



Національний університет

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет водного господарства та природокористування

Л. В. Зубик,

Я. Я. Зубик,

І. М. Карпович

ІНФОРМАТИКА ТА КОМП'ЮТЕРНА ТЕХНІКА У ВОДНОМУ ГОСПОДАРСТВІ

Навчальний посібник

Рекомендовано

*Міністерством освіти і науки України як
навчальний посібник
для студентів напрямку “Гідротехніка”*

Рівне – 2008



Національний університет

УДК 004(075.8)

ББК 32.973Я73

З-91

*Рекомендовано Міністерством освіти і науки України
(Лист № 1.4/18-Г-1698 від 09.07.2008 р.)*

Рецензенти:

Марценюк В.П., доктор технічних наук, професор, проректор з науково-педагогічної роботи, завідувач кафедри медичної інформатики Тернопільського державного медичного університету;

Рокочинський А.М., доктор технічних наук, професор, завідувач кафедри гідромеліорації національного університету водного господарства та природокористування;

Сяський А.О., доктор технічних наук, професор, завідувач кафедри інформатики та прикладної математики Рівненського державного гуманітарного університету.

Зубик Л. В., Зубик Я. Я., Карпович І. М.

З-91 Інформатика та комп'ютерна техніка у водному господарстві.

Посібник. – Рівне: НУВГП, 2008. – 306 с.

ISBN 978-966-327-103-3

У посібнику викладено засоби інформатики, необхідні для розв'язування сучасних інженерно-технічних задач. Розглянуто призначення операційних систем та принципи роботи з програмами MS Office. Детально викладено прийоми і методи створення програмних продуктів у середовищі Visual Basic 6.0.

Для студентів технічних спеціальностей вищих навчальних закладів, а та-кож усіх, хто використовує інформаційні технології в практичній діяльності.

УДК 004(075.8)

ББК 32.973Я73

ISBN 978-966-327-103-3 ©Зубик Л. В., Зубик Я. Я., Карпович І. М., 2008

©НУВГП, 2008

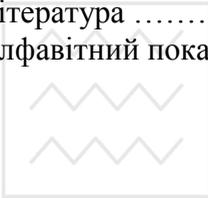


ЗМІСТ

Передмова	5
Розділ 1. Основи інформаційних технологій	6
1.1. Поняття інформації та інформатики	6
1.2. Обчислювальні машини	12
1.3. Сучасні персональні комп'ютери	17
1.4. Програмне забезпечення ПК	31
Розділ 2. Основні відомості про операційну систему Windows	35
2.1. Основи операційних систем	35
2.2. Файлова система операційних систем	38
2.3. Windows XP	43
2.4. Основні стандартні програми Windows	55
Розділ 3. Сервісне програмне забезпечення	67
3.1. Файлові менеджери	67
3.2. Методи роботи із стиснутими даними	69
3.3. Віруси та антивірусне програмне забезпечення	72
3.4. Мультимедіа-програми	78
Розділ 4. Офісне програмне забезпечення	82
4.1. Пакет програм MS Office	82
4.2. MS Word	83
4.3. MS Excel	121
4.4. MS PowerPoint	148
4.5. MS Access	159
Розділ 5. Основи алгоритмізації та програмування	210
5.1. Основи алгоритмізації	210
5.2. Основи програмування мовою Visual Basic 6.0	215
5.3. Діалогові вікна	233
5.4. Робота з файлами	236
5.5. Користувацькі процедури та функції	241
5.6. Виведення інформації	246



5.7. Робота з графікою	249
5.8. Робота з мишею та клавіатурою	252
5.9. Середовище розробки програм Visual Basic 6.0	255
5.10. Програмування деяких алгоритмів	263
5.11. Основні поняття об'єктно-орієнтованого програмування	269
Розділ 6. Комп'ютерні мережі	273
6.1. Історія створення та склад мережі	273
6.2. Призначення та принципи побудови комп'ютерних мереж	277
6.3. Безпека роботи в комп'ютерних мережах	281
6.4. Мережа Internet	282
Додатки	296
Література	303
Алфавітний покажчик термінів	305





ПЕРЕДМОВА

Підручник “Інформатика та комп’ютерна техніка” призначено для базової підготовки бакалаврів напряму “Гідротехніка (водні ресурси)” відповідно до вимог державного стандарту освіти. Його завдання – сформулювати фундамент інформаційних знань і основних практичних навичок, який в подальшому полегшить вивчення пакетів прикладних програм чи нових програмних засобів. З іншого боку, знання, одержані при вивченні інформатики, набудуть практичного змісту в дисциплінах спеціального циклу, а досвід, здобутий при вивченні фахових дисциплін з використанням інформаційних технологій, буде використаний при переході випускника до професійної діяльності.

Зміст курсу інформатики змінюється швидкими темпами. Тому програма викладання комп’ютерних технологій повинна перебувати у тісній взаємодії з навчальними програмами загальноосвітніх і спеціальних дисциплін.

Пропонуючи різноманітний комплекс засобів, прийомів і технологій розв’язування задач проектування, будівництва і експлуатації водогосподарських об’єктів, інформатика створює умови для реалізації єдиного методичного підходу до вивчення всіх дисциплін навчального плану, що сприяє системності навчання.

Підручник містить 6 розділів, в яких, на думку авторів, достатньо повно наведені відомості про сучасний стан та історію розвитку апаратних і програмних засобів комп’ютерної техніки. Зокрема, розділи 1,3,6 складають методичну базу як для вивчення інформатики, так і для здійснення міжпредметних зв’язків. Розділи 2,4,5 створюють технологічну основу для взаємодії інформатики з іншими дисциплінами та для професійної підготовки майбутнього фахівця. В кінці кожного розділу вміщено запитання і завдання для самостійної роботи та повторення вивченого матеріалу.

Книга буде корисною студентам технічних спеціальностей вищих навчальних закладів, викладачам дисциплін, в яких використовуються комп’ютерні технології, а також спеціалістам-практикам, які самостійно освоюють комп’ютерну техніку.



РОЗДІЛ 1. ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1 Поняття інформації та інформатики

Діяльність людини передбачає пізнання оточуючого світу. Об'єкти і процеси, що вивчаються, є джерелами інформації, яка поширюється у формі повідомлень (сигналів), наприклад, світлових, звукових, теплових тощо. Зареєстровані сигнали називають даними. Інформація (лат. informatio – роз'яснення, виклад) – це продукт взаємодії даних і відповідних їм методів.

Інформація – сукупність відомостей (даних), які приймають із навколишнього середовища, передають у навколишнє середовище або зберігають всередині певної системи.

Іншими словами, інформація – це відомості, які передаються людьми усним, письмовим способом або з допомогою технічних засобів. Сьогодні під інформацією розуміють обмін відомостями не лише між людьми, а й між людиною і пристроєм, між пристроєм і пристроєм, обмін сигналами у тваринному і рослинному світі, передачу ознак від організму до організму, від клітини до клітини.

Основи математичної теорії інформації у 1911 році розробив Клод Шеннон. Зокрема, кількість інформації запропоновано вимірювати співвідношенням:

$$I = \log_2 Q,$$

де Q – кількість інформаційних елементів (сигналів, знаків тощо). Одиницею вимірювання кількості інформації є біт (bit).

1 біт – кількість інформації, яка може бути збережена в елементарній комірці пам'яті з двома стійкими станами або може бути передана одним каналом за найменший момент часу.

В технічних системах два можливих стани кодуються двома цифрами: 0 і 1. Цифри 0 і 1 є символами **двійкової системи числення**. У кожній цифрі (знаку) двійкового числа міститься 1 біт інформації. Щоб оцінити більший обсяг інформації, використовують похідні від біта одиниці, зокрема:

$$1 \text{ Кілобіт (Кбіт)} = 2^{10} \text{ біт} = 1024 \text{ біт};$$

$$1 \text{ Мегабіт (Мбіт)} = 2^{10} \text{ Кбіт} = 1024 \text{ Кбіт}.$$

Для вимірювання обсягів інформації в обчислювальній техніці використовується спеціальна одиниця **1 байт** (byte), який становить



групу із 8 біт. 1 байт відповідає восьми розрядам двійкового числа, які використовуються як єдине ціле. Він може опрацюватися 8-розрядним логічним пристроєм. В одному байті можна закодувати значення одного із $2^8 = 256$ символів.

Одним байтом, як правило, кодується тільки один символ текстової інформації. Група із двох байтів (16 бітів), які опрацюються одночасно, називається **словом**. Відповідно, група із чотирьох взаємозв'язаних байтів (32 розряди) називається **подвійним словом**.

Вимірювання великих обсягів інформації відбувається з використанням похідних від байта одиниць:

$$1 \text{ Кілобайт (Кб)} = 2^{10} \text{ байт} = 1024 \text{ байт};$$

$$1 \text{ Мегабайт (Мб)} = 2^{10} \text{ Кб} = 1024 \text{ Кб};$$

$$1 \text{ Гігабайт (Гб)} = 2^{10} \text{ Мб} = 1024 \text{ Мб};$$

$$1 \text{ Терабайт (Тб)} = 2^{10} \text{ Гб} = 1024 \text{ Гб}.$$

Наприклад, обсяг 3,5-дюймової дискети становить 1,44 Мб; лазерного компакт-диска (CD) – сотні мегабайт; сучасних жорстких дисків – сотні гігабайт.

Основною одиницею зберігання даних є **файл**. Файл являє собою послідовність байтів, яка має власне ім'я. В окремому файлі, як правило, зберігаються дані однакового типу. Тип файлу визначається типом даних, які в ньому зберігаються.

На прийняття рішень можуть суттєво впливати властивості інформації, серед яких слід відмітити об'єктивність, повноту, вірогідність, доступність, адекватність та актуальність.

Неухильне ускладнення умов управління виробництвом, а також швидке впровадження нових носіїв інформації, вимагає структурних перетворень даних. До найбільш поширених операцій з даними належать:

- нагромадження і формалізація даних;
- фільтрація і сортування даних;
- архівування і захист інформації;
- обмін даними в системі “клієнт - сервер”;
- перетворення даних з однієї форми в іншу.

Автоматизація роботи з даними різних типів вимагає уніфікації форми їх подання. Оскільки комп'ютер може працювати лише з інформацією, яка подана в числовій формі, всі дані (символи, звуки, зображення) для опрацювання комп'ютером повинні бути

перетворені в числову форму. Для цього використовують кодування даних. При виведенні на зовнішні пристрої (екран, принтер) за числовим кодом будується зображення відповідних символів. В обчислювальній техніці використовуються **двійкові коди**, які передбачають подання даних послідовністю двох цифр двійкової системи: 0 і 1. Щоб одержати код цілого чи дійсного числа, необхідно записати це число у двійковій системі числення.

Для переведення цілого десяткового числа у двійкову систему необхідно послідовно ділити це число на 2, фіксуєючи остачі і частки на кожному кроці ділення до тих пір, поки остання частика стане меншою від 2. Число, одержане в двійковій системі, складається із останньої частки та остач від ділення на кожному кроці, записаних послідовно, починаючи з останньої частки.

Приклад. Перевести ціле десяткове число $N_{(10)}=83$ в двійкову систему числення та виконати перевірку.

Для розв'язування задачі частки і остачі, отримані при послідовному діленні числа $N_{(10)}$ на 2, запишемо у вигляді стовпчиків. Порядок запису числа в системі з основою 2 показаний стрілкою.

83	1	↑
41	1	
20	0	
10	0	
5	1	
2	0	
1		

Таким чином, $N_{(2)}=1010011$.
Виконаємо перевірку

$$N_{(2)}=1010011=1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 64 + 16 + 2 + 1 = 83_{(10)}$$

Правила виконання арифметичних дій над двійковими числами задаються таблицями додавання, віднімання і множення:

додавання	віднімання	множення
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 * 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 * 0 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 * 1 = 1$



Зауважимо, що одиниця в результаті останньої суми в таблиці додавання переходить у старший розряд. В останній різниці таблиці віднімання одиниця береться із старшого розряду і перетворюється в дві одиниці молодшого розряду.

Наведемо приклади арифметичних операцій над двійковими числами і відповідними їм десятковими числами:

11101,011	29,375	11110,011	30,375
+ 10101,001	+ 21,125	- 10111,101	- 23,625
110010,100	50,500	110,110	6,750

$$\begin{array}{r}
 101,01 \\
 \times 11,01 \\
 \hline
 10101 \\
 00000 \\
 10101 \\
 10101 \\
 \hline
 10001,0001
 \end{array}$$

$$5,25 \times 3,25 = 17,0625 \quad 50 : 10 = 5$$

$$\begin{array}{r}
 110010 \\
 - 1010 \\
 \hline
 001010 \\
 - 1010 \\
 \hline
 0000
 \end{array}
 \quad \left| \begin{array}{r}
 1010 \\
 - 101 \\
 \hline
 \end{array} \right.$$

При введенні даних з клавіатури користувач здійснює набір чисел у десятковій системі числення. Усі подальші перетворення чисел (кодування та опрацювання) здійснюються автоматично. Кожному символу алфавіту поставлене у відповідність ціле число. Це дозволяє з допомогою двійкової системи кодувати і текстову інформацію.

Різними комбінаціями нулів і одиниць у восьми двійкових розрядах, які складають 1 байт, можна подати всі великі і малі символи як англійського алфавіту, так і кирилиці, а також цифри, розділові знаки, знаки арифметичних операцій та допоміжні символи.

Останнім часом зроблені спроби створити універсальні таблиці кодування символів. Стандартизована таблиця кодів ASCII (додаток 1.1) якраз і використовує кодування символів восьмибітовими послідовностями. Перша половина цієї таблиці (коди з 0 до 127) загальноприйнята у всьому світі, тому її називають **базовою**. З коду 0 по код 31 у базовій таблиці розміщені коди управління апаратними

засобами. З 32-го по 127-й код розміщені коди символів англійського алфавіту, розділових знаків, цифр, знаків арифметичних дій і деяких спеціальних символів. Розширена частина (коди з 128 по 255) містить, як правило, коди символів національного алфавіту, псевдографіки тощо.

Перехід до двобайтової системи кодування (16 біт) значно збільшує діапазон можливих значень кодів. Для 16-розрядного кодування символів розроблена система UNICODE, яка дістала назву **універсальної**. Шістнадцять розрядів дає змогу закодувати 65536 різних символів. Цього достатньо для кодування символів більшості національних алфавітів.

При збільшенні графічного друкованого зображення можна помітити, що воно складається з дрібних точок, які утворюють певний орнамент, який називають **растром**. Координати і властивості кожної точки можна виразити цілими числами. Відтак можна стверджувати, що в растровій системі для подання графічної інформації можна використовувати двійковий код. Як правило, чорно-білі зображення являють собою комбінації точок, кожна з яких може набувати одного із 256 відтінків сірого кольору. Таким чином, для кодування яскравості будь-якої точки достатньо восьмирозрядного двійкового числа.

Для кодування кольорових зображень будь-який колір розбивається на основні складові. В якості складових використовують три основні кольори: червоний, синій, зелений. Вважається, що будь-який видимий колір можна отримати змішуванням цих трьох кольорів. Якщо для кодування яскравості кожної із основних складових використати 8 двійкових розрядів, то на кодування кольору однієї точки необхідно 24 розряди. Така система кодування дозволить отримати понад 16 млн. різних відтінків кольорів. Щоб використати кольорову графіку із 4-х кольорів, необхідно мати 32 двійкових розряди.

***Інформатика** – галузь науки, що досліджує структуру і властивості інформації, систематизує прийоми створення, зберігання, перетворення, використання, опрацювання і передачі даних засобами комп'ютерної техніки, а також принципи функціонування цих засобів.*

Серед основних понять інформатики насамперед виділяють апаратне та програмне забезпечення засобів обчислювальної техніки.

Важливою є проблема взаємодії, яка дістала назву **інтерфейс**. Зокрема, засоби взаємодії апаратного і програмного забезпечення назвали апаратно-програмним інтерфейсом. Методи взаємодії людини з апаратними і програмними засобами називають користувачьким інтерфейсом.

Основним завданням інформатики є систематизація методів роботи з апаратними і програмними засобами з метою підвищення їх ефективності. Для цього постійно удосконалюються архітектура та інтерфейси обчислювальних систем, розробляються нові прийоми і методи розробки комп'ютерних програм, захисту інформації, перетворення даних тощо.

Слово **інформатика** походить від французького терміну *informatique*, який набув поширення в 60 – х роках 20 століття. Поряд з цим, в більшості англійських країн використовується термін *Computer Science* (комп'ютерна наука).

Однією із попередниць інформатики вважається кібернетика (грецьке *kybernetikos* – мистецтво управління). Основи кібернетики були сформульовані в працях американського математика Норберта Вінера, опублікованих у 1948 році. В наш час кібернетика досліджує принципи побудови і функціонування систем автоматизованого управління, зокрема, методи моделювання процесу прийняття рішень технічними засобами, розробку принципів і методів штучного інтелекту. В багатьох питаннях сучасна кібернетика перебуває у тісній взаємодії з інформатикою.

Інформатика як наука за рахунок соціально організованої діяльності розробляє методи створення та функціонування різноманітних інформаційних систем, які базуються на досягненнях комп'ютерної техніки. Цим самим вона створює **інформаційну технологію**, яка швидко розвивається і охоплює всі види суспільної діяльності: виробництво, науку, управління, освіту, проектування, банківську справу, торгівлю, медицину, криміналістику, охорону довкілля, побут тощо. Вперше поняття і перспективи розвитку інформаційної технології детально розробив академік АН України В.М. Глушков (1923 – 1982).

Інформаційне суспільство – це суспільство, яке в усіх сферах діяльності використовує комп'ютерні системи та інші засоби збереження, обробки і передачі інформації, що дають можливість моделювати реальні процеси і явища, швидко виконувати значні



обсяги обчислень та перетворення інформації. В сучасному суспільстві інформація стала важливим економічним, соціальним і політичним ресурсом.

Для фахівця, який готує себе до виробничої діяльності, комп'ютер є знаряддям інтелектуальної праці. Тому він повинен оволодіти системою знань, умінь і практичних навичок роботи з комп'ютерною технікою, засвоїти загальні принципи побудови і функціонування персонального комп'ютера та його основних елементів. Сучасний фахівець повинен володіти методами алгоритмізації та програмування, вміти використовувати прикладні програми, призначені для роботи з текстом, графічними об'єктами, електронними таблицями, базами даних. Все це забезпечить ефективне розв'язування виробничих, проектних, економіко-управлінських, інженерних, наукових і спеціальних задач.

1.2 Обчислювальні машини

Серед перших засобів автоматизації робіт з даними, як правило, називають пристрої-суматори. Одним із таких механічних калькуляторів для виконання дій додавання і віднімання вважається пристрій, який у 1642 році розробив французький вчений Блез Паскаль (1623 - 1662). У 1673 році німецький математик і філософ Г.В. Лейбніц (1646 - 1717) побудував пристрій для виконання 4-х арифметичних дій. Принцип роботи цього пристрою був використаний для створення арифмометра, який широко використовувався в 20-му столітті до винайдення калькулятора.

В 30-х роках 19-го століття англійський математик і винахідник Чарльз Беббідж (1792 - 1871) намагався створити універсальний обчислювальний пристрій – **аналітичну машину**. В основу роботи цього механічного комп'ютера був покладений програмний принцип. Програму планувалося вводити з використанням набору перфокарт, кожна з яких містила команду, подану в закодованому вигляді. Для запису інформації існував спеціальний механічний пристрій.

В аналітичній машині вперше був реалізований принцип поділу інформації на команди і дані. Машина повинна була виконувати арифметичні операції з запам'ятовуванням проміжних і остаточних результатів. На жаль, з ряду причин задум Беббіджа не був доведений



до завершення. Однак він сформулював чимало блискучих ідей, які і донині не втратили своєї актуальності.

