операцій, проведених фірмою;
- 2) планів, стандартів, бюджетів та інших нормативних документів, що визначають планований стан об'єкта керування (підрозділи фірми).

## **12.7. Інформаційна технологія підтримки прийняття рішень**

Ефективність і гнучкість інформаційної технології багато в чому залежать від характеристик інтерфейсу системи підтримки прийняття рішень. Інтерфейс визначають: мова користувача; мова повідомлень комп'ютера, що організує діалог на екрані дисплея; знання користувача.

Мова користувача - це ті дії, які користувач робить по відношенню до системи шляхом використання можливостей клавіатури; електронних олівців, що пишуть на екрані; джойстика; "миші"; голосових команд, тощо. Найбільш простою формою мови користувача є створення форм вхідних і вихідних документів. Отримавши вхідну форму (документ), користувач заповнює його необхідними даними і вводить у комп'ютер. Система підтримки прийняття рішень ро-

бити необхідний аналіз і видає результати у вигляді вихідного документа заданої форми.

Мова повідомлень - це те, що користувач бачить на екрані дисплея (символи, графіка, колір), дані, надруковані принтером, звукові вихідні сигнали і т.п. Важливим показником ефективності інтерфейсу, який використовується, є обрана форма діалогу між користувачем і системою. В даний час найбільш поширеним є такі форми діалогу: режим “запитання-відповідь”, командний режим, режим меню, режим заповнення пропусків у виразах, запропонованих комп'ютером. Кожна форма в залежності від типу задачі, індивідуальності користувача і рішення, яке приймається, може мати свої переваги і недоліки. Довгий час єдиною реалізацією мови повідомлень був надрукований або виведений на екран дисплея звіт або повідомлення. Тепер з'явилася нова можливість представлення вихідних даних - машинна графіка. Вона дає можливість створювати на екрані і папері кольорові графічні зображення в тривимірному вигляді. Використання машинної графіки, яка значно підвищує наочність і інтерпретованість вихідних даних, стає усе більш популярним в інформаційній технології підтримки прийняття рішень.

Знання користувача - це те, що користувач повинен знати, працюючи з системою. До них відносяться не тільки план дій, що знаходиться в голові у користувача, але і підручники, інструкції, довідкові дані, які видає комп'ютер.

Удосконалення інтерфейсу системи підтримки прийняття рішень визначається успіхами у розвитку кожного з трьох зазначених вище компонентів.

Інтерфейс повинен мати такі можливості:

- маніпулювати різноманітними формами діалогу, змінюючи їх у процесі ухвалення рішення за вибором користувача;
- передавати дані до системи різноманітними засобами;
- одержувати дані від різноманітних пристроїв системи в різних форматах;
- гнучко підтримувати (надавати допомогу за вимогою, підказувати) знання користувача.

## **12.8. Інформаційна технологія експертних систем**

Децентралізоване опрацювання інформації пов'язане з появою в 80-х рр. персональних комп'ютерів і розвитком засобів телекомунікацій. Вона дуже істотно потіснила попередню технологію, оскільки дає користувачу широкі можливості в роботі з інформацією і не обмежує його ініціатив.

Перевагами такої методології є:

- гнучкість структури, що забезпечує простір ініціативам користувача;
- посилення відповідальності нижчої ланки співробітників;
- зменшення потреби в користуванні центральним комп'ютером і відповідно контролі з боку обчислювального центру;
- більш повна реалізація творчого потенціалу користувача завдяки використанню засобів комп'ютерного зв'язку.

Проте ця методологія має і свої недоліки:

- складність стандартизації через велику кількість унікальних розробок;

- психологічне неприйняття користувачами що рекомендуються обчислювальним центром стандартів у готових програмні продукти;
- нерівномірність розвитку рівня інформаційної технології на локальних місцях, що в першу чергу визначається рівнем кваліфікації конкретного працівника.

Такий підхід назвемо раціональною методологією і покажемо, як у цьому випадку будуть розподілятися обов'язки:

- обчислювальний центр повинен відповідати за створення загальної стратегії використання інформаційної технології, допомагати користувачам як у роботі, так і у навчанні, установлювати стандарт і визначати політику застосування програмних і технічних засобів;
- персонал, який використовує інформаційну технологію, повинен дотримуватися вказівок обчислювального центру, здійснювати розробку своїх локальних систем і технологій відповідно до загального плану організації.

## **12.9. Небезпека та труднощі використання інформаційних технологій**

Складне програмне забезпечення має недоліки, якими можуть скористатися сторонні особи (хакери) і використати їх на свою користь. Так, наприклад, один хакер вкрав з кредитних карток Парекс-банку біля 7000 доларів, з комп'ютерної бази поліції одного з міст Америки зникла вся база по автомобілям, що перебували у розшуку. Для попередження несанкціонованого доступу використовуються дуже дорогі системи захисту, а також вдосконалюється програмне забезпечення.

При використанні програмного забезпечення існує можливість втрати інформації, спричинена дією вірусів, які використовують його недоліки. У зв'язку з тим, що вартість інформації росте, втрати можуть бути суттєвими.

Легкість тиражування інформаційних продуктів надає змогу з легкістю порушувати авторські права. Це стосується, в першу чергу, програмного забезпечення.

### **Контрольні запитання**

1. Поясніть принципи агрегації інформації.
2. Наведіть час, коли комп'ютер став персональним.
3. Який інструментарій інформаційних технологій?
4. Як опрацювати інформаційні дані?

*У цьому розділі подано поняття, що таке інформаційна технологія, етапи розвитку інформаційних технологій, складові частини інформаційної технології, Інструментарій інформаційної технології, небезпека та труднощі використання інформаційних технологій*

# ВИСНОВКИ

Одним із засобів керування розвитком інтелекту і підвищення його організованості на сучасному етапі є інформатизація суспільства, що ґрунтується насамперед на розвитку інформаційних комп'ютерних технологій. Значення інформаційної технології величезне - вона формує передній край науково-технічного прогресу, створює інформаційний фундамент розвитку науки і всіх інших технологій. Головними, визначальними стимулами розвитку інформаційної технології, є соціально-економічні потреби суспільства, і саме зараз суспільство як ніколи зацікавлене в якомога швидшій інформатизації та комп'ютеризації всіх без винятку сфер діяльності.

Дуже важливою властивістю інформаційної технології є те, що для неї інформація є не тільки продуктом, але і вихідною сировиною. Особлива роль приділяється всьому комплексу інформаційної технології і техніки в структурній перебудові економіки у бік наукоємності. Більш того, інформаційна технологія є свого роду перетворювачем всіх інших галузей господарства, як виробничих, так і невиробничих, основним засобом їхньої автоматизації, якісної зміни продукції і, як наслідок, їх переходу частково або цілком у категорію наукомістких. Пов'язаний з цим і працеозаощаджувальний характер інформаційної технології, що реалізується, зокрема, у керуванні багатьма видами робіт і технологічних операцій.

Безсумнівною перевагою інформаційної технології є те, що вона сама створює засоби для своєї еволюції. Формування системи, що саморозвивається - найважливіший підсумок, досягнутий у сфері інформаційної технології.

Таким чином, усі вищевикладені риси інформаційної технології вказують на те, що вона й у майбутньому залишиться самим перспективним видом технології, що допомагає людині впевнено крокувати шляхом прогресу.

# СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Автоматизовані системи обробки економічної інформації: Підручник / Г.В.Лавнський, Т.В. Оболенська, Б.В. Маринченко, А.М. Тимошенко, Г.Ф. Стояк, Р.С. Гром'як, В.П. Попов, С.В. Бабій. За ред. Г.В. Лаврінського. – К.: Вища школа, 1995. – 287 с.
2. Архипов В.Ю. Інформаційно-пошукові системи Internet. // Секретарська справа. – 2001. – № 2. – С. 85-89.
3. Береза А.М. Інформаційні системи і технологій в економіці: Навч.-метод. посібник для сом ост. вивч. дисц. – К.: КНЕУ, 2002. – 80 с.
4. Борис Б. Энциклопедия Windows.-СПб.: Питер,.1998. – 816с.
5. Вебер Ральф «Сборка, конфигурирование, настройка, модернизация и разгон ПК. Энциклопедия пользователя» - пер. с немецкого. – К.: ООО «ДиаСофтЮП», 2002г. – 704с.
6. Виктор Пасько. BASE 2000 (русифицированная версия). – К: Издательская группа ВНУ, 1999. – 384 с.
7. Габбасов Ю.Ф. Internet 2000. Эффективные технологии. – СПб.: ВХВ- Санкт Петербург. – 1999 –VII. – 440 с.
8. Ганенко А.П., Милованов Ю.В. Оформление текстовых и графических материалов: Учеб. пособие – М.- 2001. – 352 с.
9. Гук М. «Аппаратные средства локальных сетей. Энциклопедия.» - СПб: Издательство «Питер», 2000г. – 576 с.
10. Джанад Д.С. Бизнес в Internet. Пособие по интерактивному маркетингу. – Ростов н/Д: Феникс. – 1987 – 460 с.
11. Динеш А. Linux Red Hat 7.1. Полное руководство: Учебн. пособие. – К.:ВЕК+, СПб.: Корона-Принт, М.: Энтроп, 2002. – 686 с.
12. Захист економічної інформації: Навчальний посібник./ за ред. Проф.. В.О. Хорошка. – Київ, НАУ, 2002. – 78 с.
13. Интернет. Всемирная компьютерная сеть. Практическое пособие и путеводитель. К. – 1995. – 178 с.
14. Карпов Б. Microsoft BASE 2000 : справочник – СПб: Издательство "Питер",2000. – 416 с.
15. Колесник А. П, Компьютерные системы в управлении финансами. — М.: Финансы и статистика, 1994. — 312 с.
16. Колесников А Excel 2000. – К, Наука. – 1999. – 530 с..
17. Колисниченко Д.Н. Linux-сервер своими руками. – 3-е изд., перераб. И доп. – СПб.: Наука и техника, 2005. – 744 с.

18. Кондрашова С.С. Информационные технологии в управлении. – К.: МАУП, 1998. – 132 с.
19. Коржинский С.Н. Настольная книга Web-мастера: Эффективность применения HTML. – изд. 2-т. – М.: Изд.Торг. дом «КноРус». – 2000. – 320 с.
20. Кошкин Л.И. Хачатуров А.Е. Булатов И.С. Менеджмент на промышленном предприятии. К.:Эколайн, 2000. – 250 с.
21. Крейнак Дж., Хабрейкен Дж. Интернет. Энциклопедия. – СПб.: Питер – 2000 – 555 с.
22. Кузнецов С.Л. Выбор и опытное внедрение системы электронного архива // Секретарское дело.- 2001.-№ 3. - С. 44-47.
23. Кульгин М. Технологии корпоративных систем. –СПб.: Питер, 2000. – 250 с.
24. Максимович Г.И. Электронные библиотеки. // Делопроизводство. - 2001. - № 1. - С. 85-89.
25. Мелюхин И.С. Проблемы информации //Состояние и тенденции развития рынка коммерческих онлайн-услуг.-№2.– 1996. – С.25-29.
26. Меррей Р. Компоненты цифровой библиотеки и их взаимодействие //Доклад на конференции и выставке (OnLine'99), – Лондон, 8., 1999, – С. 5.
27. Методические указания к комплексной лабораторной работе “Проектирование и создание базы данных в среде СУБД BASE” по дисциплине проектирование баз данных для студентов направлений подготовки: 0501 Экономика и предпринимательство, 0502 Менеджмент / сост.: С.Я. Фридман и др. – Днепрпетровск: Национальная горная академия Украины, 2002. – 227с.
28. Немет Э., Снайдер Г., Хейн Т. «Руководство администратора Linux» перевод с английского – М.: Изд. дом «Вильямс», 2005. – 880 с.
29. Оглтри Т.«Модернизация и ремонт сетей, 2-е изд.» пер. англ.:Уч. пособие – М.:Издательский дом «Вильямс», 2000. – 928 с.
30. Опенгейм Ч., Смітсон Д. Що таке гібридна бібліотека? //Новітні інформаційні технології та бібліотека – К.: Британська Рада в Україні. – С. 8-9.
31. Павлов В.А. Компьютеризация юридических процессов. –М.: ДИАСОФТ, 1995. – 250 с.
32. Пасько В Word 2000. – К,: Наука. – 1999. – 453 с..
33. Петров М. Информационные системы: Учебник для вузов. – СПб: - 2001. – 688 с.
34. Писаревська Г.А. Інформаційні системи в управлінні трудовими ресурсами: Навч.-метод. посібник для сом ост. вивч. дисц. – К.: КНЕУ, 1999. – 164 с.
35. Праг Керри Н., Ирвин Майкл Р. BASE 2000. Библия пользователя : Пер.с англ.: Уч. пособие – М: Издательский дом "Вильямс", 2000 – 1040 с.

36. Рогач І. Ф., Сендзюк М. А., Антонюк В. А. Інформаційні системи у фінансово-кредитних установах: Навч. посібник. – 2-ге вид., перероб. і доп. – К.: КНЕУ, 2001. – 239 с.
37. Рогач І.Ф., Сендзюк М.А., Антонюк В.А., Денисова О.О. Інформаційні системи у фінансово-кредитних установах: Навч.-метод. посібник для сом ост. вивч. дисц. – К.: КНЕУ, 2001. – 324 с.
38. Системное администрирование Linux: Учебн. пособие / М. Кармент, С. Деглер, Д. Деннис. – М.: Изд. Дом «Вильямс», 2000. – 319 с.
39. Скрипкин К. Г. Финансовая информатика. – М.: ТЕИС, 1997. – 160с.
40. Спортеш М.А., Пит Р., Коуди Д.Ф. Компьютерные сети. Энциклопедия пользователя. – К.: Диасофт, 1998 – 425 с.
41. Страссман Поль. Информация в век электроники. М., 1987. – 520 с.
42. Сютюренко В. Електронні інформаційні ресурси: проблеми створення і використання // Електронні бібліотеки.-№2.-1999. – С.25-29.
43. Тейт Windows 2000 для системного администратора. – СПб.: Питер, 2001.–786с.
44. Файбел В. Энциклопедия современных сетевых технологий. – К.: Комиздат. – 1998. – 687 с.
45. Фридман С.Я., Бирицкая А.А. Microsoft BASE. Часть 1. База данных. Таблицы. Запросы: Учеб. пособие. – Днепропетровск: Национальная горная академия Украины, 2002. – 127с.
46. Фридман С.Я., Логачев Е.Н. Microsoft BASE. Часть 2. Формы. Отчеты. Макросы. Управление базой данных: Учеб. пособие. – Днепропетровск: Национальный горный университет, 2003. – 155с.
47. Хилайер С., МизикД. «Программирование Active Server Pages» пер с англ – М.: «Русская Редакция», 1999г, С. 1- 4.
48. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – СПб: Корона принт, 2000. – 416 с.
49. Шафран Э. Создание Web – страниц: Самоучитель – СПб: Питер. – 2001. – 450 с..
50. Шикин Е.В.Компьютерная графика.-М.:Диалог-Мифит.-1995.-288с.
51. Экономическая информатика / Под ред. П.В. Конюховского и Д.Н. Колесова. – СПб: Издательство "Питер", 2000. – 560 с.