Слово “комп’ютер” означає “обчислювач”, тобто пристрій для виконання розрахунків. Перші комп’ютери створювалися як автоматичні арифмометри, основним завданням яких була автоматизація обчислень. Пізніше виявилось, що будь-яка інформація може бути подана в числовій формі, а відтак опрацьована на комп’ютері.

В сучасному розумінні **комп’ютер** – універсальний засіб для обробки інформації будь-якого виду. Його призначення – приймати, опрацьовувати, зберігати і видавати інформацію за заданими командами.

Ідея створення автоматичних машин, які здатні виконувати арифметичні і логічні операції при значній швидкості розв’язування задач, була реалізована лише в 1943 р., коли в Гарвардському університеті була створена обчислювальна машина “Марк-1”. Машина була виготовлена фірмою ІВМ на основі електромеханічних елементів. Недоліками моделі були мала швидкість та ненадійність.

Перша ЕОМ на основі електронних ламп ENIAC була створена в 1946 р. під керівництвом П. Еккерта і Д. Моучлі. Основний недолік машини – довгий і складний процес налагодження програми.

У 1950 р. в Україні під керівництвом С.О.Лебедева (1902 - 1974) був створений перший комп’ютер під назвою МЭСМ (рос. малая электронная счетная машина). Він втілював кращі досягнення того часу в галузі конструювання електронних машин. У 1952 р. в Москві завершена наступна модифікація цієї машини – БЭСМ (рос. быстродействующая электронная счетная машина). Розроблені в кінці 50-х років серії БЭСМ-5 і БЭСМ-6 багато в чому випередили тогочасні західні моделі.

Починаючи із середини 40-х років 20-го століття у зв’язку з розробкою великих наукових проектів, в тому числі і у військовій галузі, значно зросли обсяги обчислювальних робіт. Розвиток комп’ютерної техніки став одним із основних факторів, які визначали темпи розвитку науково-технічного прогресу.

Комп’ютери **першого покоління**, які створювалися в цей час на основі електронних ламп, мали великі габарити, високу собівартість та труднощі в програмуванні і обслуговуванні. Кожна різновидність ЕОМ мала своє програмне забезпечення. Програмна несумісність

Национальный университет
та державного управління

гальмувала розробку великих інформаційних систем. Тому в 60-х роках почався перехід до створення ЕОМ із мікроелектронною технологічною базою. Поряд із розвитком електроніки удосконалювалося і ставало більш універсальним програмне забезпечення. На зміну програмам в машинних кодах розроблені мови програмування високого рівня, наближені до людської мови.

Поява напівпровідникових елементів дозволила зменшити розміри та масу електронної техніки, а також затрати енергії. Логіка ЕОМ **другого покоління** будувалася із застосуванням транзисторів та діодів. Використання напівпровідникових елементів в електронних схемах суттєво збільшило надійність ЕОМ, а обсяг оперативної пам'яті та швидкодія зросли в десятки і сотні разів.

Комп'ютери другого покоління значно розширили сферу використання електронної обчислювальної техніки, в першу чергу, за рахунок розвитку програмного забезпечення. З'явилися також спеціалізовані машини, наприклад, для управління виробничими процесами та системами передачі інформації, для розв'язування економічних задач тощо.

В Україні були створені напівпровідникові ЕОМ “Промінь”, “Мир”, “Мир-2”, які широко використовувались для розв'язування інженерних задач, наукової роботи та навчання.

У 1954 р. Р. Нойсом був винайдений спосіб, який на мінімальній площі дозволяв розміщення декількох транзисторів і необхідних з'єднань між ними, що давало можливість на одному кристалі кремнію розмістити від 10 до 1000 логічних елементів. Отримані електронні схеми назвали інтегральними схемами або чіпами. **Інтегральна схема** – це логічно завершений функціональний блок, який виконує функції достатньо складної транзисторної схеми. Широке застосування інтегральних схем сприяло створенню ЕОМ **третього покоління**.

Досягнення мікроелектроніки суттєво покращили техніко-експлуатаційні характеристики комп'ютерів: збільшився обсяг оперативної пам'яті, зросла швидкодія, підвищилася надійність, а енергозатрати, робоча площа і маса пристроїв зменшилися.

У 70-х роках 20-го століття в нашій країні серед найбільш поширених комп'ютерів третього покоління було сімейство ЕОМ єдиної системи. Ця серія машин була успішною міждержавною розробкою за участю шести країн Східної Європи. Сімейство



програмно-сумісних машин мало однакову елементну базу, конструктивно-технологічну основу, однакову систему математичного забезпечення та уніфікований набір периферійних пристроїв.

Важливою особливістю комп'ютерів третього покоління, яка вигідно відрізняє їх від попередників, є можливість паралельної (автономної) роботи окремих пристроїв. Це дозволяє, на відміну від однопрограмних машин, одночасно виконувати декілька програм, реалізуючи принцип мультипрограмування. Завдяки цьому принципу стала можливою робота в режимі роздільного часу та в режимі діалогу. Декілька користувачів, навіть віддалених від ЕОМ, спілкувалися з нею незалежно один від одного, що дозволило збільшити ефективність використання апаратної частини.

Одна із головних ознак комп'ютерів третього покоління – наявність розвинутої операційної системи. **Операційна система** – комплекс програмних засобів, які забезпечують автоматизацію підготовки і виконання програм та оптимізацію функціонування апаратних засобів (детальніше див. розділ 2).

Потужне математичне забезпечення ЕОМ третього покоління, зокрема, пакети прикладних програм, створили умови для якісного розв'язування багатьох інженерно-технічних, наукових, економічних задач та розробки автоматизованих систем управління.

Четверте покоління ЕОМ базувалося на використанні великих інтегральних схем. Високий рівень інтеграції сприяв збільшенню компактності електронної апаратури, підвищенню її надійності, завдяки чому у 80-х роках знизилась вартість ЕОМ, а їх швидкодія досягла десятків мільйонів операцій за секунду.

Поряд із вдосконаленням математичного забезпечення і апаратної частини обчислювальних систем четвертого покоління вчені і конструктори світу, починаючи з кінця 20-го століття, займаються розв'язанням проблем розробки основних концепцій **ЕОМ п'ятого покоління**, серед яких:

- підвищення інтелектуальності комп'ютерів для забезпечення їх спілкування з людиною на рівні ідей, образів і звичайної мови;
- розробка комп'ютерів, які могли б працювати в недосліджених галузях науки і техніки над розв'язанням поставлених перед ними проблем і завдань;



створення комп'ютерних експертних систем в конкретних галузях науки і техніки, які на основі бази знань (ідей) здатні робити логічні висновки і приймати самостійні рішення.

Якщо систематизувати ЕОМ за найбільш характерними ознаками, то можна встановити декілька способів їх класифікації.

За принципом дії ЕОМ поділяють на аналогові і цифрові. Аналогові (машини неперервної дії) оперують неперервними величинами (функціями), які характеризують певний фізичний процес. Незалежно від фізичної природи процесу, який досліджується, аналогові машини здійснюють операції над електричними величинами, які є аналогами параметрів неелектричної природи, з використанням прийомів і засобів математичного моделювання. Цифрові машини оперують цифрами або їх наборами (словами). Вони набули широкого поширення, оскільки більш універсальні і точні.

За призначенням ЕОМ поділяють на універсальні (загального призначення), керуючі, проблемно-орієнтовані, інформаційні, спеціальні.

За масштабом завдань, які виконуються, комп'ютери поділяють на великі ЕОМ, міні-ЕОМ, мікро-ЕОМ та персональні комп'ютери. Ця класифікація більшою мірою стосується цифрових машин.

Великі ЕОМ розв'язують задачі управління, наприклад, для великих водогосподарських організацій, утворюючи інформаційну систему.

Міні-ЕОМ призначені для управління локальними технологічними процесами, зокрема, забезпечення неперервної подачі води на об'єкт, роботи установки чи групи установок. Ці ж машини можуть виконувати роль робочих станцій в інформаційній системі.

Тривалий час розвиток ЕОМ передбачав створення все більших і більших обчислювальних систем складної архітектури. Однак в середині 70-х років завдяки мікромініатюризації електронних схем вдалося створити мікро-ЕОМ, а згодом **персональний комп'ютер** (ПК), де основним елементом є мікропроцесор.

***Мікропроцесор** – завершений цифровий обчислювальний пристрій, який виконує обробку введеної в нього інформації за алгоритмами різної складності.*

ПК може розв'язувати задачі управління окремими об'єктами, що входять до складу системи управління технологічним процесом, а також розв'язувати задачі звичайної інженерної практики.

Незважаючи на невеликі розміри, сучасні ПК мають високу продуктивність і за основними характеристиками перевершують великі ЕОМ 70-х років. Особливо широкої популярності персональні комп'ютери набули у зв'язку із розвитком інформаційних мереж.

1.3. Сучасні персональні комп'ютери

Початком відліку нової комп'ютерної ери можна вважати 1971 рік, коли під керівництвом Теда Хоффа був створений перший мікропроцесор Intel 4004. Це зробило можливим у 1975 р. появу першого персонального комп'ютера Altair-8800 на основі мікропроцесора Intel-8080. Успішна реалізація цього проекту сприяла появі багатьох фірм, які спеціалізувалися на виробництві комп'ютерів, випуску відповідної літератури, розробці програмного забезпечення (текстові і графічні редактори, табличні процесори та інше). В 1977 р. розробила персональний комп'ютер фірма Apple. У 1979 р. фірма IBM приступила до розробки своєї моделі ПК. Бурхливе поширення персональних комп'ютерів у кінці 70-х років привело до зниження попиту на великі і міні-комп'ютери. Адже персональний комп'ютер – пристрій, який виконує чимало функцій великого комп'ютера, але розрахований на одного користувача.

У 1981 р. завершилась робота над ПК IBM PC. Він був побудований на основі 16-розрядного процесора Intel-8088 з дотриманням принципу відкритої архітектури, що сприяло його неабиякій популярності. IBM PC став стандартом, якого почали дотримуватися розробники апаратних засобів та програмного забезпечення.

Відкритість архітектури полягає в тому, що комп'ютер не є нероздільним, а складається з окремих частин, які виробляються незалежними постачальниками. Набір основних і додаткових пристроїв можна легко змінювати в залежності від специфіки використання ПК чи завдань, які необхідно виконувати.

Надалі ми зосередимо увагу на вивченні програмних продуктів, які дозволяють розв'язання широкого кола інженерно-технічних задач з використанням IBM PC-сумісних комп'ютерів.

Принципова схема будови комп'ютера була запропонована американським математиком Джоном фон Нейманом у 1945 році. Ці

принципи лягли в основу розробки персональних комп'ютерів (рис. 1.2). За схемою Неймана комп'ютер складається із:

- **арифметико-логічного пристрою (АЛП)**, призначеного для виконання арифметичних і логічних операцій;
- **пристрою управління (ПУ)**, який організовує процес виконання програми;
- **пам'яті** (запам'ятовуючих пристроїв) для зберігання програм та даних;
- **зовнішніх пристроїв** для введення-виведення інформації.

В сучасних комп'ютерах функції арифметико-логічного пристрою та пристрою управління виконує один пристрій – процесор (рис. 1.1).



Рис. 1.1

***Процесор** – пристрій, призначений для безпосереднього виконання операцій над інформацією та управління роботою всього комп'ютера.*

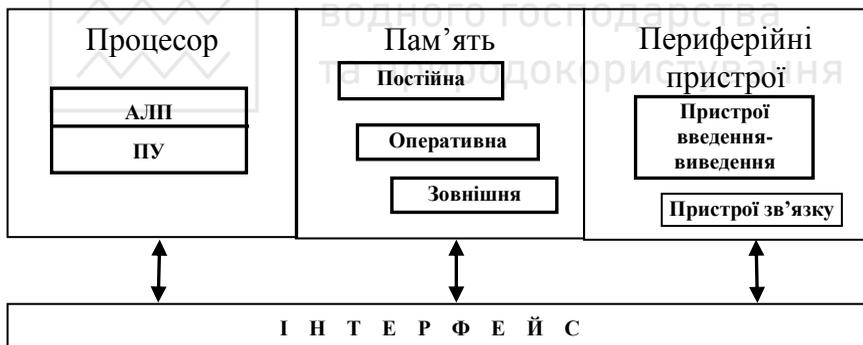


Рис. 1.2. Структурна схема ПК

Персональний комп'ютер може працювати з різним набором зовнішніх пристроїв. Конфігурацію ПК (склад обладнання) можна змінювати в залежності від потреб користувача. **Базова конфігурація** ПК передбачає мінімальний набір пристроїв, які забезпечують функціонування комп'ютера. До них належать:

- системний блок;
- монітор;
- клавіатура;



Системний блок об'єднує в собі найбільш важливі компоненти ПК. Пристрої, розміщені всередині системного блока, умовно називають **внутрішніми**, а пристрої, які під'єднуються до нього – **зовнішніми**. Додаткові зовнішні пристрої, призначені для введення, виведення і тривалого зберігання даних, називають ще **периферійними**.

Одним із головних пристроїв візуального подання даних є монітор (рис. 1.3). Виведення інформації на екран здійснюється в текстовому або графічному режимі. У текстовому режимі екран поділений на 25 рядків по 80 символів (позицій). У графічному режимі екран складається з точок (пікселів). Кількість точок по горизонталі та вертикалі, які монітор здатний



Рис. 1.3

якісно відтворити, називається його **роздільною здатністю**. Залежно від типу монітора та відеокарти, на екрані може виводитися 640 на 480, 800 на 600 або більше точок. Кожна точка може відобразитися одним із 16, 256 або більше кольорів.

Розмір монітора визначається відстанню в дюймах (1 дюйм=1"=2,54 см) між протилежними кутами екрана кінескопа. Стандартні розміри діагоналі монітора: 15", 17", 19", 20", 21". Для роботи з графікою бажано використовувати монітор розміром не менше 19 – 24 дюймів.

Важливою характеристикою монітора є **частота регенерації** (поновлення) зображення. Вона показує, скільки разів за секунду монітор може повністю змінити зображення. Частота зміни кадрів вимірюється в герцах (Гц). Чим більша частота, тим якісніше і стійкіше зображення на моніторі. Мінімальним значенням частоти кадрів вважається 75 Гц, нормативним – 85 Гц, а комфортним – понад 100 Гц.

Багатьма параметрами зображення, отриманого на екрані монітора, можна керувати з допомогою програмних засобів, які входять до складу операційної системи комп'ютера (див. розділ 2).

1.3.1. Клавіатура ПК

Клавіатура є основним пристроєм введення знакових даних. Вона використовується для введення в комп'ютер інформації та команд керування.



Клавіатура стаціонарного ПК є окремим блоком. У переносних ПК клавіатура вмонтована в корпус. Комбінація клавіатури і монітора забезпечує найпростіший **інтерфейс користувача**.

Стандартна клавіатура містить, як правило, 104 клавіші, які умовно можна розбити на такі групи:

- функціональні клавіші;
- алфавітно-цифрові (символьні) клавіші;
- клавіші керування та допоміжні клавіші;
- додаткові цифрові клавіші.

Функціональні клавіші **F1**, **F2**,..., **F12** розміщені у верхньому рядку клавіатури. Найчастіше ці клавіші використовуються для того, щоб подати команду чи вказати на конкретну дію (функцію), яку повинен виконати комп'ютер. Призначення кожної із функціональних клавіш визначається програмою, з якою в даний момент працює користувач. Однак в більшості програм клавіша **F1** викликає систему допомоги (довідку), яка містить додаткову інформацію про конкретний режим роботи чи програмний продукт.

Алфавітно-цифрові клавіші (лівий нижній блок клавіш) призначені для введення окремих символів, а також команд, які набираються буквами. Кожна клавіша може працювати в двох режимах (регістрах), що дозволяє використовувати її для введення одного чи двох символів.

Крім символьних клавіш у цій частині клавіатури розміщені клавіші керування, натискування на які (окремо або в комбінації з іншими клавішами) змінює значення інших клавіш.

|| *Клавіші керування призначені для зміни значень інших клавіш.*

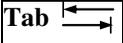
Наприклад, перемикання з нижнього регістра (введення малих букв) на верхній регістр (введення великих букв) виконують утримуванням клавіші **Shift** (англ. shift - зсув). Таке перемикання є нефіксованим. Якщо треба набрати багато великих літер, перехід на верхній регістр можна зафіксувати клавішею—перемикачем **Caps Lock**. Після одноразового натискання на клавішу **Caps Lock** всі клавіші, які ми будемо натискати надалі, крім клавіш верхнього цифрового ряду, будуть клавішами верхнього регістра. Повторне натискання на клавішу **Caps Lock** переведе клавіатуру на нижній регістр. До клавіш керування належать також клавіші **Ctrl** (англ. to control - керувати) та **Alt** (англ. to alternate - змінювати).



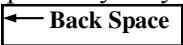
Якщо з клавіатури вводяться дані, в кінці рядка натискають клавішу **Enter** (англ. to enter - входити). При цьому введення інформації почнеться з нового рядка. Якщо ж клавіатуру використовують для введення команд, натисканням клавіші **Enter** завершується введення команди і починається її виконання.

Символи українського чи російського алфавіту (кирилиця) та символи латинського алфавіту закріплені за алфавітно-цифровими клавішами за відповідними схемами. Для переходу з кирилиці на латинський алфавіт у Windows XP використовують комбінацію клавіш **Ctrl + Shift**, яка задається програмним способом.

Клавіша **Esc** (англ. to escape – виходити) зазвичай використовується для виходу із поточного режиму роботи комп'ютера.

Клавіша  називається клавішею **табуляції**. При введенні тексту вона використовується для швидкого пересування курсора на задане число позицій вперед. **Курсор** – це екранний елемент, який вказує місце введення символу. Для стандартного режиму крок табуляції складає 8 позицій. В інших задачах клавіша **Tab** може виконувати інші функції.

При введенні даних використовується символ **пробілу** – одна порожня позиція. Слова відокремлюються одне від одного одним чи кількома пробілами. Клавіша пробілу (англ. Space bar) розміщена в нижньому рядку символної клавіатури. При натисненні клавіші пробілу курсор зміщується на одну позицію праворуч.

В правому верхньому куті алфавітно-цифрової клавіатури розміщена клавіша . Натиснення на цю клавішу вилучає останній набраний символ і переміщує курсор на одну позицію ліворуч, тобто на позицію вилученого символу.

Справа від групи функціональних клавіш розміщені клавіші **Print Screen**, **Scroll Lock** і **Pause/Break**, які виконують специфічні функції, що залежать від активного в даний момент програмного продукту. Загальноприйнятими вважаються такі дії:

Print Screen (англ. to print screen - друкувати екран) - збереження поточного зображення, розміщеного на екрані, в **буфері обміну** (спеціальна область оперативної пам'яті). Натиснення комбінації клавіш **Shift+Print Screen** дозволяє надрукувати копію екрана на принтері, якщо він підключений до комп'ютера і готовий до роботи.



Scroll Lock – переключення в деяких задачах режиму переміщення інформації на екрані. Такий режим називають **скролінгом** (англ. scrolling - переміщення, прокрутка).

Pause/Break (англ. pause - пауза) – призупинення/переривання роботи комп'ютера. Припинити паузу (продовжити роботу) можна натисканням на будь-яку іншу клавішу.

Натиснення комбінації клавіш **Ctrl+Break** перериває виконання поточної програми. Те саме відбувається і при натисненні комбінації клавіш **Ctrl+C**. Для припинення виконання програми використовується комбінація клавіш **Ctrl+S**. Щоб продовжити роботу комп'ютера після паузи, що виникла, слід натиснути на клавіші **Ctrl+S** ще раз.

Клавіші з позначеннями \uparrow , \downarrow , \leftarrow , \rightarrow , **Home**, **End**, **PgUp**, **PgDn**, називають **клавішами керування курсором**. Вони дозволяють керувати позицією введення. Натиснення на клавішу із стрілкою викликає зміщення курсора на одну позицію у вказаному напрямку.

Клавіша **Home** (англ. home – дім) переводить курсор на початок поточного рядка, а клавіша **End** (англ. end – кінець) – у його кінець. Клавіші **PgUp** і **PgDn** (англ. Page Up – сторінка вгору, Page Down – сторінка вниз) переміщують курсор на одну екранну сторінку вгору чи вниз. **Екранною сторінкою** є фрагмент документа, що знаходиться на екрані (як правило, 25 рядків тексту).

Традиційно клавіша **Insert** (англ. to insert – вставляти) є клавішею-перемикачем режимів вставки та заміни. У режимі **вставки** символ, що вводиться, вставляється в рядок, а частина рядка праворуч від курсора разом із введеними раніше символами зміщується на одну позицію праворуч. У режимі **заміни** символ у позиції курсора замінюється на введений символ без будь-якого пересування інших символів, що розміщені на екрані.

Клавіша **Delete** (англ. to delete – стирати, знищувати) призначена для видалення символу в позиції курсора. При цьому положення позиції введення залишається незмінним (на відміну від дії клавіші **Backspace**, яка позицію введення зміщує ліворуч).

Додаткові цифрові клавіші дублюють дію цифрових та деяких знакових клавіш основної частини панелі, а також дію клавіш керування курсором. Клавіша **Num (Num Lock)**, яка розміщена у верхньому ряду, є клавішею-перемикачем між режимом введення цифр і режимом керування курсором. На стан перемикача **Num Lock**

та перемикачів **Caps Lock** і **Scroll Lock** вказують світлові індикатори, розміщені в правому верхньому куті клавіатури. Зокрема, якщо індикатор **Num Lock** світиться, клавіатура перебуває в режимі введення цифр.

Натиснення комбінації клавіш **Ctrl+Num Lock** створює паузу в роботі комп'ютера. Для продовження його роботи користувачеві слід натиснути на будь-яку іншу клавішу.

Комбінація клавіш **Ctrl+Alt+Del** викликає так зване **гаряче перезавантаження комп'ютера** (комп'ютер починає працювати так, ніби його щойно ввімкнули).

Якщо будь-яку клавішу тримати натиснутою тривалий час, введення відповідного символу буде автоматично повторюватися. Цю властивість клавіатури зручно використовувати, наприклад, для переміщення курсора на певну кількість позицій. Інтервал часу після натиснення, через який почнеться автоматичне повторювання символу, а також кількість повторів за секунду визначаються засобами налаштування клавіатури.

В цьому розділі описане стандартне використання клавіш клавіатури. Проте при розв'язуванні прикладних задач призначення клавіш або їх комбінацій може бути іншим.

1.3.2. Периферійні пристрої

Периферійні пристрої ПК підключаються до його інтерфейсів і призначені для виконання допоміжних дій. Завдяки периферійним пристроям комп'ютер є гнучкою та універсальною системою.

За призначенням периферійні пристрої можна поділити на наступні типи: пристрої введення даних; пристрої виведення даних; пристрої зберігання даних; пристрої обміну даними.

Пристрої введення даних

Як уже відмічалось, для введення знакових даних використовують різні типи клавіатур. Окрім звичайних, існують спеціальні клавіатури, які призначені для підвищення ефективності введення даних, зокрема, ергономічні клавіатури.

За типом підключення клавіатур до системного блоку розрізняють провідникові та безпроводникові клавіатури. В останньому випадку передача даних здійснюється за допомогою інфрачервоного променя.

Поширеним пристроєм командного управління, який полегшує роботу користувача,



Рис. 1.4

особливо при використанні графічного режиму роботи дисплея, є маніпулятор миша (рис. 1.4).

Миша – пристрій, за допомогою якого можна керувати місцезнаходженням курсора на екрані монітора і вибирати об'єкти для подальшої роботи.

Робота миші потребує підтримки спеціальної системної програми, яку називають **драйвером миші**. Драйвер встановлюють при першому підключенні миші або при встановленні операційної системи комп'ютера.

При підключенні миші до комп'ютера на екрані монітора як в текстовому, так і в графічному режимах роботи з'являється **курсор миші**. Форма курсора може змінюватися в залежності від режиму роботи. Для переміщення курсора по екрану необхідно пересунути мишу по горизонтальній поверхні.

Введення інформації в комп'ютер відбувається за допомогою кнопок, вбудованих у корпус миші. Більшість сучасних комп'ютерів використовують дво- чи трикнопові миші. Замість середньої кнопки в деяких моделях може бути розміщений ролик, який можна використовувати для прокручування зображення на моніторі.

Коротке натискання однієї із кнопок називають **клацанням**. Залежно від того, яка із кнопок натискується, і способу натискування на неї – одне клацання, подвійне клацання, натиснення і утримування кнопки в цьому положенні під час пересування миші – комп'ютер виконує різні дії. Переміщення миші і клацання її кнопок з точки зору її програми – драйвера є **подіями**. Аналізуючи ці події, драйвер передає інформацію прикладній програмі, з якою працює користувач. За цією інформацією програма може визначити команду, яку хотів реалізувати користувач, і приступити до її виконання.

Комбінація монітора і миші забезпечує сучасний тип інтерфейсу користувача, який називають **графічним**. З допомогою миші користувач може змінювати властивості графічних об'єктів і приводити в дію елементи керування комп'ютерною системою, які розміщені на екрані.

Підключення миші, як і клавіатури, до системного блоку може бути провідниковим або з використанням інфрачервоного променя.

Для введення інформації в комп'ютер може використовуватися ще один пристрій маніпуляторного типу – **трекбол** (англ. trackball, track – слід, ball - куля), який за принципом дії та функціональними

можливостями подібний до миші. На відміну від миші, трекбол встановлюється стаціонарно. Куля, що керує рухом курсора на екрані, розміщена зверху і приводиться в рух рукою. Трекбол широко використовується у портативних ПК.

Крім того, до пристроїв введення інформації належать **джойстики, світлові пера, пристрої оптичного розпізнавання символів, сенсорні екрани**. Ці пристрої використовуються при розв'язуванні спеціальних задач.

Для введення графічної інформації можуть використовуватися **сканери** (англ. to scan – пильно розглядати), **графічні планшети (дигітайзери)** і **цифрові фотокамери**.

Планшетні сканери призначені для введення графічної інформації з прозорого чи непрозорого матеріалу. Принцип дії цих пристроїв полягає в тому, що промінь світла, відбитий від поверхні матеріалу, фіксується спеціальними приладами із зарядовим зв'язком. Основні параметри сканерів: роздільна здатність, продуктивність, динамічний діапазон, максимальний розмір сканованого матеріалу. Типова роздільна здатність для офісних сканерів – 600..1200 dpi (англ. dots per inch – точок на дюйм). Для професійних сканерів цей показник становить 1200..3000 dpi. На продуктивність сканера впливає швидкість та досконалість механічного пристрою сканування і тип інтерфейсу зв'язку сканера з ПК. Динамічний діапазон визначається логарифмом відношення яскравості найбільш світлих ділянок зображення до найбільш темних (офісні 1,8..2,0, професійні від 2,5).

Ручні сканери за принципом дії нагадують планшетні. Рівномірність і точність протягування лінійки сканування залежить тут від користувача і забезпечує роздільну здатність 150..300 dpi.

У **барабанних сканерах** сканований матеріал закріплюється на барабані та провертається з великою швидкістю. Барабанні сканери забезпечують роздільну здатність 2500..5000 dpi. Їх використовують для сканування невеликих зображень (слайди, плівки та ін.).

Сканери форм використовують для сканування заповнених форм. Їх застосовують тоді, коли необхідна велика швидкість обробки зображень і немає високих вимог до якості зображення (наприклад, обробка даних перепису населення, обробка анкет тощо).

Штрих-сканери використовують для сканування штрих-кодів.

Графічні планшети (дигітайзери) призначені для введення художньої графічної інформації. В основу їх роботи покладена

фіксація переміщень графічного пера відносно планшета і міри його натискання. Дигітайзери використовуються в основному художниками та ілюстраторами.

Цифрові фотокамери, як і сканери, сприймають зображення з допомогою пристроїв із зарядовим зв'язком, які об'єднані в прямокутну матрицю. Кількість елементів матриці визначає роздільну здатність цифрових фотоапаратів, яка для якісних моделей складає понад 1 млн. точок.

Пристрої виведення даних

Для виведення інформації, окрім монітора, використовують **принтери** (англ. printer - друкувальний пристрій, рис. 1.5), які дозволяють отримати копію документа на папері, плівці чи іншому матеріалі. За принципом дії розрізняють матричні, лазерні, світлодіодні та струменеві принтери.



Рис. 1.5

В **матричних принтерах** формування зображень відбувається при ударі перпендикулярних до поверхні друкування циліндричних стержнів (голок) по паперу через фарбувальну стрічку. Якість друку залежить від кількості голок, яких у друкувальній головці може бути 9, 18 або 24. Продуктивність роботи матричних принтерів визначається кількістю надрукованих знаків за секунду.

Лазерні принтери мають високу швидкість та якість друку. Принцип дії лазерних принтерів наступний. Зображення спочатку формується на фоточутливій поверхні спеціального барабана у вигляді сукупності електричних зарядів. Для цього використовують лазерний промінь, який модулюється за командами комп'ютера. При обертанні поверхня барабана проходить через контейнер, наповнений тонкодисперсним барвником (тонером). Завдяки електростатичній взаємодії частинки барвника прилипають до наелектризованих ділянок поверхні барабана. При подальшому обертанні відбувається контакт барабана з папером, в результаті чого тонер переноситься на папір. Зображення закріплюється короткочасним прогріванням.

Роздільна здатність лазерних принтерів становить понад 600 точок на дюйм (тобто розмір точки складає $\sim 0,04$ мм). При виведенні текстів швидкість друку відповідає 5..15 с на сторінку. Для виведення сторінки з малюнками часу витрачається значно більше. До основних

параметрів лазерних принтерів, крім цього, належать формат паперу, який можна використовувати для друкування, та обсяг власної оперативної пам'яті. Вартість друкування однієї сторінки вважається найменшою в порівнянні з іншими типами принтерів.

Принцип дії **світлодіодних принтерів** схожий із принципом дії лазерних принтерів. Відмінність полягає в тому, що джерелом світла тут є лінійка світлодіодів, яка розміщена по всій ширині сторінки, що спрощує формування зображення.

Струменеві принтери формують зображення на папері з допомогою плям спеціального чорнила. Викид мікрокраплин чорнила відбувається під тиском через сопла друкувальної головки. Якість друку залежить від форми та розміру каплі чорнила, характеру всмоктування чорнила поверхнею паперу. Ці принтери широко використовуються для кольорового друку. За показником якість/ціна струменеві принтери значно переважають кольорові лазерні принтери. Струменеві принтери є набагато дешевшими від лазерних принтерів, але вартість надрукованої сторінки може бути в декілька разів вищою.

Для якісного виведення графічної інформації використовують графопобудовники, які ще мають назву плотерів. **Плотери** (англ. to plot - креслити) можуть виводити інформацію у вигляді малюнків, графіків, діаграм чи креслень. Плотери використовують для одержання складних конструкторських креслень, архітектурних планів, географічних і метеорологічних карт, ділових схем. Плотери малюють зображення за допомогою пера. До основних параметрів графопобудовників належать: розміри відтворюваного зображення, точність відтворення, кількість відтворюваних кольорів, швидкодія. За способом подачі паперу розрізняють фрикційні, планшетні та барабанні графопобудовники.

Пристрої зберігання даних

Необхідність у зовнішніх пристроях зберігання даних виникає:

- коли на ПК обробляється більше даних, ніж можна розмістити на вінчестері;
- коли дані мають велику цінність і необхідно виконувати регулярне резервне копіювання на зовнішній пристрій (копіювання даних на вінчестері не є резервним, воно створює тільки ілюзію безпеки).



Для зовнішнього зберігання даних використовують декілька типів пристроїв з магнітними чи магнітооптичними носіями.

Стримери – пристрої для зберігання даних на магнітній стрічці. Мають досить низьку продуктивність та недостатню надійність. Можуть містити декілька сотень мегабайт інформації.

Зір-накопичувачі виробляються у внутрішньому та зовнішньому виконанні. Носієм інформації є дискета, яка незначно відрізняється за розмірами від звичайної дискети і має ємність 100 або 250 Мб.

Накопичувачі HiFD дозволяють працювати як із звичайними дискетами, так і з спеціальними носіями ємністю 200 Мб.

Накопичувачі Jaz за своїми характеристиками наближаються до вінчестерів, але є змінними. В залежності від моделі накопичувач може містити 1..2 Гб інформації.

Магнітооптичні пристрої отримали широке застосування завдяки своїй універсальності. Мають відносно високу вартість як носія, так і накопичувача. Зараз існують носії з фарм-фактором 5.25" ємністю до 5.2 Гб та 3.5" ємністю 640 Мб.

Накопичувачі на компакт-дисках (CD-ROM) дозволяють тільки зчитувати дані, записані на диск. Маючи велику ємність (до 700 Мб) та високу швидкість зчитування, вони ефективні при зберіганні і використанні значних обсягів інформації.

Накопичувачі CD-RW (CD-Rewritable) сумісні із звичайними компакт-дисками, але дають змогу формувати, записувати і читати RW-диски.

Накопичувач DVD – пристрій для читання і перезапису цифрової інформації. При використанні двошарової схеми запису на односторонньому диску можна розмістити 8,5 Гб інформації, а на двосторонніх – до 17 Гб.

Пристрої обміну даними

Засоби передачі інформації на відстань називають **телекомунікаціями** (грецьк. tele – вдалину, лат. communicatio - спілкування). Найперспективнішим видом передачі даних є комп'ютерні телекомунікації. Для обміну інформацією між віддаленими комп'ютерами каналами зв'язку використовують **модеми**. Найбільш поширені модеми, які підключаються до комутованих телефонних каналів зв'язку. Цифрові дані, які надходять в модем з комп'ютера, модулюються в ньому за амплітудою, частотою і фазою відповідно стандарту (протоколу) і передаються в телефонну



лінію. Модем-приймач за цим же протоколом здійснює обернене перетворення (демодуляцію) і пересилає відповідні цифрові дані у свій комп'ютер. Так здійснюється віддалений зв'язок і обмін даними між комп'ютерами. Основні параметри модемів: продуктивність (біт за секунду), підтримувані протоколи зв'язку та корекції помилок, шинний інтерфейс (для внутрішніх модемів).

1.3.3. Пристрої системного блоку

Системний блок (рис. 1.6) - головна частина комп'ютера. В ньому розміщені основні елементи, що визначають тип і характеристики комп'ютера (рис.1.7):



Рис.1.6

- **системна (материнська) плата** (рис 1.8) - основна плата комп'ютера, на якій розміщені процесор, пам'ять, шини, контролери пристроїв;

- ✓ **процесор** - основна мікросхема, що виконує математичні і логічні операції;

- ✓ **оперативна пам'ять** (RAM – Random Access Memory) - електронна плата, в якій виконуються всі обчислювальні процеси. Дані в ній зберігаються лише при



Рис.1.7. Блок-схема пристроїв персонального комп'ютера



ввімкненому комп'ютері;

✓ **кеш-пам'ять** – ділянка оперативної пам'яті, де розміщені дані, які використовуються часто;

✓ **шина** – канал зв'язку (група провідників) для обміну даними між окремими платами комп'ютера і центральним процесором;

✓ **контролери пристроїв** - електронні плати, мікросхеми-посередники між системною шиною і підключеними пристроями, які забезпечують управління цими пристроями і їх роботу;

✓ **BIOS** (Basic Input/Output System) – базова система введення-виведення. Ця частина програмного забезпечення зберігається в постійній (енергонезалежній) пам'яті і здійснює тестування і запуск комп'ютера при включенні, завантаження операційної системи та її зв'язок з апаратними засобами;

✓ **CMOS** – мікросхема, яка містить відомості про конфігурацію і склад обладнання комп'ютера, а також про режими його роботи;

• **вінчестер** (накопичувач на жорсткому диску, рис. 1.9) - пристрій для запису і зчитування програм і даних;

• **порт** – засіб для підключення до комп'ютера певних пристроїв або канал доступу до комп'ютера ззовні;

• **дисководи для накопичувачів на гнучких дисках** - пристрій для запису і зчитування інформації з дискет;

• **дисководи для компакт-дисків** (CD-ROM, CD-RW);

• **блок живлення** перетворює змінний струм стандартної електромережі (220 В, 50 Гц) на постійний струм низької напруги 12 і 5 В.

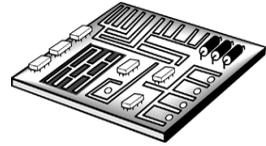


Рис. 1.8



Рис. 1.9



1.4. Програмне забезпечення ПК

ПК можна розглядати як апаратно–програмний комплекс, в якому набір взаємозв'язаних пристроїв (Hardware) тісно взаємодіє з програмним забезпеченням (Software). Керування роботою окремих пристроїв відбувається не лише апаратними, а й програмними засобами.

***Комп'ютерна програма** (англ. *computer program*) – запис алгоритму розв'язування задачі у вигляді послідовності команд або операторів мовою, зрозумілою комп'ютерові.*

***Програмне забезпечення** – сукупність програм, процедур і правил та відповідної документації, що використовуються для автоматизації опрацювання інформації комп'ютером.*

Склад програмного забезпечення обчислювальної системи називають **програмною конфігурацією**. За своїм функціональним призначенням все програмне забезпечення можна поділити на дві групи: системне та прикладне.

Системне програмне забезпечення - це сукупність програм, що використовуються для організації роботи комп'ютера як системи (операційна система), та програм, що є інструментальними засобами створення нових програм (інструментальне програмне забезпечення).

Зрозуміло, що операційна система (ОС) є найважливішою компонентою системного програмного забезпечення, адже від її ефективності залежить продуктивність роботи комп'ютера в цілому. ОС, як правило, постачається виробником разом з апаратною частиною. Разом з тим, операційну систему можна вважати гнучкою складовою частиною комп'ютера, оскільки її заміна чи модифікація дозволяє нарощувати потужність обчислювального комплексу.

До **інструментального програмного забезпечення** належать інструментальні мови і системи програмування. Ці засоби призначені для створення програм. Інструментальне забезпечення поділяють на мови низького рівня та мови високого рівня. До програмних засобів низького рівня належать мови **асемблера**, які близькі до машинної мови. Для написання програм у більшості випадків використовуються інструментальні мови високого рівня, які виконують роль посередників між людиною і комп'ютером. Текст складеної програми перекладається на мову машинних кодів спеціальною програмою, яку



називають **транслятор**. Транслятори бувають двох типів: інтерпретатори і компілятори. **Інтерпретатор** послідовно читає, аналізує і, якщо відсутні помилки, виконує оператори один за одним. Якщо в операторі виявлена помилка, на екран виводиться відповідне повідомлення. **Компілятор** читає, аналізує та перекладає на мову машинних кодів усю програму. Після завершення трансляції програма може бути виконана.

Перед написанням програми мовою високого рівня необхідно скласти **алгоритм** розв'язування задачі, тобто поетапний план дій, які треба виконати, щоб одержати розв'язок задачі. Тому такі мови часто називають **алгоритмічними мовами**. Приклади алгоритмічних мов високого рівня: Visual BASIC, C, Delphi, Pascal та ін. Детальніше основи алгоритмізації будуть розглянуті в розділі 5.

До складу **системи програмування** крім транслятора входять текстовий редактор, бібліотека стандартних програм, редактор зв'язку (компонувальник), система виконання програми, засоби автоматизації програмування тощо. Системи програмування використовуються для створення нових прикладних чи системних програм, робота яких суттєво пов'язана із специфікою операційної системи, під керівництвом якої вони виконуються. У зв'язку з цим інструментальне програмне забезпечення часто називають **розширенням ОС**. Іноді воно частково чи повністю входить до складу операційної системи (наприклад, до ОС UNIX).