# ПРЕДМЕТНИЙ ПОКАЖЧИК

- HTML-коди 106
- ICQ 83
- IN, BETWEEN, LIKE, is NULL 155
- Internet Explorer 59
- Intranet Chat 46
- IP-адресація комп'ютерів у Internet 50
- JavaScript 124
- Mozilla Firefox 62
- Net Wampire 84
- Netscape Navigator 60
- Opera 66
- Outlook Express 70
- The Bat 74
- WinPopUp 45
- Агрегатовані і групові функції 158
- Адаптивні інформаційні системи 210
- Антивіруси 95
- Апаратне забезпечення глобальних мереж 48
- Апаратне забезпечення локальних мереж 39
- Архітектура комп'ютера 25
- Браузери Інтернету 59
- Використання COUNT замість EXISTS 166
- Використання UNION з ORDER BY 167
- Використання оператора EXISTS 164
- Вимірювання інформації 20
- Візуальні редактори html-сторінок 125
- Віртуальна електронна пошта 75
- Вкладені підзапити 162
- Глобальні мережі 47
- Дані VBA: типи даних, змінні і константи 179
- Доменна система адрес в Інтернеті 54
- Експертні системи (ЕС) 202
- Етапи розвитку інформаційних технологій 211
- Захист операційних систем 93
- Зв'язані підзапити в HAVING 164
- Інструментарій інформаційної технології 213
- Інформаційна технологія експертних систем 217
- Інформаційна технологія керування 215
- Інформаційна технологія опрацювання даних 214
- Інформаційна технологія підтримки прийняття рішень 216
- Історія розвитку інформатики 16
- Команди настройки локальної мережі засобами Windows 43
- Комп'ютерні віруси 91
- Локальні мережі 39
- Магістрально-модульний принцип архітектури персональних комп'ютерів .....
- Мейлери Інтернету 68
- Мережні хробаки 93
- Міжмережні екрани 97
- Міжнародні пошукові сервери 79
- Можливості Інтернету 75
- Надійність інформації 22



Носії інформації. Види і властивості інформації 17

Оператор об'єднання UNION 166

Основні об'єкти і методи вивчення науки інформатики 13

Основні підходи до визначення поняття "інформація" 17

Основні поняття реляційних баз даних 146

Основні прийоми роботи з редактором Visual Basic 195

Основні тенденції розвитку апаратного забезпечення комп'ютера 27

Особливості та класифікація ЕС 202

Перетворення висновку і вбудовані функції 157

Покоління ЕОМ і їх основні характеристики 28

Поняття "чату" і форуму в Інтернеті 80

Поняття сервера та робочої станції 42

Постачальник послуг Інтернету (ISP або Провайдер) 52

Поштовий агент браузера Opera 75

Представлення інформації в комп'ютері 30

Приклади Web-сторінок 116

Принципи архітектури і історія розвитку комп'ютерів 25

Програма REGEX 86

Програмні засоби для обчислювальних робіт 138

Програмні засоби для роботи з текстом 137

Програмні засоби роботи з графікою 141

Програмні засоби роботи із звуком 143

Проектування структури сайту 104

Публікація Web-вузла в Internet 130

Редактори html-кодів 128

Російські пошукові сервери 78

Самонавчальні системи 207

Символьні функції 148

Системи з інтелектуальним інтерфейсом 201

Системи контролю змісту та антиспаму 99

Складові частини SQL 147

Способи з'єднання з Інтернетом 52

Структура ЕС 205

Таблиці стилів CSS 118

Теорія інформації 22

Технології інформаційної безпеки 93

Технологія проектування і розробки ЕС 205

Типи даних SQL 148

Топологія мереж 40

Тривалість передачі інформації 23

Троянські програми 92

Українські пошукові сервери 78

Упорядкування вихідних полів (ORDER BY) 161

Усунення дублювання в UNION 166

Формування зв'язаних під запитів 163

Числові типи даних 148

## Додаток А.

# СЛОВНИК ЧАСТО ВЖИВАНИХ ТЕРМІНІВ

**BC:** (blind copy) – електронні адреси тих, кому направляються копії вашого листа, але основний одержувач листа, на відміну від CC не знатиме, що копії пішли ще комусь.

**CC:** (carbon copy) – електронні адреси тих, кому направляються копії вашого листа. Адреси записуються через кому.

**CD-ROM** – пристрій для зчитування компакт-дисків. (CD)

**DATE:** (дата) – вставляється автоматично.

**DBF-формат** – формат файлів БД.

**Default Gateway** – стандартний шлюз, який підказує, де шукати потрібну адресу.

**FAMT (Fully-automated machine translation)** - повністю автоматизований машинний переклад;

**Finger** - одержання інформації про користувачів вилученого комп'ютера.

**FROM:** (від кого) електронна адреса одержувача листа, наприклад [pupkin@ab.com](mailto:pupkin@ab.com). Вставляється автоматично мейлером – програмою, що відправляє листи електронною поштою.

**FTP** (File Transfer Protocol) - протокол передачі файлів. Дає можливість абоненту обмінюватися двоїчними і текстовими файлами з будь-яким комп'ютером мережі. Установивши зв'язок з вилученим комп'ютером, користувач може скопіювати файл із вилученого комп'ютера на свій чи скопіювати файл зі свого комп'ютера на вилучений.

**Gopher** - найбільше широко розповсюджений засіб пошуку інформації в мережі Internet, що дозволяє знаходити інформацію з ключовим словом і фразам. Робота із системою Gopher нагадує перегляд змісту, при цьому користувачу пропонується пройти крізь ряд вкладених меню і вибрати потрібну тему. У Internet у даний час понад 2000 Gopher-систем, частина з яких є вузькоспеціалізованою, а частина містить більш різнобічну інформацію.

**HAMT (Human-assisted machine translation)** - машинний переклад при участі людини;

**HTML-сторінка** – файли, записані у спеціальних командах (тегах) текстом та посиланнями на малюнки або інші сторінки.

**ICQ** (гра слів від англійського I Seek You - "я тебе шукаю") - це програма, що дозволяє кожному користувачу Інтернету установити в будь-який конкретний момент, хто з його друзів по заданому списку (які також є клієнтами програми) приєднаний до мережі

**Internet Explorer** – браузер Інтернету

**Iptunnel** - дає можливість доступу до сервера ЛВС NetWare з яким немає безпосереднього зв'язку по ЛВС, а мається лише зв'язок по мережі Internet.

**Lpq** – мережний друк. Показує файли, що стоять у черзі на друк на мережному принтері.

**МАНТ (Machine-assisted human translation)** - переклад, здійснюваний людиною з використанням комп'ютера.

**Microsoft Outlook** - інформаційна система, що допомагає вам зберігати дані й обмінюватися ними з іншими користувачами.

**Net Vampire** – універсальний менеджер, якого ви можете використовувати, щоб знайти, список і завантажити файли з серверів Інтернету

**Netscape Navigator** – браузер Інтернету

**News.com2com.ru** – сервер новин, на якому можна підписатися на телеконференції новин по різних групах:

**NFS (Network File System)** - розподілена файлова система. Дає можливість абоненту користатися файловою системою вилученого комп'ютера, як своєї власній.

**Page Composer** – програми створення веб-сторінок

**Ping** - перевірка доступності віддаленої ЕОМ по мережі.

**PPP (Point to Point Protocol)** – протокол з точки на точку,

**REGEX** – створює чергу викачування

**Rexec (Remote Execution)** - виконання однієї команди на віддаленій UNIX-машині.

**Rsh (Remote Shell)** - віддалений доступ. Аналог Telnet, але працює тільки в тому випадку, якщо на вилученому комп'ютері стоїть ОС UNIX.

**SLIP (Serial Line Internet Protocol)** – протокол для серійного підключення

**SUBJECT:** (або SUBJ) – тема вашого листа. Інколи, щоб не писати в тексті вдруге цю тему, пишуть просто subj. Наприклад, якщо темою листа є купівля комп'ютера, то пишуть : “Куплю subj, не гірше за пентіум-2”.

**SWIFT** – міжнародна банківська система електронних платежів.

**Talk** - дає можливість відкриття "розмови" з користувачем віддаленої ЕОМ. При цьому на екрані одночасно видний текст, що вводиться, і відповідь вилученого користувача.

**Talk** - дає можливість відкриття "розмови" з користувачем віддаленої ЕОМ. При цьому на екрані одночасно видний текст, що вводиться, і відповідь вилученого користувача.

**TCP/IP** - множина комунікаційних протоколів, що визначають, як комп'ютери різних типів можуть спілкуватися між собою.

**Telnet** - віддалений доступ. Дає можливість абоненту працювати на будь-якій ЕОМ мережі Internet як на своїй власній. Тобто запускати програми, змінювати режим роботи і т.д.

**TO:** (кому) – електронна адреса одержувача листа, наприклад [ekit@ngau.ua](mailto:ekit@ngau.ua).

**WAIS** - ще більш могутній засіб одержання інформації, чим Gopher, оскільки воно здійснює пошук ключових слів у всіх текстах документів. Запити посилаються в WAIS на спрощеній англійській мові. Це значно легше, ніж

формулювати їх мовою алгебри логіки, і це робить WAIS більш привабливою для користувачів-непрофесіоналів.

**Webster** - мережна версія тлумачного словника англійської мови.

**Web-сайт** – HTML-сторінка, розміщена для представлення інформації про ту чи іншу фірму або особу.

**Whois** - адресна книга мережі Internet. По запиті абонент може одержати інформацію про приналежність вилученого комп'ютера, про користувачів.

**WinSock** – програми, що з'єднують Інтернет з Windows.

**WWW** - система для роботи з гіпертекстом. Потенційно вона є найбільш могутнім засобом пошуку. Гіпертекст з'єднує різні документи на основі заздалегідь заданого набору слів. Наприклад, коли в тексті зустрічається нове чи слово поняття, система, що працює з гіпертекстом, дає можливість перейти до іншого документу, у якому це чи слово поняття розглядається більш докладно.

**YaHoo, Rambler, Torba, Brama** – пошукові сервери

**АРМ** – автоматизоване робоче місце.

**База даних (БД)**- набір пов'язаних таблиць, запитів, форм, звітів, макросів і модулів. У BASE база даних являє собою єдиний файл з розширенням .MDB.

**Бекбон (Backbone)** – загальна, первинна лінія зв'язку первинних провайдерів. Оплата за Інтернет – погодинна або за місяць, без обмежень в часі користування.

**Бланк** - очищення всіх полів на екрані.

**Браузер** – програма: яка інтерпретує HTML-сторінки для представлення їх у вигляді документу.

**Вербальна інформація** – Інформація, послана відправником з використанням слів

**Видалення** - тимчасове видалення / оновлення записів в загальному списку

**Викачування** – безперервне отримання інформації з Інтернету

**Вираз** - є основним засобом підтримки виконання багатьох операцій Microsoft BASE. Вираз представляє комбінацію символів, ідентифікаторів, операторів і значень, що дає певний результат.

**Вихідні показники** – ті показники системи, зміни яких викликають вихідну дію або вихідний сигнал, або самі є такою дією або сигналом.

**Виходи** – елементи системи, які здійснюють дію або передають сигнал в іншу систему.

**Вікно БД** - основне вікно, за допомогою якого в BASE здійснюється перегляд або конструювання об'єктів БД.

**Вікно діалогу** - форма, що належить додатку Windows або створена в BASE, яка дозволять користувачеві вводити інформацію.

**Властивість** - атрибут або характеристика об'єкта.

**Впроваджений об'єкт** - документ (джерело), що зберігається всередині іншого документа (приймача).

**Вхідні показники** – показники системи, які змінюються в результаті вхідної дії або сигналу.

**Входи** – елементи системи, до яких прикладені вхідні дії або на які поступають вхідні сигнали.

**Гіперлінк** – посилення на іншу веб-сторінку

**Графічний інтерфейс користувача (ГІК)** - це сукупність засобів спілкування користувача з БД. Він містить: систему панелей управління (кнопкові форми); меню користувача, включаючи і контекстне меню користувача; панелі користувача інструментів; різного роду кнопки.

**Дані** – Інформація, оброблена і представлена у формалізованому вигляді для подальшої обробки.

**Економічна діяльність** - процес поєднання дій, які призводять до отримання відповідного набору продукції чи послуг. Вид діяльності характеризується використанням ресурсів, виробничим процесом, випуском продукції та наданням послуг.

**Електронна пошта** - обмін поштовими повідомленнями з будь-яким абонентом мережі Internet. Існує можливість відправлення як текстових, так і двоїчних файлів. На розмір поштового повідомлення в мережі Internet накладається наступне обмеження - розмір поштового повідомлення не повинний перевищувати 64 кілобайт.

**Електронна пошта** - обмін поштовими повідомленнями з будь-яким абонентом мережі Internet. Існує можливість відправлення як текстових, так і двоїчних файлів

**Електронний перекладач** – робить переклад присланого на нього тексту з однієї мови на іншій. Звертання до електронних перекладачів відбувається за допомогою електронної пошти.