При підключенні до обчислювальної системи нових пристроїв на системному рівні повинна бути встановлена програма, яка забезпечує для інших програм взаємозв'язок з цим пристроєм.

*Програми, які здійснюють взаємодію з певними пристроями, називаються **драйверами пристроїв**.*

Ефективність роботи з комп'ютером залежить від групи програм системного рівня, які відповідають за взаємодію з користувачем. Ці програмні засоби називають **засобами забезпечення користувацького інтерфейсу**. Вони дають можливість вводити дані в обчислювальну систему, керувати її роботою і отримувати результат у зручній формі.

*Службові програми, які призначені для організації безпечного використання даних, їх відновлення та інших сервісних функцій, називають **утилітами**.*



Серед відомих утиліт можна відзначити Norton утиліти, архіватори, антивіруси. Більш детальна інформація про склад і можливості ОС розміщена в розділі 2.

Прикладне програмне забезпечення складається із спеціалізованих пакетів програм, які використовуються для розв'язування тих чи інших конкретних задач, що не пов'язані з обслуговуванням та забезпеченням працездатності ПК. До таких пакетів належать:

- редактори документів (наприклад, MS Word);
- табличні процесори (MS Excel, Lotus 1-2-3, Quattro Pro);
- програми підготовки презентацій (MS PowerPoint);
- графічні редактори (Paintbrush, Adobe Photoshop, CorelDraw);
- видавничі системи (PageMaker, QuarkXpress);
- системи керування базами даних (MS Access, FoxPro, Oracle, MS SQL Server, Informix);
- системи — автоматизованого проектування (CAD-системи, зокрема, AutoCad);
- браузер (internet Explorer);
- редактори HTML (WEB-редактори);
- системи штучного інтелекту і експертні системи;
- бухгалтерські системи (1С:Бухгалтерія);
- інтегровані системи діловиробництва (АРМ керівника);
- програми для анімації і створення комп'ютерного відео (3DStudio);
- правові бази даних (Гарант, Кодекс);
- персональні інформаційні менеджери (Lotus Organizer);
- програми планування (MS Project);
- програми розпізнавання графічної інформації (FineReader);
- перекладачі (Stylus);
- словники (Мультилекс);
- фінансові аналітичні системи;
- системи відеомонтажу;
- геоінформаційні системи та ін.

Окремо слід назвати групу програм, які перешкоджають нормальній роботі комп'ютера як цілісної системи: віруси, програми зламування систем захисту інформації і ПК, інші хакерські програми.



Контрольні запитання і завдання

1. Дайте визначення інформації. В яких одиницях вимірюється інформація?
2. Що вивчає інформатика? Сформулюйте її основні завдання.
3. Сформулюйте принципи кодування інформації.
4. Як перевести десяткове число в двійкову систему числення?
5. Опишіть історію розвитку ЕОМ. Чим визначаються покоління ЕОМ?
6. За якими ознаками здійснюють класифікацію ЕОМ?
7. Для чого призначений процесор?
8. Охарактеризуйте структурну схему ПК.
9. Що складає базову конфігурацію ПК?
10. Який склад і призначення периферійних пристроїв?
11. Перелічіть основні характеристики монітора.
12. Охарактеризуйте призначення клавіатури ПК та основних груп клавіш.
13. Які існують пристрої введення-виведення інформації?
14. Які пристрої використовуються для збереження інформації?
15. Опишіть засоби телекомунікації, їх призначення та характеристики.
16. Які пристрої розміщені:
 - а) в системному блоці;
 - б) на системній (материнській) платі?
17. Дайте визначення програмного забезпечення комп'ютерів. Які засоби входять до прикладного програмного забезпечення ПК?



2.1. Основи операційних систем

Операційна система – сукупність програмних засобів, що керують всіма процесами та ресурсами комп'ютера, організовують взаємодію з користувачем та автоматично завантажуються в оперативну пам'ять після ввімкнення комп'ютера.

Основні завдання операційної системи:

- керування роботою комп'ютера;
- запуск програм на виконання;
- забезпечення роботи файлових систем та захист даних;
- забезпечення функціонування мультизадачності;
- забезпечення міжкомп'ютерного зв'язку;
- забезпечення відновлення робоздатності комп'ютера після збоїв та виконання сервісних функцій;
- виконання команд.

Ресурсами комп'ютера, що підлягають керуванню операційною системою, є процесор (а точніше задачі, що виконуються на ньому), пам'ять та пристрої введення і виведення інформації.

Поняття процесу є головним при розгляді операційних систем. В залежності від предмету розгляду, під терміном **процес** розуміють: програму на стадії виконання, асинхронну роботу виконуючого модуля, засоби операційної системи для процедури, що виконується та інше. Іноді термін процес ототожнюють з поняттям задача (програма) на стадії виконання. Надалі потрібно враховувати, що програми, які завантажуються для виконання, можуть складатися з декількох процесів та створювати нові процеси. Ми будемо використовувати наступне визначення процесу.

Процес – програмний модуль на етапі виконання разом із своїми даними та виділеними операційною системою структурами даних.

Користувач є специфічним ресурсом операційної системи. З точки зору операційної системи користувач асоціюється з певним набором апаратних засобів, які він використовує, та певним переліком програм з наданими правами доступу.



Засоби взаємодії користувача з операційною системою називають інтерфейсом.

Основні функції операційної системи:

- 1) керування процесами (їх запуск та зняття з виконання, постановка в чергу, облік, надання пріоритетів тощо),
- 2) керування пам'яттю (виділення та вилучення ділянок пам'яті, перерозподіл та уникнення фрагментації, резервування областей для системних потреб тощо);
- 3) керування даними (логічна організація даних на зовнішніх запам'ятовувальних пристроях);
- 4) керування периферійними пристроями введення, виведення та зберігання інформації (враховуючи способи фізичної організації даних, використання переривань, драйверів пристроїв та ін.);
- 5) організація інтерфейсу з користувачем (різноманітні командні мови, командні оболонки, графічний віконний інтерфейс);
- 6) організація міжкомп'ютерного зв'язку (локальних та глобальних мереж, доступу до ресурсів Internet);
- 7) налаштування на конкретну конфігурацію комп'ютера.

Функції ОС, які найчастіше використовуються, реалізовані в ядрі, інші – в окремих файлах у вигляді програм, драйверів тощо.

Кожна програма та користувач використовує операційну систему і працює під її керівництвом. Від виду ОС залежить продуктивність роботи користувача, ступінь захисту даних, необхідні апаратні засоби для надійної роботи комп'ютерної системи.

Наведемо перелік операційних систем за хронологією їх розробки: Multics, OS/360, Unix, CP/M, MS DOS, MacOS, Net Ware, OS/2, MS Windows, Linux (1991) та інші. Кожній операційній системі в процесі її вдосконалення та розвитку надаються різні номери або імена (версії), які можуть відрізнитися для робочих станцій і серверів.

Сучасні ОС є багатокористувацькими та багатозадачними, що дозволяє використовувати один комп'ютер одночасно декільком користувачам. Як правило, за персональним комп'ютером працює один користувач, а багатозадачність у ньому реалізована в режимі розподілу часу. В цьому випадку кожен процес виконується певний короткий проміжок часу роботи процесора, протягом якого користувач може взаємодіяти із процесом у режимі діалогу. Окрім того, багатозадачність передбачає фоновий друк. Якщо комп'ютер



містить декілька процесорів, то можливе одночасне виконання декількох процесів (по одному на кожний процесор).

В операційних системах з розподілом часу не гарантується надійне виконання команди протягом певного проміжку часу. Тому їх не можна використовувати для керування складними та небезпечними виробництвами (атомні станції, хімічні заводи тощо). В таких випадках застосовують операційні системи реального часу, що гарантують виконання запитів (команд) за визначений час.

Кожна ОС містить наступні основні складові:

- ядро – невелика частина ОС, що резидентно (постійно) перебуває в оперативній пам'яті та виконує основні функції;
- інтерпретатор команд;
- командна оболонка;
- графічна оболонка;
- інструментальні засоби;
- драйвери пристроїв;
- прикладні стандартні програми;
- утиліти – програми обслуговування роботи пристроїв комп'ютера та ін.

Після ввімкнення або перезавантаження комп'ютера починає працювати спеціальна програма **BIOS** (базова система введення-виведення). Досить часто BIOS називають складовою частиною ОС. Ця програма записана в постійну пам'ять комп'ютера і виконує наступні дії:

- забезпечує перевірку апаратної частини комп'ютера;
- копіює завантажувальну частину ОС з носіїв інформації в оперативну пам'ять;
- передає управління роботою комп'ютера операційній системі;
- здійснює всі стандартні операції низькорівневого введення-виведення.

Операційні системи розміщують спеціальним чином (встановлюють) на стандартних носіях інформації. Як правило, це вінчестер, але можуть бути дискети, лазерні диски та інше.

|| *Системним диском називається диск, на якому встановлено операційну систему.*



2.2. Файлова система операційних систем

Одне із основних призначень операційної системи – надійне зберігання даних на носіях інформації. Вся доступна користувачеві інформація зберігається у файлах.

|| **Файл** – іменована область даних на носіях інформації.

Першими носіями інформації були дискети та вінчестери. Кожен з цих носіїв перед його використанням повинен бути певним чином підготовлений до роботи. Для вінчестерів спочатку виконується низькорівневе форматування фірмою-виробником. Після купівлі вінчестера та встановлення його в комп'ютер, виконують програму Fdisk, яка створює системну область вінчестера, розбиває його на розділи та логічні диски. Кількість розділів не може перевищувати 5 (перший розділ є обов'язковим). У першому розділі розміщують тільки один логічний диск, в інших – можна створити декілька.

Дискети та логічні диски перед використанням повинні бути відформатовані командою **Format**. При стандартному форматуванні дискети виконуються наступні операції: 1) розбивка поверхні на доріжки та сектори (відбувається форматування та запис секторних міток); 2) тестування на предмет наявності фізичних дефектів поверхні створених секторів; 3) заповнення всієї поверхні спеціальним кодом; 4) форматування на дискеті системної області та занесення в стартовий сектор завантажувального запису, 5) створення двох копій таблиці розміщення файлів (FAT), ініціалізація її нулями та реєстрація в них кластерів, які містять дефектні сектори. При форматуванні логічного диска вінчестера виконується сканування наявних на диску доріжок для пошуку тих, які містять дефектні сектори, та форматування на диску системної області.

Форматування диска приводить до стирання всієї інформації. Тому слід пам'ятати, що при випадковому форматуванні відбувається втрата даних. В нових операційних системах введено так зване безпечне (швидке) форматування, яке полягає не у перезаписі секторних міток, а тільки в тестуванні їх на цілісність. При безпечному форматуванні програма намагається створити на диску копію системної області. Якщо це виконано, то дані можна відновити повністю і після випадкового форматування. При безпечному форматуванні вінчестера поверхня дисків не тестується.



Вся відформатована поверхня диска логічно поділена на дві частини: **системну ділянку** та **ділянку даних**.

В ділянці даних автоматично створюється **кореневий каталог диска**.

Каталог – іменована ділянка диска, де зареєстровані файли та інші каталоги (кажуть - підкаталоги). Каталог є файлом спеціального виду. В ОС Windows каталоги називають **папками**.

З точки зору операційної системи дискети і вінчестери мають аналогічну структуру та способи доступу для зчитування або запису інформації (рис. 2.1). Вся поверхня диска після форматування розділена на **доріжки** (англ. tracks) – концентричні кола різних радіусів. Кожна доріжка, в свою чергу, розбивається на **сектори** (англ. sectors). Кількість доріжок та секторів визначається типом диска та його форматом. На дискеті 3,5” обсягом 1,44 Мб кожна з 80 доріжок містить 18 секторів. Всі сектори на одному диску мають однаковий розмір, стандартний обсяг пам’яті одного сектора становить 512 байт. Дані, які записуються на диск, можуть розміщуватися тільки повними секторами.

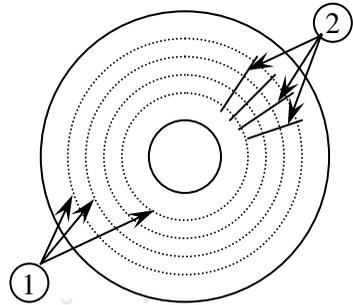


Рис. 2.1. Структура диска:
1 – треки (доріжки);
2 – сектори

Запис та зчитування інформації проводиться кластерами.

Кластер – один або декілька послідовних секторів в області даних.

Кластер є мінімальною адресованою одиницею дискової пам’яті. На дискеті він займає 1 сектор. Для дисків вінчестера розмір кластера залежить від ємності цього диска (як правило, не менше 2048 байт (4 сектори), але кратний степеням двійки). Чим більший диск, тим більший розмір кластера. Файл або каталог займає ціле число кластерів, які зв’язані в ланцюжок. Останній кластер може заповнюватися не повністю. Тому, чим більший розмір кластера та більше малих файлів на диску вінчестера, тим менш ефективно використовується ділянка даних.

Після того, як диски готові до роботи, на них розміщується файлова система.

Файлова система – набір специфікацій та відповідне програмне забезпечення, яке відповідає за створення, вилучення, організацію, читання, запис, модифікацію, переміщення файлової інформації, а також за керування доступом до файлів та керування ресурсами, які використовуються файлами.

До функцій обслуговування файлової системи належать: створення файлів і присвоєння їм імен; створення каталогів (папок) і присвоєння їм імен; перейменування файлів і каталогів; копіювання, переміщення та вилучення файлів і каталогів; навігація у файловій структурі з метою доступу до заданого файлу; керування атрибутами файлу.

Кожен диск у файловій системі має ім'я, що складається з латинської літери та двокрапки. Наприклад, A: і B: - дисководи для роботи з дискетами; C:, D:, E:, ... - назви логічних дисків на вінчестері та інших пристроях.

Файлову систему можна зобразити у вигляді ієрархічної структури – дерева каталогів (рис 2.2.). Кореневий каталог, наприклад, диска C: позначається C:\. Каталог першого рівня реєструється в кореневому каталозі. Всі решта каталогів є підкаталогами. Наприклад, каталог MAIL є підкаталогом каталога HELP, який, в свою чергу, є підкаталогом каталога WINDOWS, що розміщений у кореневому каталозі диска C: (записується c:\windows\help\mail).

При створенні (запису на диск) файлу чи каталога, його ім'я реєструється у вказаному каталозі, записується дата і час створення, номер початкового кластера у таблиці розміщення файлів, розмір файлу в байтах. Крім того, можна вказувати атрибути файлу: тільки для читання (r), архівний (a), прихований (h), системний (s). Відповідно до розміру файлу, на носіїві інформації виділяється певна кількість кластерів. Для каталога при його створенні відводиться стандартна кількість кластерів. Якщо каталог заповнений, але у ньому потрібно зареєструвати новий елемент, до каталога добавляється вільний кластер. Користувач здійснює доступ до файлів та каталогів за іменем, а операційна система – за номером початкового кластера.

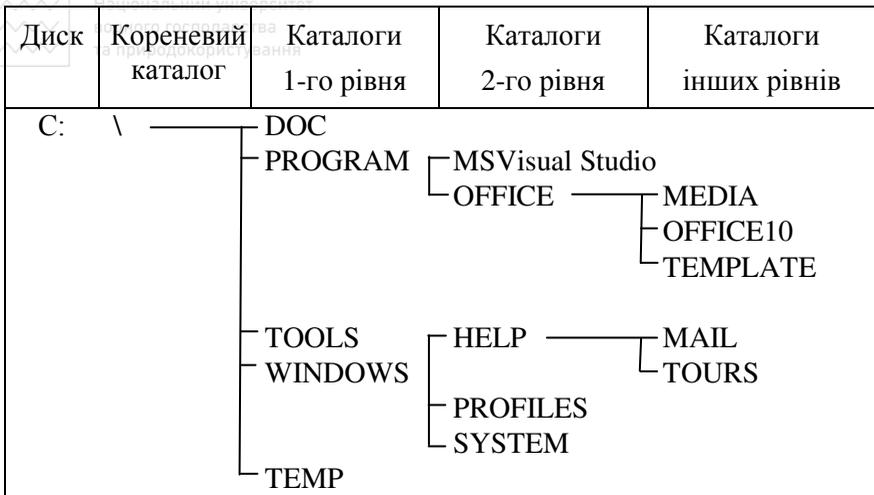


Рис 2.2. Приклад структури каталогів

У різних операційних системах вимоги до імен файлів та каталогів різні. Стандартом можна вважати наступні правила формування імені: повне ім'я файлу складається з назви файлу (від одного до восьми символів) та розширення (до 3-х символів). Назва файлу від розширення імені відокремлюється крапкою. При написанні імені файлу можна використовувати латинські літери, цифри та деякі спеціальні символи: - _ \$ # @ ! % . Наступні сполучення символів є зарезервованими і їх не можна використовувати в якості імен файлів: PRN (стандартний пристрій друку), LPT1-LPT4 (стандартні паралельні порти), COM1-COM4 (стандартні послідовні порти), AUX (комунікаційний порт), CON (стандартний пристрій введення – клавіатура або виведення – монітор), NUL (пристрій, в якому інформація при її виведенні пропадає, а при введенні виникає повідомлення кінця файлу введення). Найкраще надавати файлам оригінальні імена, враховуючи їх призначення. В одному каталозі не можна зареєструвати два різних файли з однаковими іменами.

Записані вище правила формують так звані короткі імена файлів. Основним недоліком коротких імен є їх низька змістовність. В ОС Windows допустима довжина повного імені файлу збільшена до 255 символів. Крім того, в іменах файлів можна використовувати символи національного алфавіту (кирилиці), пробіл, декілька крапок. Такі імена називають „довгими”. Але для кожного „довгого” імені ОС

Windows автоматично формує коротке ім'я, яке використовується для різноманітних цілей. Зокрема, при роботі в глобальній мережі та в деяких інших випадках краще використовувати короткі імена файлів.

Розширення імені файлу є необов'язковим і, як правило, вказує на тип інформації у файлі. В залежності від розширення, розрізняють наступні типи файлів: COM, EXE – файли з готовими до виконання програмами; BAT - командні; TXT – текстові файли; DOC – файли-документи MS Word; WQ1, XLS - файли електронних таблиць; PCX, BMP - графічні файли тощо. Сучасні програми за розширенням автоматично розпізнають файли, з якими вони можуть працювати, а ОС Windows додає до імені значок відповідної програми. При запису файлу на диск в цих програмах не обов'язково вказувати розширення, воно автоматично додається відповідно до типу інформації у файлі.

Часто для вибору групи файлів з одного каталога використовують **маску (шаблон)**, в якій застосовуються символи ? та *.

*Символ * замінює довільну кількість будь-яких символів у шаблоні групи файлів. Символ ? замінює тільки один будь-який символ.*

Наприклад: *.exe - всі файли з розширенням EXE (тільки ті файли, які мають це розширення); f?.* - ім'я файлу складається з двох символів, перший з яких f, другий – будь-який, розширення – будь-яке.

При роботі з файлами використовують поняття шляху.

***Шлях** вказує місце реєстрації файлів у дереві каталогів і складається з назви диску та послідовності імен каталогів, що розділені знаком \.*

Наприклад, C:\WINDOWS\HELP - диск C:, каталог WINDOWS, підкаталог HELP. Каталог (папка), розкритий користувачем чи програмою, називається **активним (поточним)**. При звертанні до файлів з інших каталогів перед іменем файлу вказується шлях. Наприклад, C:\WINDOWS\HELP\atm.chm.

Виконання команд операційної системи здійснюється різними способами. В інтерактивному режимі та в командних оболонках є спеціальне місце для введення команд з клавіатури. Після набору тексту команди потрібно натиснути клавішу **Enter**. Графічні оболонки дозволяють виконувати більшість команд шляхом вибору мишею потрібного пункту меню.