**Забезпечення цілісності даних** – правила зв'язку ключового поля однієї таблиці із зовнішнім ключем іншої.

**Запис** – інформація, що міститься в одному рядку списку або таблиці.

**Запит** – об'єкт бази даних, який дозволяє відбирати записи і поля таблиці, а також сортувати їх.

**Запит ді** – запит, який оновлює таблицю.

**Зв'язки один-до-багатьох (1:m)**-это такі зв'язки, коли кожного запису таблиці А може відповідати декілька записів таблиці В, а кожного запису таблиці В може відповідати тільки один запис таблиці А.

**Зв'язки один-до-одного (1:1)** мають місце, коли кожного запису таблиці А відповідає тільки один запис таблиці В і навпаки, кожного запису таблиці В відповідає один запис таблиці А.

**Індекс** – список службових даних з таблиці BASE, що дозволяє підвищити швидкість пошуку і сортування. Індекс може будуватися на основі одного або декількох полів.

**Інсталяція** – встановлення програми на ваш комп'ютер.

**Інтернет** – всесвітня комп'ютерна мережа, яка має доступ по IP протоколу.

**Інформаційна база** – файли на комп'ютері, які містять потрібну інформацію, розміщену з можливістю пошуку потрібних її елементів.

**Інформаційна культура** – Уміння цілеспрямовано працювати з інформацією і використовувати для її отримання, обробки і передачі комп'ютерну інформаційну технологію, сучасні технічні засоби і методи.

**Інформаційна технологія (ІТ)** - процес, що використовує сукупність засобів і методів збору, обробки і передачі даних (первинної інформації) для одержання інформації нової якості про стан об'єкта, процесу або явища.

**Інформаційне суспільство** – Суспільство, в якому більшість працюючих зайнята виробництвом, зберіганням, переробкою і реалізацією інформації, особливо вищої її форми – знань.

**Інформаційний менеджмент** – Технологія, компонентами якої є документарна інформація, персонал, технічні і програмні засоби забезпечення інформаційних процесів, а також нормативно встановлені процедури формування і використання інформаційних ресурсів.

**Інформаційні ресурси** – Сукупність даних, організованих для отримання достовірної інформації.

**Картотека** – електронний аналог паперової К., що створюється користувачем програми

**Каскадне видалення** - засіб підтримки цілісності даних в пов'язаних таблицях, забезпечуюче автоматичне видалення всіх пов'язаних записів в підлеглих таблицях при видаленні запису в головній таблиці.

**Каскадне оновлення** - засіб цілісності даних в пов'язаних таблицях, забезпечуюча автоматична зміна всіх пов'язаних записів в підлеглих таблицях при зміні запису в головній таблиці.

**Ключ або ключове поле** - це поле є ідентифікатором запису.

**Контроль** – Процес перевірки і зіставлення фактичних результатів із завданнями.

**Конфігурування** – зміна параметрів розрахунків в бухгалтерських програмах.

**Конфігурування** – зміна параметрів розрахунків в бухгалтерських програмах.

**Логічна структура РБД** включає всі таблиці БД і зв'язки між ними.

**Майстер діаграм** - інструмент додатку BASE.

**Макрос** - заделегідь записаний набір команд, що автоматично виконується програмою BASE.

**Макросередовище** – Загальні умови, в яких організація знаходиться.

**Машинний переклад** – комп'ютерний переклад.

**Менеджмент** – Управління в соціально-економічних системах: сукупність сучасних принципів, методів, засобів і форм управління виробництвом з метою підвищення його ефективності і збільшення прибутків.

**Меню** – перелік можливих дій в програмі

**Меню** – перелік можливих дій в програмі

**Мережева карта** – додаткова електронна плата вставлена в материнську плату.

**Мета** – Очікуваний, кінцевий результат.

**Мікросередовище** – Безпосереднє оточення організації.

**Місія** – Чітко виражена причина існування організації, яка підкреслює її відмінності від аналогів.

**Модем** (це скорочення від фрази “модулятор-демодулятор”) – електронний пристрій: що забезпечує передачу комп’ютерних даних по виділених лініях зв’язку.

**Модуль** - набір процедур.

**Мотивація** – Спонука працівників до активної діяльності за допомогою зовнішніх чинників (матеріальне і моральне стимулювання).

**Невербальна інформація** – Інформація, послана відправником без використання слів.

**Неформальна інформація** — Інформацію, одержувану неформальним шляхом від компетентних працівників.

**Новини** - одержання мережних новин і електронних дощок оголошень мережі і можливість приміщення інформації на дошки оголошень мережі. Електронні дошки оголошень мережі Internet формуються по тематиці. Користувач може по своєму виборі підписатися на будь-які групи новин.

**Новини** - одержання мережних новин і електронних дощок оголошень мережі і можливість приміщення інформації на дошки оголошень мережі. Електронні дошки оголошень мережі Internet формуються по тематиці. Користувач може по своєму виборі підписатися на будь-які групи новин.

**Нормалізація таблиць** - це покроковий оборотний процес розкладання початкових (універсальних) таблиць БД на інші більш дрібні і прості. При цьому усувається вся недопустима функціональна залежність (ФЗ), досягається мінімальне дублювання даних і забезпечується їх обробка і оновлення без труднощів.

**Off-line** – дослівно означає “вимкнути з лінії”. Для глобальних мереж це означає такий режим роботи сервера, коли він, отримавши інформацію від клієнта: чекає сеансу зв’язку з потрібним йому сервером, що лежить на маршруті передачі повідомлення.

**On-line** – режимі роботи, в яких всі сервери в мережі водночас підтримують зв’язок із усіма іншими серверами, або мають можливість негайно: за запитом клієнта, встановити цей зв’язок.

**Об’єкт** - цей термін означає будь-які таблиці, запити, форми і інші елементи, розташовані у вікні бази даних.

**Онлайнний переклад** – переклад із застосуванням web-сторінки.

**Орган управління** – це система, на вхід якої поступають сигнали про стан керованого об’єкту і середовища, а на виході - сигнал про необхідне в даній ситуації управління.

**Організація** – Впорядкування діяльності менеджера і виконавців.

**Пакет програм** – набір програм, які вирішують єдину задачу.

**Панель елементів** - панель інструментів для форми або звіту, яка містить кнопки, що додають в бланк конструктора різні елементи управління.

**Параметри** – режими роботи програми.

**Парус** – інформаційно-розрахункова бухгалтерська система для великих підприємств та корпорацій.

**Первинний ключ** - поле (або декілька полів) які однозначно ідентифікуює запис таблиці.

**Перетин двох систем** – система, що складається з елементів, що належать одночасно обом цим системам.

**Планування** – Процес підготовки рішень.

**Повідомлення** – це сукупність сигналів.

**Поле** - інформація, що міститься в стовпці таблиці або списку.

**Пошук по ключу** – пошук інформації в базі законів України по заданій ознаці (ключу), наприклад, за назвою документу: за органом, що його випустив, датою введення в дію і т.ін.

**Предметна область (ПО)** - це частина реального світу, що підлягає вивченню з метою організації управління і, зрештою, автоматизації. ПО може бути підприємство, міністерство, учбовий заклад і інш.

**Прикладення СУБД BASE** - ця БД, забезпечена графічним інтерфейсом користувача (ГИП).

**Редагування** - редагування вже існуючих відомостей;

**Режим конструктора** - метод відображення таблиць, запитів, форм і макросів, який дозволяє створювати і модифікувати структуру об'єктів.