2.3. Windows XP

2.3.1. Загальні відомості

Починаючи з середини 90-х років XX ст. найпоширенішою операційною системою для персональних комп'ютерів стала система Windows фірми Microsoft. Найновіші її версії MS Windows XP та Vista. Для встановлення та нормальної роботи Windows XP і мінімального набору додатків рекомендовано мати в наявності комп'ютер з процесором не гіршим за Pentium 500 МГц, оперативною пам'яттю не менше 256 Мб, та вінчестер, на якому відведено логічний диск обсягом не менше 10 Гб.

Графічний інтерфейс Windows відповідає основним вимогам існуючих міжнародних стандартів і дає змогу користувачеві досить просто керувати роботою комп'ютера. В ньому більшість дій можна виконати з допомогою миші, використовуючи такі поняття, як **Мій комп'ютер**, **Мережеве оточення**, **кнопка Пуск**, **Панель задач**, **контекстне меню**, **вікно**, **ярлик**, **значок**, технології **“Вкажи і вибери” (Point an Click)**, **“Перенеси і відпусти” (Drag and Drop)** та застосовуючи такі офісні аналогії, як **робочий стіл**, **папка**, **документ**, **кошик**. Основний принцип роботи в цій операційній системі - дивись і вибирай з допомогою маніпулятора “миша”.

Операційна система Windows XP забезпечує: підтримку режиму відображення WYSIWYG (What You See Is What You Get = Те, що Ви бачите, отримуєте на папері при друкуванні); підтримку специфікації Plug&Play (поставив і працюй або грай) з використанням широкого набору драйверів для різноманітних пристроїв; підключення до глобальних та локальних комп'ютерних мереж (електронна пошта, факс, Інтернет); пряму адресацію оперативної пам'яті, завдяки чому у програмах-додатках можуть використовуватися до 4 Гб віртуальної пам'яті (оперативна пам'ять і пам'ять на жорсткому диску); підтримку обміну даними між додатками за допомогою OLE-технології (англ. Object Linking and Emdedding – зв'язування та вбудовування об'єктів). Наприклад, таблиці чи діаграми, побудовані в табличному процесорі Excel, можуть використовуватися в документі, створеному в текстовому редакторі Word. Додаткові зручності користувачеві створюють стандартні програми ОС Windows XP: Проводник, Блокнот, графічний редактор Paint, текстовий редактор WordPad, Калькулятор та ін.



Операційна система Windows XP оперує великою кількістю структур даних про апаратні засоби, програмне забезпечення, користувачів. Вся ця інформація зібрана в окрему базу даних – **реєстр**. В реєстрі міститься майже вся інформація, необхідна для завантаження та конфігурування системи.

Кожен користувач системи повинен бути зареєстрований з унікальним іменем (**login**) та мати свій пароль (**password**). Користувачеві виділяються ресурси і надаються права доступу до них, встановлюються правила роботи та обмеження щодо використання можливостей ОС. Зокрема, папки одних користувачів можуть бути недоступні іншим користувачам та ін.

Програми, призначені для роботи в цій операційній системі, повинні бути встановлені. Це означає, що їх необхідно записати спеціальним чином у певне місце диска (за замовчуванням в папку **Program Files** системного диска), зареєструвати в системному реєстрі, налаштувати на конкретну конфігурацію комп'ютерної системи. Більшість програм не буде виконуватися, якщо їх просто скопіювати на диск. Як правило, для встановлених програм створюється відповідний підпункт меню у пункті головного меню **Все програми**.

|| *Програми, призначені для роботи в операційній системі Windows, називаються **додатками**.*

Кожен додаток має свій ярлик та працює з тими чи іншими типами файлів даних. В свою чергу, кожен тип файлів може мати своє розширення. Розширення імені файлу ставиться у відповідність додатку (зв'язується з додатком), який автоматично повинен опрацювати файли даного типу.



2.3.2. Робота з об'єктами

Інтерфейс користувача операційної системи Windows зображений робочим столом за аналогією робочого столу в офісі (рис. 2.3).

Робочий стіл – графічний інтерфейс операційної системи

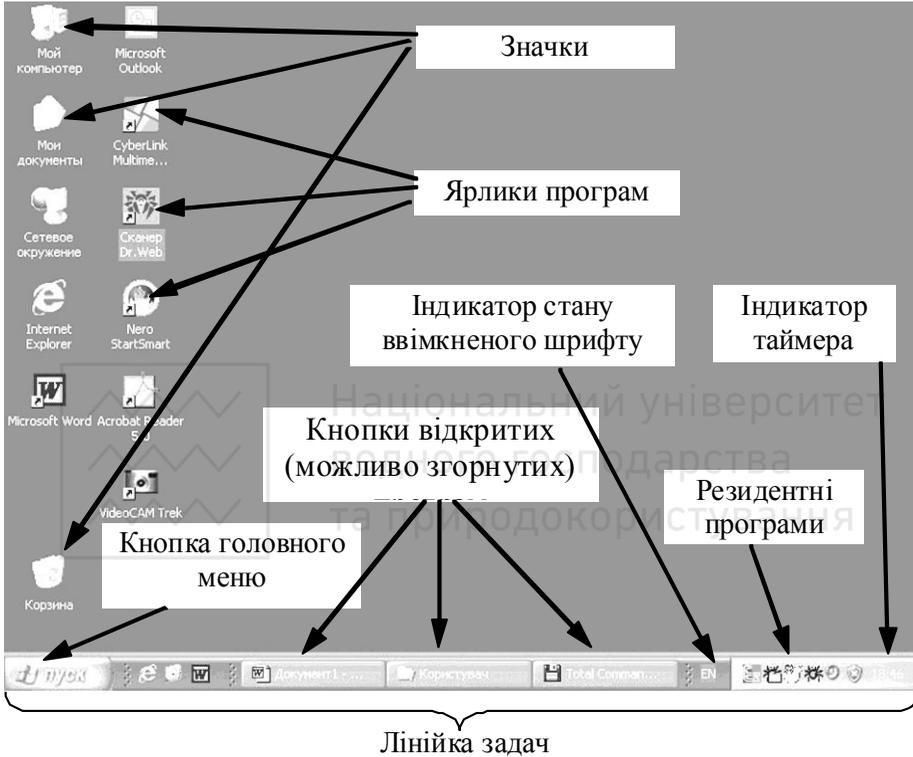


Рис.2.3. Робочий стіл MS Windows

Windows, на якому розташовані об'єкти. Всі дії користувача при роботі в системі можна звести до роботи з об'єктами або елементами керування

Об'єктами робочого столу є значки, ярлики, папки, документи, елементи управління. Кількість відображених елементів залежить від конфігурації операційної системи та встановленого програмного забезпечення.

Національний університет
та прикладного використання

Об'єкт - документ, файл, папка, програма, група програм, диск, значок, ярлик.

Документ - файл, що містить деякі дані (текст, таблицю, малюнок, програму тощо), який призначений для обробки будь-яким додатком Windows.

Значок - графічне відображення об'єкта, який можна вибрати (диск, документ), відкрити (папка), виконати (додатки Windows, програми). При знищенні значка знищується і сам об'єкт.

Ярлик - посилання на деякий об'єкт, який розміщений у файлової системі. При знищенні ярлика об'єкт не знищується. Використовується, як правило, для швидкого запуску програм або вибору об'єктів (наприклад, диска чи документа). Графічне зображення містить стрілку.

У нижній частині робочого столу розміщена **Лінійка задач** (панель завдань) з кнопкою **Пуск** (головне меню операційної системи), індикаторами резидентних програм (праворуч) та кнопками відкритих задач (ліворуч). Вихід із системи або вимкнення комп'ютера відбувається з допомогою відповідної команди головного меню ОС, вимкнення іншими способами може привести до втрат даних або краху системи. Поряд із кнопкою **Пуск** може розміщуватися панель швидкого запуску програм.

Будь-яка програма, що працює у віконному режимі, відображена кнопкою на лінійці задач, незважаючи на те, згорнута вона чи ні. Її легко активізувати клацанням лівою кнопкою миші на відповідній кнопці.

Після вибору кнопки **Пуск** клацанням лівою кнопкою миші або натискання клавіші **Window** на клавіатурі, на екрані з'являється головне меню операційної системи. В заголовку меню міститься ім'я користувача, нижче – значки з назвами програм, основні системні папки, команди, пункт меню **Все програми**, кнопки **Виход из системи** та **Виход**. Зліва в головному меню міститься список програм, які найчастіше використовуються. Призначення основних елементів головного меню наведено в таблиці 2.1.

Кожен розроблений додаток Windows працює під управлінням певних засобів ОС, які називають **монітором вікна**. Монітори



повинні забезпечувати функціонування стандартних елементів віконного інтерфейсу, передачу управління операційній системі для виконання сервісних функцій. Тобто, будь-яка програма виконується тільки у вікні, яке може бути видимим або невидимим.

Таблиця 2.1.

Призначення основних елементів головного меню

Пункт меню	Призначення
Мои документы	Папка Мои документы активного користувача (розміщена у файловій системі диска та на робочому столі)
Мой компьютер	Провідник у системі дисків та пристроїв, які під'єднані до комп'ютера
Панель управления	Список утиліт, які дозволяють змінювати системні налаштування
Выбор программ по умолчанию	Вибір програм, які використовуються за замовчуванням для різних операцій
Принтеры и факсы	Папка із встановленими принтерами та факсами з можливістю їх додаткових налаштувань
Поиск	Команда для запуску утиліти пошуку на диску папки, файлів тощо
Выполнить	Виклик вікна для введення та виконання команди (програми)
Выход из системы	Команда для зміни користувача або завершення роботи користувача в системі
Завершение работы	Команда коректного завершення роботи у Windows XP або перезавантаження комп'ютера

***Вікно** - обмежена рамкою (границею) прямокутна область екрана, всередині якої можуть розташовуватися об'єкти або виконуватися певні дії.*



На робочому столі може бути відкрито декілька вікон, що відповідають додаткам, які виконуються. Вікна можуть перекриватися одне одним чи розміщуватися поряд. У будь-який момент часу тільки одне з вікон є активним.

Способи подання вікон: повноекранне (вікно займає робочий стіл, границь немає), звичайне (вікно займає частину робочого столу, є границі), згорнуте (вікно відображене кнопкою на лінійці задач або у вікні додатку, а самого вікна на робочому столі немає).

Типи вікон: вікна прикладних програм (додатків), підпорядковані вікна (документів та діалогові).

Елементи управління вікон діалогу (рис. 2.4):

- 1) кнопки;
- 2) поля введення;
- 3) списки;
- 4) лічильники
- 5) прапорці
- 6) перемикачі

Крім того, діалогові вікна можуть містити наступні елементи: селектори, індикатори стану, закладки.

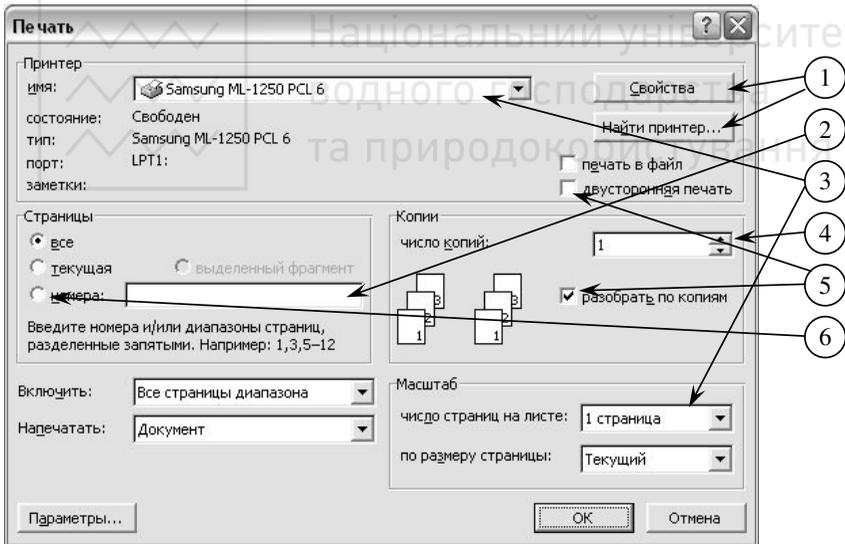


Рис. 2.4. Пример диалогового окна

У роботі з діалоговими вікнами використовують такі клавіші чи їх комбінації:

Tab - перехід від однієї опції до іншої вперед;

Shift+Tab - перехід від однієї опції до іншої у зворотному

Пробіл - вибір із списку або відмова від вибору.

Структура та основні елементи вікна подані на рис. 2.5.

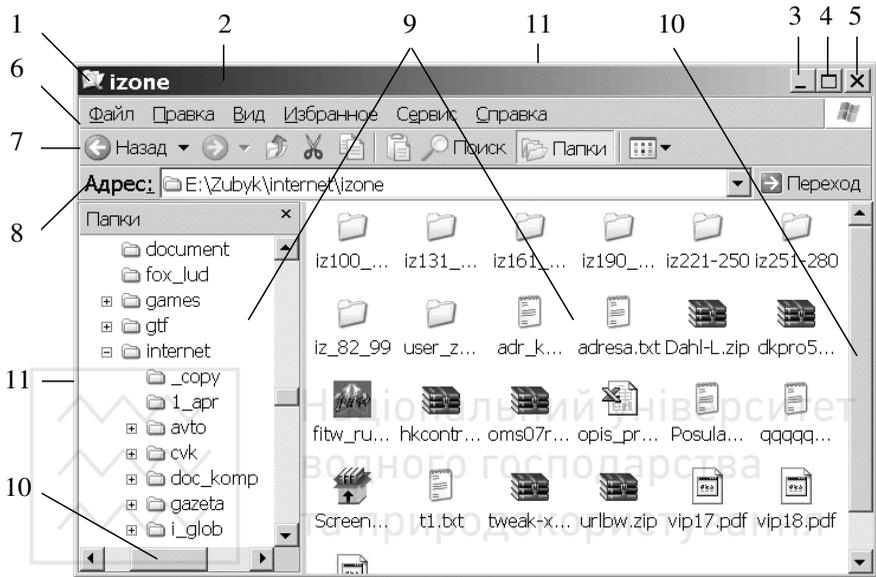


Рис. 2.5. Структура вікна на прикладі програми Провідник

- 1 – системна кнопка вікна,
- 2 – рядок заголовка,
- 3 – кнопка згортання вікна,
- 4 – кнопка розгортання вікна на весь екран або відновлення його попередніх розмірів,
- 5 – кнопка закриття вікна,
- 6 – рядок меню,
- 7 – панель інструментів,
- 8 – адресний рядок,
- 9 – робоча зона вікна (зліва – панель папок, а справа – вміст вибраної папки),
- 10 – смуги скролінгу,
- 11 – границя вікна.



2.3.3. Основні прийоми роботи з мишею

Зручна робота у Windows XP забезпечується графічним середовищем та активним використанням маніпулятора „миша”. Надалі будемо вважати, що у властивостях папки (Панель управління, Свойства папки) вказаний перемикач **Открывать двойным, а выделять одним щелчком**. Вказівник миші може мати різне графічне зображення на робочому столі та автоматично змінювати вигляд в залежності від свого місцезнаходження та режиму роботи (рис.2.6).

Основні прийоми роботи з мишею:

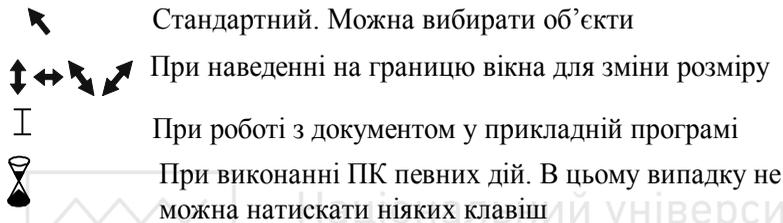


Рис. 2.6. Види вказівника миші

- **навести вказівник миші на об'єкт** – перемістити мишу на столі (килимку) так, щоб кінець стрілки її вказівника був розміщений приблизно на середині об'єкта. Як правило, в цьому випадку біля вказівника миші з'являється підказка, яка пояснює призначення об'єкта;
 - **клацнути** - короткочасно натиснути і відпустити кнопку миші. Зазвичай клацають лівою кнопкою миші, якщо не вказано інше;
 - **подвійно клацнути** – швидко двічі клацнути лівою кнопкою миші практично без затримки між клацаннями;
 - **відмітити об'єкт** – навести вказівник на об'єкт і клацнути лівою кнопкою миші;
 - **транспортувати** – переміщувати вказівник миші при натиснутій лівій або правій кнопці миші;
 - **перетягнути об'єкт** – навести вказівник миші на об'єкт та транспортувати (в деяких випадках об'єкт можна скопіювати, якщо транспортувати його при натиснутій клавіші **Ctrl**);
 - **виконати пункт меню чи дію, що позначена кнопкою**, – навести вказівник миші на відповідний елемент та клацнути лівою кнопкою миші;



• **виконати програму, відкрити вікно** – подвійно клацнути на відповідному значку (ярлику);

• **викликати контекстне меню об'єкта** – навести вказівник миші на об'єкт та клацнути правою кнопкою миші. Контекстне меню містить основні команди для роботи з вибраним об'єктом, які найчастіше використовуються;

• **створити папку на робочому столі** – клацнути правою кнопкою миші на робочому столі, в контекстному меню вибрати спочатку пункт **Создать**, а потім **Папку**;

• **створити ярлик на робочому столі** – клацнути правою кнопкою миші на робочому столі, в контекстному меню вибрати спочатку пункт **Создать**, а потім **Ярлык**. Після натискання кнопки **Обзор...** знаходимо відповідну програму на диску, після чого вказуємо назву програми;

• **сортувати значки на робочому столі** – клацнути правою кнопкою миші на робочому столі і в контекстному меню вибрати пункт **Упорядочить значки**;

• **вилучити активний (відмічений) об'єкт** – натиснути клавішу **Delete** або клацнути правою кнопкою миші на робочому столі і в контекстному меню вибрати пункт **Удалить**;

• **відмінити меню** – клацнути мишею в робочій зоні вікна (але не на об'єктах, що викликають дії) або натиснути клавішу **Esc**;

• **відміна деяких дій** – пункт контекстного меню **Отмена** або **Правка / Отменить**;

• **викликати контекстне меню панелі задач** – клацнути правою кнопкою миші на порожньому місці лінійки задач (комбінація клавіш **Shift+F10**). Це меню містить наступні пункти: **Панели инструментов, Настройка даты/времени, Окна каскадом, Окна сверху вниз, Окна слева направо, Свернуть все окна, Свойства**;

• **перемістити вікно** – навести вказівник миші на заголовок і транспортувати його на нове місце;

• **змінити розмір вікна** – навести вказівник на границю вікна і транспортувати її.

Виконання команд, активних пунктів меню, натискання активних кнопок може здійснюватися не тільки клацанням лівою кнопкою миші, а й натисканням клавіші **Enter**. Якщо клацнути мишею на будь-якому об'єкті, – об'єкт стає активним.



2.3.4. Деякі прийоми роботи

Для копіювання інформації між різними додатками чи в різні частини одного додатку в ОС Windows створений спеціальний засіб, який називається **Буфером обміну** (англ. Clipboard). Це невидима користувачу ділянка пам'яті, у якій дозволяється зберігати тільки один об'єкт. Скопіювати або вирізати активний об'єкт у буфер можна за допомогою команд **Копировать** чи **Вырезать**. Ці команди містяться у меню **Правка**, а також дублюються відповідними піктограмами панелі інструментів **Стандартная**. Вміст **Буфера** можна вставити у будь-яке місце об'єкта, використовуючи інструмент **Вставить** (рис. 2.7) або команду меню **Правка / Вставить**. Вказані команди меню є у всіх додатках, що працюють з виділеними об'єктами. Окрім того, вони містяться і в контекстному меню об'єктів. Якщо немає активного або відміченого об'єкта, команди копіювання та вирізання не активні. Якщо буфер обміну порожній, то команда вставки недоступна.

В деяких програмах є свої засоби для роботи з буфером обміну, які дозволяють працювати одночасно з декількома об'єктами.



Рис. 2.7. Інструменти

Операції копіювання, вирізання та вставки можна виконувати і з

допомогою клавіатури. В таблиці 2.2 наведені основні клавішні комбінації, які полегшують роботу користувача. В таблиці 2.3 вміщено короткі інструкції для виконання основних дій.

Вырезать, Копировать, Вставить

Таблиця 2.2.

Призначення клавіш та їх комбінацій

Комбінація клавіш	Призначення
Ctrl+X	Перенесення виділеного фрагмента у буфер



Ctrl+C або Ctrl+Ins	Копіювання виділеного фрагмента у буфер
Ctrl+V або Shift+Ins	Вставка вмісту буфера в поточну позицію
Ctrl+A	Об'єднання всіх об'єктів у групу
Alt+F4	Закривання прикладного вікна або завершення додатка
Alt або F10	Вибір першого елемента меню або вихід з меню
Alt+буква	Вибір елемента меню, який містить підкреслену <i>букву</i>
PrtScr	Копіювання вмісту екрана в буфер
Alt+PrtScr	Копіювання вмісту активного вікна в буфер
Alt+пробіл	Відкривання системного меню (для прикладного вікна)
Alt+Esc або Alt+Tab	Перехід до наступного об'єкта (програми, вікна)
Ctrl+F4	Закривання вікна документа
Ctrl+F6	Перехід до наступного документа
Ctrl+Alt+Del	Виклик вікна Диспетчер задач Windows для виконання сервісних функцій з процесами та перегляду інформації про процеси, мережу чи користувачів

Таблиця 2.3.

Виконання основних дій

Дія	Порядок виконання дії (комбінація клавіш)
Відмітка групи об'єктів	Відмітити перший, Ctrl + ЛКМ відмітити інші або відмітити перший, Shift + ЛКМ відмітити останній



Копіювання групи об'єктів	“Перетягування” ЛКМ відмічених об'єктів при натисненій клавіші Ctrl
Перенесення групи файлів	“Перетягування” ЛКМ відмічених файлів
Вилучення об'єктів	“Перетягування” ЛКМ відмічених об'єктів у Корзину або відмітити об'єкти і натиснути Delete
Настроювання властивостей об'єктів	Головне меню, Панель управління... або контекстне меню / Свойства
Створення папки / ярлика 	На робочому столі клацнути правою кнопкою миші, вибрати пункт меню Создать , в підменю вибрати папку або ярлик. У файлової системі на дисках: виконати програму Проводник ; вибрати місце розташування об'єкта (наприклад, диск A: ; зайти в пункт меню Файл , далі – Создать, Папку / Ярлык (або скористатися контекстним меню)
Зміна властивостей робочого столу та екрана	На робочому столі клацнути правою кнопкою миші, вибрати пункт меню Свойства . Для зміни фонового малюнка робочого столу вибрати вкладку Рабочий стол . Для зміни властивостей екрана (роздільна здатність, палітра, частота тощо) вибрати вкладку Параметры . Для зміни заставки та параметрів і часу її появи потрібно вибрати вкладку Заставка
Доповнення панелей інструментів на лінійку задач	На лінійці задач клацнути правою кнопкою миші, вибрати пункт меню Панели инструментов та потрібний підпункт меню
Відміна останньої дії	Ctrl+Z



2.4. Основні стандартні програми Windows

2.4.1. Програма Проводник

Програма **Проводник** (англ. Explorer) – основний стандартний засіб операційної системи Windows для роботи з файловою системою. Програми **Мой компьютер**, **Internet Explorer** та інші мають аналогічний інтерфейс, команди і правила роботи. За принципом роботи програми **Проводник** організована робота у вікнах команд **Сохранить**, **Открыть**, **Поиск** та інших прикладних програмах Windows.

Для запуску програми на виконання можна клацнути правою кнопкою миші на кнопці **Пуск** лінійки задач і вибрати пункт меню **Проводник**. Інша можливість – **Пуск / Все программы / Стандартные / Проводник**.

Вікно програми зображене на рис. 2.5. Воно містить стандартні елементи, які характерні для всіх програм Windows. Робоча зона вікна може бути поділена на дві частини з допомогою пункту меню **Вид** підпункту **Панели обозревателя**. Для роботи з папками у лівій частині вікна програми **Проводник** вибирають режим **Папки**. У цьому випадку зліва відображено дерево об'єктів робочого столу: **Мои документы**, **Мой компьютер**, **Мое сетевое окружение**, **Корзина** та ін. Об'єкт **Мой компьютер** містить інформацію про наявні диски та папки на ПК і програму **Панель управления**. Біля кожного об'єкта зліва може знаходитися символ "+". Це означає, що об'єкт містить інші об'єкти, але вони приховані. Якщо перед об'єктом стоїть знак "-", то об'єкт містить інші об'єкти, які показані на дереві нижче і правіше. Якщо біля об'єкта немає цих символів – він не містить інших об'єктів, окрім файлів. Для перегляду вмісту об'єкта необхідно клацнути на "+". Щоб приховати вміст, клацаємо на символі "-".

Програма **Проводник** дозволяє виконувати наступні **основні операції з файловою структурою**: здійснювати навігацію у файльовій системі; запускати програми і відкривати документи; створювати папки; копіювати файли і папки; переміщувати файли і папки; знищувати файли і папки; створювати ярлики, форматовувати диски та інше.

Щоб показати вміст папки (розгорнути папку), необхідно клацнути мишею на значку відповідної папки у лівій частині вікна

Проводник (розділ **Папки**) або двічі клацнути на папці у правій частині вікна. Назва активної папки чи диска записується в адресній смужі. **Перенесення об'єкта** з правої частини вікна у будь-яку папку лівої частини вікна здійснюється транспортуванням з допомогою ЛКМ.

Щоб вийти з активної папки, клацаємо на інструменті **Вверх** (рис.2.8 а)). Повернутися на крок назад можна за допомогою інструменту **Назад**. Зміна вигляду інформації про елементи файлової системи виконується наступними командами меню **Вид** (інструмент на рис. 2.8 б)): **Ескизы страниц, Папки, Значки, Список, Таблица**.



а) б)
Рис. 2.8

Сортування інформації про вміст папок у правій частині вікна програми виконується з допомогою меню **Вид, Упорядочить значки, по имени / по типу (розширенню) / по размеру / по дате / автоматически**. Зміна вигляду інформації відбувається з використанням одного із підпунктів пункту меню **Вид: Крупные значки / Мелкие значки / Список / Таблица**.

Для пошуку інформації потрібно виконати команду **Найти...** з пункту меню **Файл**. В результаті відкриється вікно програми пошуку інформації, зображене на рис. 2.9. Інша можливість – клацнути інструмент **Поиск**.

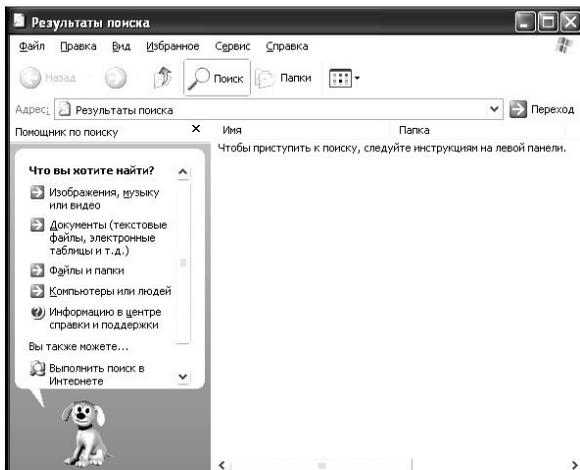


Рис. 2.9.

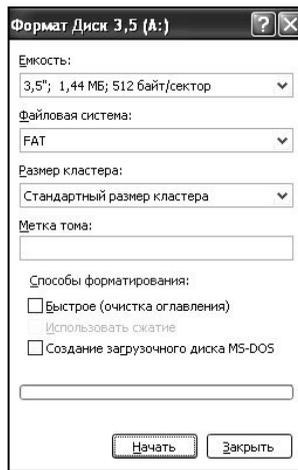


Рис. 2.10.



Форматування диска можна здійснити програмою, яка викликається з допомогою пункту **Форматировать...** контекстного меню диска (рис. 2.10).

Таблиця 2.4.

Робота з активними об'єктами

Дія	Порядок виконання
Відмінити останню команду роботи з об'єктом	Правка / Отменить
Перемістити об'єкт	Правка / Переместить в папку
Скопіювати об'єкт	Правка / Копировать в папку
Вилучити об'єкт	Файл / Удалить або Delete
Змінити назву об'єкта	Файл / Переименовать
Виконати файл, відкрити вікно папки, відкрити додаток, який зв'язаний з даним файлом, і завантажити в нього цей файл	Файл / Открыть або Enter
Відмітити (вибрати) всі об'єкти	Правка / Выделить все
Переглянути (змінити) властивості об'єкта	Файл / Свойства або Alt+Enter
Створити папку / ярлик або інший об'єкт	Файл / Создать / Папку, Ярлык або вибрати із списку інше
Показати всі підкаталоги активного каталога	* або Ctrl+*
Копіювати вміст однієї дискети на іншу	Клацнути правою кнопкою миші на значку диска і вибрати Копирование Диска



Налаштування параметрів роботи провідника та деяких властивостей ОС Windows можна здійснити командою **Свойства папки** у пункті меню **Сервис** програми **Проводник**. У відкритому вікні (рис. 2.11) існує 4 вкладки:

Общие, **Вид**, **Типы файлов**, **Автономные файлы**.

На першій вкладці можна змінити режим відображення файлів і папок у програмі **Проводник** та на робочому столі, а також спосіб відкривання чи виділення файлів.

Друга вкладка дозволяє вказати спеціальні способи відображення всіх елементів файлової системи, що дозволяє налаштовувати систему для потреб конкретного користувача.

На вкладці **Типы файлов** наведений список зареєстрованих у MS Windows типів файлів з інформацією про те, якою програмою за

замовчуванням опрацьовується той чи інший тип. Тут можна змінити програму, що відповідає за дане розширення імені файлу, або зареєструвати нове розширення. На четвертій вкладці можна змінити налаштування для роботи з мережевими файлами.

Після знищення файлів стандартним чином, з незмінних носіїв інформації вони переносяться у спеціальну папку **Корзина** (рис. 2.12). Це дає можливість відновити випадково знищені файли. Об'єкти, що перебувають у папці **Корзина**, займають місце на диску, тому її потрібно періодично очищати або це буде виконуватися операційною системою в автоматичному режимі. Якщо елемент **Корзина** був очищений, то відновити файли практично неможливо. Для того, щоб вилучити файли без занесення їх в **Корзину**, при виконанні операції знищення потрібно тримати натиснутою клавішу **Shift**.

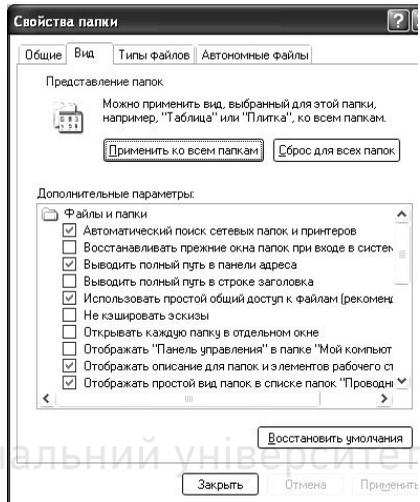


Рис.2.11. Вікно Свойства папки

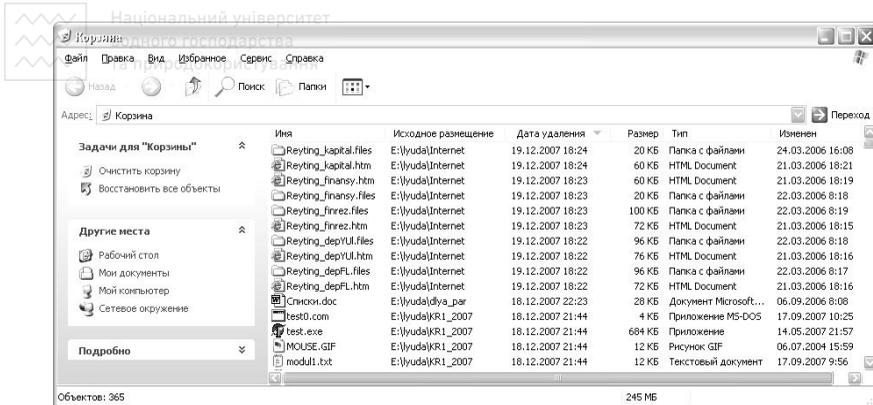


Рис. 2.12. Папка Корзина

2.4.2. Службові програми

Повний перелік встановлених службових програм вміщений у пункті підменю **Службные** пункту меню **Стандартные**. До основних службових програм належать: **Архивация данных, Восстановление системы, Дефрагментация диска, Мастер переноса файлов и параметров, Назначенные задания, Очистка диска, Сведения о системе, Таблица символов, Центр обеспечения безопасности.**

Програма архівації даних призначена для резервного копіювання важливої інформації з дисків комп'ютера на змінні носії інформації з використанням механізмів компресії. Зарезервована таким чином інформація може бути при потребі відновлена (наприклад, при виході з ладу вінчестера). Після виконання команди архівації в окремому вікні починає працювати **Мастер архивации или восстановления**, який з допомогою підказок та налаштувань дозволяє виконати потрібну дію.

Програма **Дефрагментация диска** дозволяє оптимізувати розміщення файлів на диску. Кнопкою **Анализ** можна виконати перевірку розміщення файлів на диску у даний момент часу.

В деяких випадках потрібно виконати перевірку диска на наявність фізичних дефектів поверхні та логічних помилок у файлової системі. Це можна зробити командою **Проверка Диска**, яка розміщена на вкладці **Сервис** вікна властивостей диска. Відкрити це вікно можна із контекстного меню диска, наприклад, у програмі **Проводник**.

Перевірка дисків, як правило, відбувається після збоїв операційної системи або некоректного завершення її роботи.

Програма **Очистка диска** дозволяє вилучити на диску непотрібні файли, стиснути наявні файли для вивільнення дискового простору. Операційна система може автоматично виконати цю програму, якщо на диску залишилося недостатньо вільного місця для роботи.

Програма **Сведения о системе** дозволяє переглянути основні відомості про встановлені в системі пристрої, прикладні програми, драйвери, кодеки, ресурси апаратури, програмне середовище, параметри роботи в мережі тощо.

Окрім шрифтів з алфавітно-цифровими символами, в ОС можна використовувати спеціальні символні набори з додатковими елементами оформлення текстових документів, зокрема, грецькі літери, математичні символи, різноманітні логотипи, стрілки і т.д. Оскільки позначення цих символів на клавіатурі немає, потрібні спеціальні засоби для їх відображення і вставки в документ.

Програма **Таблица символов** (рис. 2.13) дозволяє побачити на екрані всі символи заданого набору і встановити, якій клавіші відповідає символ. У вікні програми вибирається потрібний шрифт, а потім символ, який автоматично збільшується. Вибрані символи можна скопіювати і вставити.

При деяких режимах інсталяції ОС Windows на ПК встановлюються не всі службові програми. Для додаткового запису потрібних програм необхідно виконати програму **Панель управління**, в якій вибрати пункт **Установка и удаление программ**. У вікні, що розкриється, треба зайти в пункт **Установка компонентов Windows**, вибрати необхідні компоненти та натиснути кнопку **Далее**.

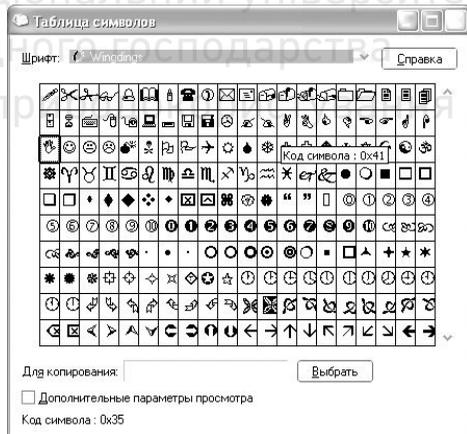


Рис. 2.13. Вікно програми Таблица символов

2.4.3. Програма Калькулятор



Програма **Калькулятор** призначена для простих обчислень та розрахунків з використанням математичних функцій. Для цього існує два режими роботи: **Обычный** та **Инженерный**. Перемикання між ними здійснюється в пункті меню **Вид**. У другому режимі можна виконувати статистичні розрахунки, обчислювати математичні, логічні та інші функції, використовувати круглі дужки для зміни порядку обчислень, переводити числа в різні системи числення. Для вставки результатів обчислень в інші додатки використовують буфер обміну.

Детальну інформацію про правила роботи з програмою **Калькулятор** та всі її можливості можна отримати з довідки, яка розміщена в програмі.

2.4.4. Програма Paint

Paint – найпростіший графічний редактор операційної системи Windows для роботи з растровими зображеннями.

У растровій графіці мінімальна одиниця зображення – точка, якій на екрані відповідає екранна точка з координатами і певним атрибутом кольору (піксель). Файл зображення у **Paint** стандартно зберігається у форматі BMP (нестиснений графічний формат, у якому кожній точці відповідає атрибут кольору), але малюнки можна зберігати і у файлах інших форматів. Основні елементи вікна програми **Paint** показані на рисунку 2.14.

Кнопки панелі інструментів використовуються для вибору графічних інструментів та способу малювання. **Кольорова палітра** служить для вибору основного кольору (ліва кнопка миші), яким виконується малювання, та кольору фону (права кнопка миші). Активний колір малювання зображено в лівій частині палітри, а під ним – колір фону.

При створенні нового малюнка спочатку потрібно задати розмір малюнка: **Рисунок / Атрибуты**. Розміри малюнка для друку краще задавати у сантиметрах, а для фону Робочого столу Windows - у пікселях, враховуючи роздільну здатність екрана. Для запису малюнка у файл необхідно послідовно виконати пункти меню **Файл / Сохранить (чи Сохранить как)**. При першому записі малюнка у файл потрібно вибрати диск, папку і задати назву та тип файлу.