**Режим таблиці** - метод відображення таблиці, запиту або форми у вигляді рядків і стовпців, які часто використовуються для введення даних.

**Режим форми** - режим відображення форми, в якому користувач може вводити дані.

**Реляційна БД (РБД)** являє собою сукупність взаємопов'язаних двумірних таблиць, в кожній з яких міститься зведення про один об'єкт.

**Ринок інформаційних продуктів і послуг** – Система економічних, правових і організаційних відносин по торгівлі продуктами інтелектуальної праці на комерційній основі.

**Рух (поведінка) системи** – це процес переходу системи з одного стану в інше, з нього в третє і т.д.

**Сервер** – вузловий комп'ютер в мережі Інтернет: який підтримує режим пересилання інформації та забезпечує зберігання Web-сайтів.

**Сервер** – вузловий комп'ютер в мережі Інтернет: який підтримує режим пересилання інформації та забезпечує зберігання Web-сайтів.

**Середовище системи** – система, що складається з елементів, що не належать цій системі.

**Сигнал** – повідомлення про стан елементу.

**Система керування базами даних (СКБД)** прикладна програма, яка дозволяє формувати БД, проводити пошук необхідних даних

**Стан системи** – сукупність значень її показників.



**Стиск** – остаточне видалення викреслених записів із загального списку працівників;

**Стратегічне управління** – Управління, яке орієнтоване на забезпечення виживання організації і її ефективного розвитку в умовах динамічно змінного зовнішнього середовища.

**Структура системи** - це сукупність її елементів і зв'язків між ними, по яких можуть проходити сигнали і дії.

**Таблиця** - первинний об'єкт бази даних, де зберігається інформація.

**Телекомунікації** - дистанційна передача даних на базі комп'ютерних мереж і сучасних технічних засобів зв'язку.

**Технологія** – Сукупність виробничих процесів і операцій, що забезпечують нормальне функціонування об'єктів управління.

**Тип даних** - опис поля, який визначає, який вигляд інформації можна вводити в це полі.

**Трансформаційні процеси** – Процеси зміни характеристик інформації.

**Універсальною** називається таблиця, яка містить всі поля (реквізити) БД.

**Факс-сервіс** - дає можливість користувачу відправляти повідомлення по факсимільному зв'язку, користаючись факс - сервером мережі.

**Фільтр** - режим, що дозволяє відображати тільки деякі записи таблиці.

**Форма** - структурований і форматований об'єкт, призначений для введення даних.

**Формальна інформація** — Інформацію, передача якої здійснюється відповідно до діючих інструкцій і розпоряджень. Недолік такої інформації у тому, що вона долає велике число каналів, що займає багато часу.

**Функціональна залежність** може бути пояснена таким чином. Нехай є два поля А і В. Якщо в будь-який момент часу кожному значенню поля А відповідає не більш, ніж одне значення поля В, то кажуть, що поле В функціонально залежить від поля А.

**Чат** – проведення конференції в режимі реального часу

**Шаблони документів** – електронні образи форм документів звітності.

**Шлюзи** - дають можливість абоненту відправляти повідомлення в мережі, що не працюють із протоколами TCP/IP (Fido, Goldnet, AT50, Relcom).

**Якісний чорновий машинний переклад** – сучасна заміна старого терміна "Якісний машинний переклад".

# Додаток Б.

## ФІНАНСОВІ РЕСУРСИ ІНТЕРНЕТУ

### Бухгалтерське та фінансове програмне забезпечення

#### 1. **"Автоматика" - комп'ютерні програми для підприємців**

Програми для приватних підприємців. Детальний складський облік, звітність в ДПА і фонди, первинна документація, виробництво, аналіз реалізації.  
<http://automatika.dnopr.net>

#### 2. **1-й ТоварищЪ**

<http://camrad.hypermart.net/>

#### 3. **A3 Consulting**

<http://a3.da.ru/>

#### 4. **Business Service (BS Integrator 2.4)**

Комплексна автоматизація підприємства: автоматизація фінансового, податкового, оперативного обліку, контроль за проходженням всіх бізнес процесів.  
<http://www.bsi.com.ua>

#### 5. **EE(C)**

Офісні програмні пакети для малого бізнесу (на базі Microsoft Office).  
<http://www.eesoft.narod.ru/>

#### 6. **EXPRESS SOFT**

<http://ivasoft.biz-page.com/>

#### 7. **IDM (FinExpert)**

<http://www.FinExpert.com>

#### 8. **IT PRO**

Ліцензійне програмне забезпечення.  
<http://www.itpro.kiev.ua/>

#### 9. **R-Style Ukraine**

Розробка програмного забезпечення для автоматизації діяльності фінансово-кредитних установ.  
<http://www.r-style.kiev.ua>

#### 10. **SoftPro**

Розробка та впровадження програмного забезпечення для управління офісною діяльністю (програмний комплекс "Універсал")

<http://www.wgsoftpro.com>

#### 11. **SoftTAXI (X-DOOR)**

<http://www.softtaxi.com.ua>

#### 12. **ІМПАКТ (Акцент 6.0)**

Розробка програмного забезпечення для автоматизації бухгалтерського обліку на підприємствах усіх форм власності та видів діяльності в Україні.

<http://www.accent6.com>

#### 13. **Інтелект-Сервіс (БЭСТ)**

<http://www.intelserv.kiev.ua>

#### 14. **АБ Системи (Офіс 2000)**

<http://www.ab-system.com>

#### 15. **АЛЬТ**

Дослідницько-консультаційна фірма (Росія. Представництво в Україні) - програмні продукти, призначені для рішення ключових задач управління фінансами підприємства.

<http://www.altrc.ru>

#### 16. **АО "Удача Баланс" (Главбух 2)**

"Главбух 2" - мережева багатофункціональна програма автоматизації бухгалтерського і управлінського обліку на підприємстві.  
<http://www.balans.com.ua>

#### 17. **Бізнес Мікро**

Фінансове ПЗ. <http://www.bmicro.ru>

#### 18. **БЕСТ ЗВІТ**

Програма для податкової звітності.  
<http://www.bestzvit.com.ua>

19. **Баланс Клуб (SoNet)**

<http://www.balance.dp.ua>

20. **Банкомзв'язок (Облік)**

"Облік" - система автоматизації управлінського та бухгалтерського обліку на підприємстві.

<http://www.bkc.com.ua>

21. **Бухгалтерські програми для України**

<http://www.yvs.makeevka.com/>

22. **Восток Сервіс**

"Кредитний Союз" - система автоматизації учета в кредитних союзах (на базі системи "bravo"). Програмний комплекс "ABACUS Professiona" фірми "Омега" для бухгалтерського учета по українським і міжнародним стандартам.

<http://www.vs.com.ua>

23. **Все про автоматизовані системи управління підприємством**

Сайт є незалежним проектом і представляє собою динамічний, постійно поновлений довідник АСУ і компаній, які їх представляють на українському ринку. На сайті розміщуються новини компаній і продуктів про можливості, технологій, що використовуються, проектів, які реалізуються.