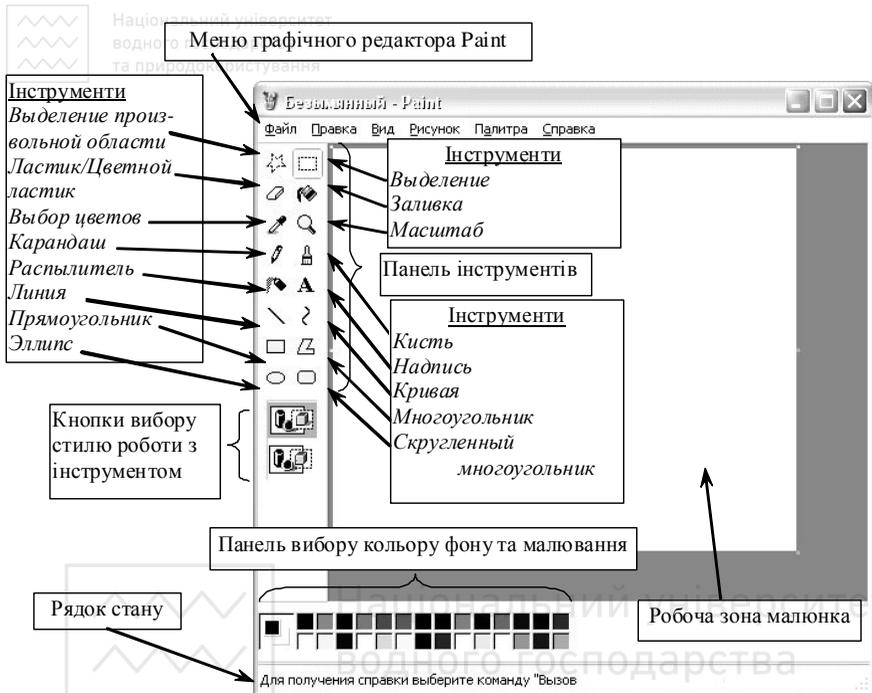


Рис. 2.14. Вікно програми Paint

Виконання робіт з інструментами

1. **Провести довільну лінію.** Вибрати інструмент **Карандаш**. Навести вказівник миші на початок потрібної лінії, натиснути ліву кнопку миші, перетягнути вказівник до кінця лінії та відпустити кнопку.
2. **Малювання за допомогою пензля.** Вибрати інструмент **Кисть** та встановити форму пензля. Навести вказівник миші на початок потрібної лінії, натиснути ліву кнопку миші, перетягнути вказівник до кінця лінії та відпустити кнопку.
3. **Малювання за допомогою розпилювача.** Вибрати інструмент **Распылитель**, а в ньому - форму розпилювача. Навести вказівник миші на початок потрібної лінії, натиснути ліву кнопку миші, перетягнути вказівник до кінця лінії та відпустити кнопку. Густина заповнення лінії залежить від швидкості заповнення.
4. **Провести пряму лінію.** Вибрати інструмент **Линия**. Встановити товщину та колір лінії. Навести вказівник миші на початок



потрібної лінії, натиснути ліву кнопку миші, перетягнути вказівник до кінця лінії та відпустити кнопку. Для малювання вертикальних, горизонтальних, під кутом 45° прямих ліній необхідно тримати натиснутою клавішу **Shift**.

5. **Провести криву.** Вибрати інструмент **Кривая**. Встановити товщину та колір лінії. Навести вказівник на початок потрібної лінії, натиснути ліву кнопку миші, перетягнути вказівник до кінця лінії та відпустити кнопку. З побудованої таким чином прямої лінії можна отримати дві дуги вигину. Для цього необхідно встановити вказівник у точку, де планується вершина дуги, натиснути ліву кнопку миші і змінити кривизну дуги шляхом переміщення вказівника.
6. **Намалювати еліпс.** Вибрати інструмент **Эллипс**. Встановити товщину лінії або вид заповнення, вибрати колір лінії або заповнення. Навести вказівник на початок потрібного еліпса, натиснути ліву кнопку миші, перетягнути вказівник до кінця еліпса та відпустити ліву кнопку миші. Для малювання кола необхідно тримати натиснуту клавішу **Shift**.
7. **Намалювати прямокутник (або прямокутник зі скругленими кутами).** Вибрати інструмент **Прямоугольник** (або **Скругленный прямоугольник**). Далі виконуємо дії, аналогічні малюванню еліпса.
8. **Намалювати багатокутник.** Вибрати інструмент **Многоугольник**. Встановити товщину лінії або вид заповнення, вибрати колір лінії або заповнення. Навести вказівник на початок потрібного багатокутника, клацнути лівою кнопкою миші. Далі необхідно перетягувати вказівник до кожного кута багатокутника та клацати лівою кнопкою миші. В останній вершині треба клацнути кнопкою миші двічі. Для того, щоб багатокутник мав кути 45 і 90 градусів, потрібно тримати натиснутою клавішу **Shift**.
9. **Заповнити замкнуту ділянку або об'єкт.** Вибрати інструмент **Заливка**. Встановити колір заливки. Навести вказівник миші у середину потрібної ділянки, натиснути ліву кнопку миші.
10. **Скопіювати колір об'єкта.** Вибрати інструмент **Выбор цветов**. Навести вказівник миші на потрібний колір об'єкта і натиснути ліву кнопку миші. Для вибору кольору фону натиснути праву кнопку миші.

11. **Очистити невелику ділянку.** Вибрати інструмент **Ластик**, а в ньому величину ластика. Протягнути вказівник миші по ділянці, яку потрібно очистити. Очищена ділянка буде заповнена кольором фону.
12. **Написати текст.** Вибрати інструмент **Надпись**. Встановити прозорий чи непрозорий фон. Намалювати на екрані прямокутник, у якому буде створюватися надпис, та набрати текст. Якщо потрібно змінити атрибути тексту, вмикають панель атрибутів тексту: **Вид / Панель атрибутів текста** та виконують необхідне форматування. Для закріплення тексту в малюнку необхідно клацнути вказівником миші у будь-якому місці екрана. Після цього повернутися до режиму редагування тексту неможливо.
13. **Відмітити прямокутний фрагмент малюнка.** Вибрати інструмент **Выделение**, встановити прозорий чи непрозорий фон. Перемістити вказівник миші в один з кутів фрагмента, який потрібно відмітити. Натиснути ліву кнопку миші і перетягнути вказівник миші до протилежного кута прямокутника.
14. **Відмітити довільний фрагмент малюнка.** Вибрати інструмент **Выделение произвольной области**. Встановити прозорий чи непрозорий фон. Перемістити вказівник миші на початок відмітки. Натиснути ліву кнопку миші і перетягувати вказівник миші до тих пір, поки не буде відмічено потрібний фрагмент.

Основні прийоми роботи

1. **Зняти відмітку фрагмента:** встановити вказівник миші поза відміченим фрагментом і клацнути лівою кнопкою миші.
2. **Скопіювати відмічений фрагмент малюнка: Правка / Копировать.**
3. **Вирізати відмічений фрагмент малюнка: Правка / Вырезать.**
4. **Вставити фрагмент малюнка з буфера: Правка / Вставить.** Можна перетягнути вставлений фрагмент у потрібне місце при натиснутій лівій кнопці миші. Для вставки декількох копій об'єкта слід натиснути клавішу **Ctrl** і не відпускати її доти, поки переміщення об'єкта не буде завершено.
5. **Зберегти відмічений фрагмент у файлі: Правка / Копировать в файл....** Вказати назву файлу та, якщо потрібно, шлях розміщення файлу.

- Національний університет
природокористування
6. **Очистити малюнок: Рисунок / Очистить.**
 7. **Змінити масштаб малюнка: Вид / Крупный або Вид / Другой, 100% (200%, 400%, 600%, 800%).** Інша можливість: інструмент **Масштаб**. При великому масштабі роботи з малюнком можна ввімкнути сітку: **Вид / Масштаб / Показать сетку**.
 8. **Збільшити розміри ділянки перегляду: Вид / Посмотреть рисунок.** Цей режим не призначений для зміни малюнка. Щоб повернутися в попередній режим, потрібно встановити вказівник у будь-яке місце малюнка і клацнути лівою кнопкою миші.
 9. **Відобразити чи повернути малюнок (фрагмент): Рисунок, Отразить/Повернуть.** Вибрати режим відображення або повороту.
 10. **Розтягнути чи нахилити малюнок (фрагмент): Рисунок, Растянуть/Наклонить.** Вибрати режим розтягування або нахилу.
 11. **Надрукувати малюнок: Файл / Печать.** Для попереднього перегляду малюнка перед друком вибирають: **Файл / Предварительный просмотр**. Для зміни параметрів сторінки: **Файл / Макет страницы**.

Виконання операцій сумісно з іншими програмами

1. **Копіювання відміченого фрагмента з одного документа в інший.**
 - У початковому документі відмітити фрагмент, який необхідно скопіювати.
 - Вибрати команду **Копировать** в меню **Правка**.
 - У документі, в якому необхідно розмістити скопійований фрагмент, навести вказівник миші в місце вставки і клацнути лівою кнопкою миші.
 - Виконати команду **Вставить** в меню **Правка**.
2. **Вставка малюнка з файлу в поточний малюнок.**
 - Відмітити ділянку, в яку потрібно вставити малюнок.
 - В меню **Правка** вибрати **Вставить из файла**.
 - Вказати ім'я файлу, вміст якого потрібно вставити.
 - Перетягнути вставлений фрагмент на потрібне місце.



Контрольні запитання і завдання

1. Дайте визначення операційної системи комп'ютера.
2. Призначення та основні функції операційних систем.
3. Яке основне призначення операційних систем реального часу?
4. Які є ресурси комп'ютера? Що називається процесом?
5. Яка структура файлової системи на носіях інформації?
6. Які є вимоги до імен файлів в операційних системах?
7. Для чого використовують розширення імені файлу? Які існують стандартні розширення імен файлів?
8. Для чого призначене контекстне меню?
9. Назвіть основні службові програми Windows та їх призначення.
10. З якою метою виконують дефрагментацію, оптимізацію і тестування дисків?
11. Для чого призначена програма **Проводник**?
12. Як створити нову папку?
13. Як скопіювати чи перенести файли в іншу папку?
14. Як перейменувати об'єкт? Як відмінити зміни?
15. Як у лівій частині програми **Проводник** розкрити або приховати вміст папки?
16. Як відкрити документ або виконати програму?
17. Вкажіть відмінність ярликів і значків.
18. Що називається вікном? Вкажіть способи зображення вікон.
19. Дати визначення документа у Windows.
20. Що таке папка? Як на Робочому столі створити ярлик, папку?
21. Як знищується непотрібний об'єкт? Як відмінити знищення?
22. Яке призначення програми **Paint**?
23. Як вказати розміри малюнка? Як вибрати основний колір малювання, колір фону?
24. Як намалювати вертикальну чи горизонтальну лінію?
25. Як скопіювати, перемістити відмічений фрагмент малюнка?
26. Як змінити товщину лінії, яка буде намальована?
27. Як скопіювати вміст активного вікна у буфер обміну та вставити вміст буфера у свій малюнок?



3.1. Файлові менеджери

Будь-яка ОС має функції роботи з файловою системою та файлами. В ОС Windows для цього використовується вбудований файл-менеджер **Проводник** (англ. Explorer).

***Файл-менеджер** – комп'ютерна програма, яка дозволяє виконувати наступні операції з файлами: створення, відкривання/програвання/запуск/перегляд, коригування, переміщення, перейменування, копіювання, вилучення, зміну властивостей, пошук файлів та призначення прав.*

Існує два види файлових менеджерів — **навігаційні** та **ортодоксальні**. Особливістю ортодоксальних менеджерів, крім відповідної моделі роботи, є наявність двох панелей.

До найбільш популярних файлових менеджерів належать: ДИСКoКомандир, FAR Manager, Konqueror, Midnight Commander, PowerDesk, Total Commander.

Konqueror - менеджер файлів, оглядач мережевих ресурсів, переглядач документів для стільникового середовища, що працює у більшості Unix-подібних ОС.

Midnight Commander (MC, у перекладі з англ. - “північний командир”) — один з консольних файлових менеджерів для Unix-подібних операційних систем. Його клоном можна вважати Norton Commander. Midnight Commander нині включено у більшість дистрибутивів Linux.

Total Commander (TC) — популярний файловий менеджер для операційної системи Windows, який раніше називався Windows Commander. Вікно Total Commander подане на рис. 3.1. Ця програма має широкий набір **функцій**, які дозволяють:

- здійснювати обслуговування файлів та папок (створення, відкривання / програвання / запуск / перегляд, коригування, переміщення, перейменування, копіювання, вилучення, зміну атрибутів, пошук файлів та призначення прав);
- використовувати клавіатурні скорочення, близькі до оригінальних Norton Commander з можливістю вільної їх зміни;
- підтримувати загальноприйняті скорочення, що діють в ОС Windows;

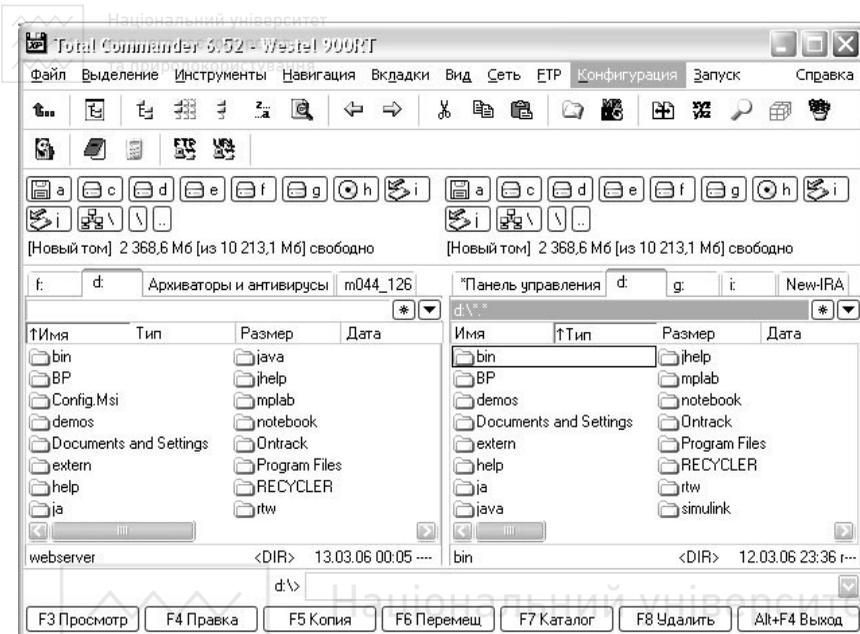


Рис. 3.1. Основне вікно файлового менеджера Total Commander

- застосовувати технологію Drag&Drop;
- працювати з архівами як з папками;
- використовувати вбудований ZIP-архіватор , а також розархіватори для архівів ZIP, RAR, ARJ та ін.;
- здійснювати розширений пошук файлів, включаючи пошук тексту в назві файлу, всередині файлу, пошук файлів за датою, властивостями, розміром, шаблонами тощо;
- переглядати файли, в тому числі, мультимедіа і графічні;
- використовувати панель інструментів та головне меню, які можна налаштувати;
- здійснювати підтримку системи **плагінів** (англ. plug in - вставляти, підключати, убудовувати):

- ✓ **архіваторних** — для підключення до ТС зовнішніх програм-архіваторів;
- ✓ **плагінів файлової системи**, які доступні через панель мережевого оточення;
- ✓ **плагінів внутрішньої програми перегляду**, які відображають спеціальні типи файлів у вбудованому



переглядачі, а також на панелі швидкого перегляду. За допомогою таких плагінів можна, зокрема, переглядати файли DBF, XLS, DOC, PDF тощо;

- ✓ **інформаційних плагінів**, призначених для вирішення декількох задач: пошук за специфічними властивостями і інформацією про вміст файлів, відображення цієї інформації у файлових списках тощо;
- копіювати файли у кілька потоків;
- здійснювати кодування / декодування і розрізання / склеювання довгих файлів;
- використовувати інтерфейс, що налаштовується, а також можливість перемикання мов програми та головного меню.

3.2. Методи роботи із стиснутими даними

Зберігання та передача інформації вимагає від учасників інформаційного процесу значних коштів. У зв'язку з цим виникає необхідність стискати дані перед їх розташуванням в архівах або передачею каналами зв'язку та відновлювати дані з архівів.

Характерною особливістю більшості класичних типів даних є надлишковість. Міра надлишковості залежить від типу даних. Наприклад, у відеоданих вона зазвичай вища, ніж у графічних даних, а графічні дані мають більшу надлишковість, ніж текстові дані. Крім того, міра надлишковості даних залежить від системи кодування.

Архівацією називають процес стискання даних.

Архіватор – програмний засіб, який використовують для стискання даних або для повернення їх до попереднього стану.

Існує три способи зменшення надлишковості даних: **зміна вмісту даних, зміна їх структури, зміна вмісту і структури.**

Якщо при стисканні даних відбувається зміна їх вмісту, метод стискання **незворотній** і при відновленні даних із стиснутого файлу не відбувається повного відновлення початкової послідовності. Такі методи називаються **методами стискання з регульованою втратою інформації**. Вони можуть бути застосовані до даних, для яких втрата частини вмісту не призводить до значного зниження їх властивостей, наприклад, до мультимедійних даних (відеоряди, музичні записи, звукозаписи, малюнки). Методи стискання із втратою інформації



забезпечують більш високу міру стиснення, але їх не можна застосовувати до текстових документів, баз даних, програмних кодів. Характерними форматами стиснення із втратою інформації є: **.jpg** – графічні дані, **.mpg** – відео, **.mp3** – звук.

Якщо при стисненні даних відбувається тільки зміна їх структури, то метод стиснення **зворотній**. Із результуючого коду можна відновити початковий масив шляхом застосування зворотного методу. Зворотні методи застосовують для стиснення будь-яких типів даних. Характерними форматами стиснення без втрат інформації є: **.gif**, **.tif**, **.psx** - графічні дані, **.avi** – відео, **.zip**, **.arj**, **.rar**, **.lzh**, **.lh**, **.cab** – будь-які дані.

Для зворотніх **методів** стиснення даних доведено наступні теореми:

*Для будь-якої послідовності даних існує теоретична **межа стиснення**, яка не може бути перевищена без втрати частини інформації.*

*Для будь-якого алгоритму стиснення можна вказати таку послідовність даних, для якої він забезпечить **кращу міру стиснення**, ніж інші методи.*

*Для будь-якого алгоритму стиснення можна вказати таку послідовність даних, для якої даний алгоритм взагалі **не дозволить отримати стиснення**.*

Сучасні засоби архівації даних використовують комбінації кількох теоретичних методів.

До основних функцій, які виконують більшість сучасних програм-архіваторів, належать:

- створення нового архіву;
- добування файлів з архіву;
- доповнення існуючого архіву файлами;
- вилучення файлів з архіву;
- створення архівів, що саморозпаковуються;
- створення багатотомних (розподілених) архівів;
- тестування цілісності структури архівів;
- відновлення пошкоджених архівів;
- захист архівів від перегляду та несанкціонованих змін шляхом встановлення пароля.



Додаткові функції програм-архіваторів дозволяють:

- переглядати файли довільних форматів, не витягуючи їх з архіву;
- здійснювати пошук файлів і даних всередині архіву;
- встановлювати програми з архівів без попереднього розпакування;
- перевіряти архів на наявність у ньому вірусів;
- здійснювати криптографічний захист архівної інформації;
- декодувати повідомлення електронної пошти;
- стискати виконувани (.exe) файли;
- вибирати або налаштовувати коефіцієнт стиснення інформації.

Нині найбільш поширеними можна вважати такі програми-архіватори: WinRAR, 7-Zip, GS CAB`er, KGB Archiver, WinAce.

Розглянемо послідовність створення та опрацювання архіву на прикладі однієї із сучасних програм – WinRAR. Вікно програми показано на рис. 3.2.

Для створення нового архіву необхідно:

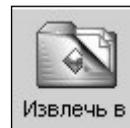
- відмітити потрібні файли;
- клацнути інструмент **Добавить**;
- у вікні **Имя и параметры архива** на вкладці



Общие вказати **назву архіву** (наприклад, архів1), клацнути кнопку **Обзор** для вибору **шляху реєстрації архіву**, вказати **тип архіву** (RAR чи ZIP), вибрати із списку **метод стиснення** (Обычный), при створенні багатотомного архіву **вказати розмір тому**. При необхідності встановити інші режими архівації. Клацнути кнопку **ОК**.

Для добування файлів з архіву можна:

- відмітити потрібні файли;
- вибрати інструмент **Извлечь в**;
- у вікні **Путь и параметры извлечения** вказати



шлях, а при необхідності вказати інші параметри, клацнути кнопку **ОК**.