<http://www.allasu.com>

24. **ДЕБЕТ (Дебет Плюс)**

<http://www.debetplus.com.ua>

25. **Дубль-Б Центр**

<http://center.kharkov.com/>

26. **ЗАТ "Бізнес-Автоматика"**

Розробка корпоративних клієнт-серверних програмних систем

<http://business-automatic.com>

27. **Земляничні поляни (Lady Fin)**

<http://www.strawberryfields.kiev.ua/>

28. **КСИКОМ СОФТ**

<http://www.ksicom.com>

29. **Компанія "Інфосервіс"**

<http://www.infoservice.kiev.ua/>

30. **Компанія "МІРАТЕХ -Про" (Віртуоз і Міракл)**

<http://www.miratech-pro.com.ua>

31. **Компанія "Омега" (ABACUS)**

Системи комплексної автоматизації фінансово-господарської діяльності підприємств і бухгалтерського обліку серії АВАСУС (Росія)

<http://www.omega.ru>

32. **Компанія SoftLine (система "Мегаполіс")**

<http://www.softline.kiev.ua/>

33. **Компанія «Технобаланс»**

Регіональний дистриб'ютор і авторизований центр впровадження в Україні КІС "ФЛАГМАН" і програм "ІНТЕГРАТОР" і "ЗАРПЛАТА". Побудова комплексних автоматизованих систем управління фінансово-господарською діяльністю підприємств.

<http://www.tb.com.ua>

34. **Корпорація "ГАЛАКТИКА"**

Програми для бізнесу і управління

<http://www.galaktika.com.ua>

35. **Корпорація "ПАРУС"**

Парус-Підприємство 8.2, Консультант, Бюджет

<http://www.parus.com.ua>

36. **НОТЕК (Товстий Ганс)**

<http://www.notek.zp.ua>

37. **НПП "Інфосервіс" (Полтава)**

Розробка, впровадження і супровід програмного забезпечення для автоматизації управління підприємством. Комплекс програм

"Бухгалтер в законі" (Держреєстр України АП № 4048)  
<http://www.infoservice.poltava.ua>

38. **НТФ "Інтес"**

Програми для учасників ЗЕД -  
<http://www.ln.com.ua/%7Enext/intes.htm>

39. **Науково-виробнича фірма "GrossBee"**

Розробка систем автоматизації обліку і управління підприємством.  
<http://www.grossbee.com>

40. **ПМКП АЛЬОША (GriS)**

Склад+бухоблік в торгівлі.  
<http://gris.la4.net>

41. **ПП Ів@новЪ**

Бухгалтерська програма "Баланс-Мастер"  
<http://www.balans.lg.ua/>

42. **Програма "Друк бланків Україна"**

Програма для обліку і ведення господарської та бухгалтерської документації в Україні  
<http://www.ukrblank.narod.ru/>

43. **Програмне забезпечення "RoboBAT"**

ПО для проектування в промисловому і громадському будівництві  
<http://www.ar-cadia.com.ua>

44. **СЕРВІС БУХГАЛТЕРА**

Програма для роботи з платіжними документами  
<http://www.hc.donetsk.ua/wserv.html>

45. **СофтПро (Універсал)**

<http://www.softpro.kharkov.com>

46. **Степ 2000**

<http://www.step2000.dp.ua/>

47. **Студія Плюс**

Автоматизація закладів громадського харчування  
<http://www.studioplus.com.ua>

48. **Український файловий архів**

Розділ "Програми для бізнесу": бухгалтерія, склад, платіжки, облік, офіс. <http://ufa.com.ua/catalog/78.html>

49. **Хакерз-Дизайн (Фінанси без проблем)**

<http://www.hdru.com/russian/index.htm>

50. **ЦПО "Октант"**

Редактор форм 8ДР.  
<http://www.octant.com.ua>

51. **Центр "Бізнес-консул"**

Автоматизація підприємств, ІТ-консалтинг, статті, книги по проблемах обліку. Програмний комплекс "Ростбиф.Общепит 7.43".

<http://www.bsconsul.net>

52. **ЮНІТОП (КРОНА)**

<http://www.unitop.poltava.ua>

53. **Юнитех+ (U-market, Fortius)**

<http://www.unitex.com.ua>

**Періодичні видання**

1. **Вісник Податкової служби України**

<http://www.visnuk.com.ua>

2. **Газета "Все про бухгалтерський облік"**

Новини законодавства, нормативні документи, консультації юристів, аудиторів, податківців, звітність,

школа бухгалтера і керівника  
<http://vobu.kiev.ua>

3. **Журнал "Бухгалтер"**

<http://www.buhgalter.kharkov.com/>

4. **Журнал "Бухгалтерія-Бізнес"**

<http://www.buhgalteria.com.ua/>

5. **Журнал "Головбух"**

<http://www.glavbuh.com.ua>

6. **Тижневик "Баланс"**

<http://www.balance.dp.ua>

7. **Тижневик "Дебет-Кредит"**

<http://www.dtk.com.ua>

8. **Тижневик "Податки і бухгалтерський облік"**

<http://www.factor.kharkov.com>

**Бланки та зразки документів**

1. **Biscont**

Бланки договорів, актів, заяв та бланки податкової та фінансової звітності у форматі zip, word

<http://biscont.com.ua/cgi-bin/ua/out.cgi?blanki1>

2. **Blank.com.ua**

Бланки податкової та фінансової звітності у форматі Excel, Word

<http://www.blank.com.ua>

3. **E-Pravo**

Розділ "Договори і форми". Крім бланків бухгалтерської та фінансової звітності представлені форми документів для підприємства, комерції, по трудовому праву. Формат - doc.

<http://www.e-pravo.com.ua/doc.shtml?gr=16>

4. **Власна справа**

Розділ "Документи і шаблони". Велика кількість бланків та документів у форматі zip, doc, pdf. Є пошук по назві.

[http://www.vlasnasprava.info/ua/business\\_tools/templates.html](http://www.vlasnasprava.info/ua/business_tools/templates.html)

5. **Дебет-Кредит**

Бланки податкової та фінансової звітності у форматі pdf.

<http://www.dtk.com.ua/service/blanky/index.html>

6. **Кодекс**

Розділ "Бланки" - бланки звітності, бланки, призначені для обліку на підприємстві, трудові договори у форматі zip, word, htm.

<http://www.kodex.com.ua/blank.htm>

7. **Ліга**

Розділ "Типові форми" - Форми звітності та форми документообігу бухгалтерії у форматі word. Доступ - платний.

<http://www.liga.kiev.ua/typicalforms/>

8. **Оберон**

Бланки бухобліку та звітності у форматі zip, word

<http://basa.tav.kharkov.ua/blank.shtml>

9. **Портал "Громадський простір"**

Розділ форм та бланків документів, які використовуються у повсякденній бухгалтерській діяльності.

<http://www.civicua.org/consultant/static/forms2.jsp>

**Законодавство, електронні правові бібліотеки**

1. **"Аверс" - Центр юридичного і бухгалтерського сервісу (м. Дніпропетровськ)**

Електронний правовий журнал "Аверс-бухгалтерія". Документи, довідники, консультації

<http://www.avers.dp.ua>

2. **"БІЗНЕС в Україні"**

Нормативні документи, консультації, довідкова інформація для спеціалістів з бухгалтерії та оподаткування, ЗЕД, господарчих відносин, фондового ринку, страхування, юристів. Щоденні та щотижневі огляди законодавства, календар бухгалтера.

<http://www.dinai.com>

3. **LIGA Online**

Правова інформація, бізнес-портал  
<http://www.liga.kiev.ua>

4. **LexQuorum ("ЛексКворум")**

<http://www.lexquorum.com/>

5. **Веб-проект системи "НАУ"**

<http://www.nau.kiev.ua>

6. **Енциклопедія "Атлас Аналітика"**

Енциклопедія "Атлас Аналітика" - бібліотечно-бібліографічна електронна База Знань і Досвіду, складена компанією "Індустрія Інтелекту"

<http://www.e-atlas.com.ua>

7. **Законопроекти України**

<http://www.zakon.com.ua>

8. **Консультант Бухгалтера і Підприємця**

Консультації, коментарі і рекомендації провідних спеціалістів (бл. 1500), роз'яснення ДПАУ, тематичні добірки, основні законодавчі акти (20 000 документів).

<http://www.infodisk.com.ua>

9. **Конус**

Економіко-правова фірма, новини законодавства

<http://www.audit.kherson.ua/konus>

10. **Лоцман**

Укр. законодавство (Інфоресурс)

<http://Info.ResourceCorp.net>

11. **Офіційний вісник України**

Все поточне законодавство України у офіційному щотижневнику - бюлетені "Офіційний вісник України"

<http://www.gdo.kiev.ua>

12. **Правовий портал для автомобілістів України**

Інформація та правові акти стосовно організації дорожнього руху в Україні: правила, інструкції, стандарти. Екзаменаційні білети ДАІ (онлайн).

<http://users.i.com.ua/~cruiser/>

13. **Ресурси українського інтернету на правову тематику**

<http://www.lawukraine.com/>

14. **Управління і бухг.облік**

Каталог посилань на укр. електронні правові бібліотеки

<http://www.comcent.nikolaev.ua/biz/index.html>

15. **Центр комерційного права України**

Законопроекти

<http://www.aba.kiev.ua>

16. **Центр розвитку українського законодавства**

<http://www.uldc.kiev.ua>

**Фінансові та банківські ресурси**

1. **Business Information Network**

<http://bin.com.ua/>

2. **RQL- Україна-Діловий центр**

Інформація по фінансовому і фондовому ринку України

<http://www.bc.rql.kiev.ua>

3. **Teletrade Consulting**

Forex в Україні. Teletrade – надання доступу на ринок Форекс для трейдерів. Forex новини. Розміщення

грошових засобів.

<http://www.teletrade.com.ua/>

4. **Uabankir.com - Банківський сервер**

Все про банки та банківські послуги

<http://www.uabankir.com>

5. **Webmoney Transfer**

Глобальна система трансфера майнових прав

<http://webmoney.com.ua>

6. **Інформаційне агентство "Инфинсервис"**

<http://www.ifs.kiev.ua>

7. **АКБ "Східно-Європейський банк"**

Офіційний сайт Банку.

<http://www.eebank.com.ua>

8. **Агентство "Кредит-Рейтинг"**

Рейтингове агентство, яке надає в Україні послуги в області незалежної оцінки кредитоспроможності суб'єктів запозичень з присвоєнням внутрішніх кредитних рейтингів.

<http://www.credit-rating.com.ua>

9. **Агентство з розвитку інфраструктури фондового ринку**

Створення та ведення інформаційної бази даних про ринок цінних паперів

<http://www.smida.gov.ua>

10. **Атланта Капітал.Ефективні інвестиції.**

Український фондовий ринок: новини, аналітика, огляди

<http://www.atlanta.com.ua>

11. **База даних Банкрутство**

<http://www.bankrut.gov.ua/>

12. **Банки України**

<http://finance.com.ua/freeinfo/banks>

13. **Блокнот банкіра**

Блокнот банкіра

<http://www.uabanker.net>

14. **Все для обліку грошей**

<http://www.cash.com.ua/main.shtml>

15. **Все про банківську систему**

<http://banker.kiev.ua>

16. **Всеукраїнський центр електронної комерції**

<http://e-commerce.com.ua/>

17. **ДП "Агенція фондового ринку"**

Арбітражне управління, питання банкрутства, антикризисний менеджмент, продаж майна і об'єктів підприємств

<http://www.arbitrage.com.ua>

18. **Компанія "Фінансовий Аналітик - Сервіс"**

<http://www.fas.com.ua>

19. **Серго-Гамма-лізинг**

Лізинг, фінансовий (інвестиційний) менеджмент, маркетинг, консультації

<http://www.sg-leasing.com.ua>

20. **Система електронної доставки і оплати рахунків Portmone.com**

Доставка і оплата рахунків за комунальні послуги, телефон, мобільний зв'язок і ін. через Інтернет, телефон або WAP карточками Visa, Visa Electron и Mastercard.

<http://www.portmone.com>

21. **Українська інформаційна вексельно-фондова система**

Інформація про котировки, організація торгівлі цінними паперами і борговими зобов'язаннями

<http://www.sellbuy.com.ua>

22. **Українська Аграрна Біржа**

<http://www.uame.com.ua>

23. **Українська міжбанківська валютна біржа**

<http://www.uice.com.ua>

24. **Українська торгова система**

<http://uts.net.ua/>

25. **Український Фінансовий Монітор**

<http://www.ukrainet.lviv.ua/ufm/ukr.html>

26. **Український сервер страхування**

Все про страхування в Україні і за кордоном

<http://uainsur.com>

27. **Український фінансовий портал**

<http://www.finport.net>

28. **Український фінансовий сервер UFS**

<http://www.ufs.kiev.ua>

29. **Український фінансовий форум**

<http://www.floatservice.com/>

30. **Фінанси в Інтернет**

Український фінансовий форум

<http://www.money.dp.ua/>

31. **Фінансовий клуб України**

<http://www.finclub.kiev.ua>

32. **Фінансовий ринок України**

Спільний інформаційний проект  
Асоціації юристів фондового ринку і  
періодичних видань

<http://www.finmarket.info>

33. **Фінансовий та банківський гід**

<http://www.bank.ukrbiz.net/>

34. **Фондовий ринок України -  
система Wallstreet (с)**

<http://wallstreet.com.ua/>

35. **Харківський Ділінговий Центр**

Повний спектр послуг для стабільної  
роботи на фінансових ринках (FOREX і  
ін.). Навчання, новини, котировки,  
графіки, прогноз і аналіз в режимі  
реального часу.

<http://www.ifc-forex.com>





Навчальне видання

**Кочура Євген Віталійович**  
**Пістунов Ігор Миколайович**  
**Борщ Тетяна Вікторівна**  
**Рогов Юрій Ігоревич**

# **ЕКОНОМІЧНА ІНФОРМАТИКА**

**Навчальний посібник**

Редакційно-видавничий комплекс

Редактор  
Комп'ютерна верстка І.М. Пістунова

Підписано до друку 2007 . Формат 30 x 42/4.  
Папір офсетний. Ризографія. Умовн. друк. арк. 17,72.  
Обліково-видавн. арк. 17,73. Тираж 600 прим. Зам. №

Підготовлено до друку та надруковано  
в Національному гірничому університеті.  
Свідоцтво про внесення до державного реєстру ДК №1842.  
49027, м. Дніпропетровськ, просп. К. Маркса, 19.