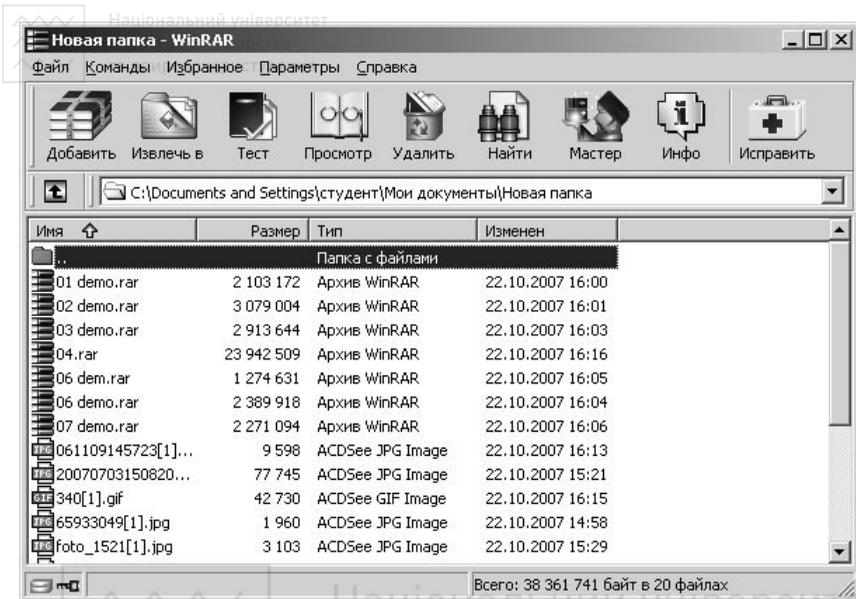


Рис. 3.2. Вікно програми WinRAR

3.3. Віруси та антивірусне програмне забезпечення

Комп'ютерний вірус - це невелика програма, здатна до саморозмноження і виконання різноманітних деструктивних дій.

Нині відомо понад 50 тис. комп'ютерних вірусів. Одним із перших вважається комп'ютерний вірус Brain, створений у Пакистані в 1986 році. Тільки у США цей вірус вразив понад 18 тис. комп'ютерів. На початку епохи комп'ютерних вірусів розробка вірусоподібних програм мала дослідницький характер, поступово перетворюючись на протистояння різних груп користувачів. Нині, за деякими оцінками, щоденно з'являється понад 60 нових вірусів. В ряді країн карне законодавство передбачає відповідальність за комп'ютерні злочини, у тому числі, за створення та розповсюдження вірусів.

Віруси діють тільки програмним шляхом. Вони, як правило, приєднуються до файлу або проникають всередину файлу. У цьому випадку кажуть, що файл заражений вірусом. Вірус потрапляє в комп'ютер тільки разом із зараженим файлом. Активізація вірусу відбувається після завантаження зараженого файлу. Деякі віруси під



час запуску зараженого файлу стають **резидентними** (постійно знаходяться в оперативній пам'яті комп'ютера) і можуть заражати інші файли та програми, що завантажуються. Інші різновиди вірусів відразу після активізації можуть спричинити серйозні пошкодження, наприклад, формувати жорсткий диск.

Дія вірусів може проявлятися по-різному: від візуальних ефектів, що заважають працювати, до повної втрати інформації. Більшість вірусів заражають виконавчі програми, тобто файли з розширенням **.exe** чи **.com**. Останнім часом віруси розповсюджуються переважно через систему електронної пошти.

Основними джерелами вірусів є:

- носій, на якому знаходяться заражені вірусом файли;
- комп'ютерна мережа, в тому числі, система електронної пошти та Internet;
- жорсткий диск, на який потрапив вірус під час роботи з зараженими програмами;
- вірус, що залишився в оперативній пам'яті після попереднього користувача.

Звідси випливає, що **зараження комп'ютера не відбудеться, якщо:**

- на комп'ютері переписуються тексти програм, документів, таблиць, файли даних системи керування базами даних (СКБД) тощо (ці файли не є програмами);
- виконується копіювання файлів з однієї дискети на іншу (проте в разі копіювання зараженого файлу його копія також буде заражена).

Основні ознаки зараження комп'ютера вірусом:

- зменшення обсягу вільної оперативної пам'яті;
- сповільнення завантаження та роботи комп'ютера;
- незрозумілі безпідставні зміни у файлах, а також зміни розмірів та дати останньої модифікації файлів;
- помилки при завантаженні операційної системи;
- неможливість зберігати файли у потрібних каталогах;
- незрозумілі системні повідомлення, звукові та візуальні ефекти тощо.

При переході вірусу в активну фазу можливі такі ознаки:

- зникнення файлів;
- форматування жорсткого диска;



- неспроможність завантаження файлів або операційної системи.

Величезну кількість різноманітних вірусів можна **класифікувати** умовно наступним чином:

1. **Завантажувальні** віруси (**boot-віруси**) заражають boot-сектори дисків. Ці віруси дуже небезпечні, бо можуть призвести до повної втрати всієї інформації, що зберігається на диску. Вірус записує початок свого тіла до boot-сектора, а решту – у вільні (інколи зайняті) кластери, помічаючи їх як пошкоджені. Туди ж вірус розміщує і справжній запис boot-сектора, щоб потім передати йому керування.

За своєю природою бутові віруси завжди резидентні. Деякі з них залишаються у пам'яті ПК навіть після “гарячого” перезавантаження (Ctrl+Alt+Del).

2. **Файлові** віруси заражають файли і поділяються на:

2.1. **Віруси, що заражають програми** (зокрема, файли з розширенням **.exe** і **.com**).

2.2. **Макровіруси**, що заражають файли даних, наприклад, документи Word або робочі книги Excel.

2.3. **Віруси-супутники**, які використовують імена інших файлів.

2.4. **Віруси сімейства DIR**, які спотворюють системну інформацію про файлові структури.

Прикладами таких вірусів є: Vanana, Burger, Druid та багато інших.

Найпоширенішим способом зараження файлу вірусом є дописування тіла вірусу на початок, у середину або в кінець файлу між фрагментами програмного коду або замість них.

При записі в середину файлу вірус інколи знаходить “порожні” місця і записує туди своє тіло, не змінюючи довжину жертви. У більшості ж випадків довжина інфікованого файлу збільшується на деяку величину, що, як правило, є постійною для вірусу, який заразив його. Ця величина зветься **довжиною** вірусу і вимірюється в байтах. Вірус може мати довжину від кількох десятків до десятків тисяч байт. У більшості випадків віруси пишуть мовою Асемблера, інколи мовами високого рівня (Pascal, C++ тощо). Існують віруси (наприклад, DICHOTOMY), які при зараженні записують частини свого тіла у два різних файли.

3. **Завантажувально-файлові** віруси здатні вражати як boot-сектор, так і файли.

4.

Віруси-невидимки (stealth-віруси) фальсифікують прочитану з диска інформацію так, що програма, якій призначена ця інформація, отримує неправильні дані. Ця технологія, яку інколи так і називають stealth-технологією, може використовуватися як у boot-вірусах, так і у файлових вірусах.

5. **Ретровіруси** заражають антивірусні програми, намагаючись знищити їх або зробити непрацездатними.

6. **Віруси-хробаки** заражають невеликі повідомлення електронної пошти так званім заголовком, який є web-адресою місцезнаходження самого вірусу. При спробі прочитати таке повідомлення вірус починає зчитувати через глобальну мережу Internet своє “тіло”, яке після завантаження починає деструктивну дію. Виявити цю різновидність вірусів дуже важко у зв'язку з тим, що заражений файл фактично не містить коду вірусу.

Для захисту інформації від вірусів використовуються загальні та програмні засоби, оскільки наслідки зараження можуть бути дуже серйозними.

До загальних засобів належать:

- резервне копіювання інформації (створення копій файлів і системних областей жорстких дисків);
- відмова від використання випадкових і невідомих програм;
- перезавантаження комп'ютера перед початком роботи, зокрема, у випадку, якщо за цим комп'ютером працювали інші користувачі;
- обмеження доступу до інформації, зокрема, фізичний захист носія під час копіювання файлів з нього.

До програмних засобів захисту належать різноманітні **антивірусні програми** (антивіруси).

Антивірус - це програма, яка виявляє і знешкоджує комп'ютерні віруси, запобігає їх розповсюдженню, а також лікує заражені ними програми.

Слід зауважити, що віруси у своєму розвитку можуть випереджати антивірусні програми, тому навіть у випадку регулярного застосування антивірусів немає повної гарантії безпеки. Антивірусні програми виявляють та знищують лише відомі віруси. При появі нового комп'ютерного вірусу захисту від нього не існує до тих пір, поки не буде розроблено відповідний антивірус. Однак, багато сучасних антивірусних пакетів мають у своєму складі спеціальний



програмний модуль, так званий **евристичний аналізатор**, який здатний досліджувати вміст файлів на наявність коду, характерного для комп'ютерних вірусів. Це дає змогу вчасно виявляти зараження новим вірусом.

Розрізняють такі типи **антивірусних програм**:

1. **Програми-детектори** виявляють файли, заражені відомими вірусами. Існують **спеціалізовані** (тобто призначені для боротьби з одним вірусом) детектори та **поліфаги** (можуть боротися з багатьма вірусами). Деякі програми-детектори можуть також лікувати файли від вірусів або знищувати заражені файли.

2. **Програми-лікарі** призначені для лікування заражених дисків і програм. Вони також можуть бути як спеціалізованими, так і поліфагами. Лікування програми полягає у вилученні із зараженої програми тіла вірусу.

3. **Програми-ревізори** призначені для виявлення ушкоджених файлів. Ці програми запам'ятовують дані про стан програми та системних областей дисків у нормальному стані (до зараження) і порівнюють ці дані у процесі роботи комп'ютера. В разі невідповідності даних виводиться повідомлення про можливість зараження.

4. **Лікарі-ревізори** аналізують зміни у файлах і системних областях дисків і у випадку їх виявлення повертають об'єкти в початковий стан.

5. **Програми-фільтри** призначені для перехоплення звернень до операційної системи, що використовуються вірусами для розмноження. Одержавши відповідне повідомлення, користувач має можливість дозволити або заборонити виконання певної операції. Такі програми є резидентними.

6. **Програми-вакцини** використовуються для обробки файлів і boot-секторів з метою попередження зараження відомими вірусами (останнім часом цей метод використовується все частіше).

Відмітимо, що вибір та застосування одного "найкращого" антивірусу не є оптимальним рішенням. Рекомендується використовувати декілька різних антивірусних пакетів одночасно.

Нині існує велика кількість різноманітних антивірусних програм, до основних завдань яких належать:

- сканування файлів і програм у режимі реального часу;
- сканування комп'ютера на вимогу;



- сканування інтернет-трафіка;
- сканування електронної пошти;
- захист від атак ворожих веб-вузлів;
- відновлення пошкоджених файлів (лікування).

До найбільш поширених в Україні антивірусів належать:

1. **DRWEB** вважається одним із кращих антивірусів. Поліфаг, здатний перевіряти файли в архівах, документи Word і робочі книги Excel, виявляти поліморфні віруси. Слід відмітити, 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. Програма викликається клацанням ЛКМ на ярлику, що має вигляд, поданий на рис. 3.3.

При завантаженні DrWeb у першу чергу 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.



Рис. 3.3

2. **ADINF** антивірус-ревізор диска. Він дозволяє знаходити та знищувати як звичайні, так і поліморфні віруси. Антивірус має в своєму розпорядженні лікувальний блок ревізора, який може знешкодити до 97% всіх вірусів. AdINF завантажується автоматично при вмиканні комп'ютера і контролює boot-сектор і файли на диску (дату і час створення, довжину, контрольну суму), виводячи повідомлення про їх зміни.

3. Антивірус **AVP** належить до поліфагів. У процесі роботи він перевіряє оперативну пам'ять, файли, в тому числі архівні, на гнучких, локальних, мережних і CD-ROM дисках, а також системні структури даних (завантажувальний сектор, таблицю розділів тощо). Програма має евристичний аналізатор, який, за твердженнями розробників антивірусу, здатний знаходити майже 80% усіх вірусів. AVP має зручний інтерфейс, а також одну з найбільших у світі антивірусну базу. Нові бази антивірусів AVP з'являються приблизно один раз на тиждень і їх можна отримати з Internet.

Популярними також є наступні антивірусні програми: NOD32, F-Secure Mobile Anti-Virus (антивірус для Pocket PC), Avast!, антивірус Касперського 7, AntiCAP, Anti-Trojan Shield.



3.4. Мультимедіа-програми

Мультимедіа означає подання інформації за допомогою декількох складових – тексту, звуку, графіки та відео.

Програми для опрацювання звуку

Цифровий звук – точна цифрова копія введених ззовні звуків. Одна хвилина цифрового звуку, записаного з дотриманням певного стандарту якості, займає близько 10 Мб дискового простору.

Програми для запису і опрацювання цифрової музики:

- CoolEdit
- SoundForge

Окрім цифрового існує ще й “**синтезований**” звук, тобто музика у форматі MIDI. MIDI-мелодії – це системи команд, що керують звуковою картою, тобто коди нот, які вона повинна відтворити (із вказуванням потрібних інструментів, тривалості та деяких інших параметрів кожної ноти). Файли з MIDI-музикою невеликі за обсягом. Їх легко коригувати, варіюючи інструменти, темп та стиль композиції.

Секвенсори – редактори MIDI-музики:

- Cakewalk Sonar
- Cubase VST
- Gigasampler

Третій вид звуку, з яким можна працювати за ПК, “**трекерна**” або “**семплерна**” технологія, яка передбачає конструювання музичної композиції із періодично повторюваних фрагментів цифрового або синтезованого звуку – петель або **семплів**.

Семплери і конструктори семплерних мелодій:

- ACID
- eJay
- FruityLoops

Програми для компресії цифрового звуку

Специфікація MPEG 1 LAYER III (MP3) дозволяє стискати звукову інформацію практично без втрати якості звуку. Стискання відбувається за рахунок ліквідації надлишкових сигналів, які не розрізняються вухом людини і при цьому не впливають на

“прозорість” звуку, або губляться, потрапляючи “в тінь” більш потужного сусіднього сигналу.

Існують різні міри стискання звукової інформації. Вони характеризуються швидкістю отриманого в результаті звукового потоку, що вимірюється в кілобітах за секунду. Ця величина називається **бітрейтом** (англ. bitrate). Після MP3-кодування бітрейт звукового потоку становить 56-320 кбіт/с. На практиці для збереження якості звуку використовуються бітрейти в діапазоні від 112 до 256 кбіт/с.

Крім MP3 існують інші формати стискання інформації, наприклад WMA. Сучасні WMA-кодеки дозволяють прослуховувати музичні композиції в режимі реального часу, забезпечуючи якість, аналогічну MP3-128 при бітрейті 48 кбіт/с.

Отже, для копіювання диска у формат MP3 потрібно застосувати наступні дві програми (окремо або в сукупності):

|| **Грабер** – система цифрового копіювання вмісту AudioCD на жорсткий диск.

|| **Кодер** – програма кодування отриманих файлів у MP3.

Приклади популярних граберів (укомплектованих кодерами): Audigraбber, CDex, EAC, WinDAC.

Програми-грабери, як правило, надають можливість користувачам швидко формувати плейлісти.

|| **Плейліст** – текстовий файл з розширенням M3U або PLS, що містить упорядкований список файлів з композиціями альбому.

Програми для роботи з відео

Зростання ємності жорстких дисків та швидкості процесорів, разом з переходом на цифрову технологію введення даних у ПК та появою нових алгоритмів стискання відеопотоку (MPEG-4), привели до “мультимедійної” революції.

До популярних програм для роботи з відео належать:

1. Adobe Premiere.

Програма орієнтована на захоплення та опрацювання як цифрового, так і аналогового відеопотоку, дозволяє працювати з великою кількістю незалежних відео- та аудіодоріжок (надає

можливість їх окремого коригування), забезпечує виконання багатшарового коригування (супровід кадрів різноманітними графічними елементами).

2. Adobe After Effect.

Програма об'єднує редактор, компоувальник і декоратор відео та звуку, дозволяє застосовувати усі види графіки (растрову, векторну, анімаційну), об'єднувати кілька відеодоріжок в одному кадрі, а також працювати із шарами.

3. Ulead MediaStudio Pro.

Програма містить наступні модулі:

- захоплення зображення із зовнішнього джерела;
- коригування окремих кадрів, їх ретушування та доповнення графічними елементами;
- створення спецефектів на базі векторної графіки, супровід анімаційними зображеннями та титрами;
- багатодоріжковий звуковий редактор із можливістю очистки звуку та доповнення його спецефектами;
- монтаж відео, звуку, анімаційних та графічних ефектів фільму в цілому;
- запис завершеного фільму на DVD.



Контрольні запитання і завдання

1. Що таке файл-менеджер? Назвіть відомі Вам файл-менеджери.
2. Опишіть можливості одного з ортодоксальних файл-менеджерів.
4. Що розуміють під поняттями архів, архівація, архіватор? З якою метою виконують архівацію?
5. Які основні дії можна виконувати із архівованою інформацією?
6. Назвіть найбільш популярні програми-архіватори.
7. Опишіть процедуру створення архіву, добування файлів з архіву.
8. Для чого виконують оцінку міри стискання файлів?
9. Що таке комп'ютерний вірус? Як відбувається зараження комп'ютера вірусом? За якими ознаками можна виявити факт зараження комп'ютерним вірусом?
10. Які Ви знаєте типи вірусів? Які деструктивні дії вони здійснюють?
11. Які заходи рекомендується вживати, щоб запобігти зараженню комп'ютерним вірусом?
12. Що таке евристичний аналізатор? Які функції він виконує?
13. Що таке антивірус? Які типи антивірусів Ви знаєте?
14. Наведіть приклади антивірусних програм.



4.1 Пакет програм MS Office

Microsoft Office – багатофункціональний комплект тісно інтегрованих офісних програм, призначених для використання організаціями і приватними користувачами. Він втілює сучасні засоби і досягнення, необхідні для організації офісної діяльності і розв’язання широкого класу інженерно-технічних і економічних задач. Універсальність програм та тісна взаємодія з веб-технологіями підвищує продуктивність роботи з пакетом, який є безперечним лідером серед програм подібного класу.

До складу MS Office входять : текстовий редактор Word, табличний процесор Excel, програма для підготовки презентацій PowerPoint, система керування базами даних (СКБД) Access, персональний інформаційний менеджер Outlook, редактор веб-сторінок FrontPage, видавнича система Publisher. Серед нових програм – пакет InfoPath для роботи з XML-формами (англ. Extensible Markup Language – універсальний стандарт обміну даними) та менеджер заміток OneNote.

Технологія використання та можливості перших чотирьох програм із наведеного переліку детально розглядається в цьому ж розділі. Тому дамо коротку характеристику інших програм MS Office.

MS Office Outlook – диспетчер особистих даних і програма зв’язку. Програма відіграє роль єдиного центру керування електронною поштою, календарями, контактами та ін.

MS Office FrontPage – програма для створення веб-вузлів і керування ними. Широке застосування функцій і елементів керування полегшує розробку веб-ресурсів, збільшує ефективність використання Інтернету завдяки можливостям функцій XML. Програма дозволяє створювати не лише статичні документи, а й інтерактивні веб-вузли.

MS Office Publisher – програма для підготовки публікацій, яка значно поліпшує якість оформлення і публікації інформаційних матеріалів. Створені матеріали можна друкувати, пересилати електронною поштою, розміщати в Інтернеті.

MS Office InfoPath – програма для збирання і обробки даних, яка дозволяє створювати динамічні форми, заповнювати їх і відправляти в XML-системи. В програмі забезпечується ефективний і гнучкий спосіб збирання та використання даних багатьма співробітниками чи робочими групами.



4.2. MS Word

Текстовий редактор **Microsoft Word** - програма для підготовки і опрацювання текстових документів.

Підтримка технології WYSIWYG (англ. What You See Is What You Get) дозволяє переглядати на екрані документ у тому ж вигляді, в якому він буде надрукований.

Запуск програми Microsoft Word

(далі у тексті - Word) здійснюється вибором відповідної команди у головному меню **Пуск / Програми / Microsoft Word** або використанням відповідного ярлика (рис. 4.1).



Рис. 4.1. Ярлик для запуску Microsoft Word

Вікно Word

Word дозволяє працювати паралельно з кількома документами, кожен із яких розташовується в окремому вікні. При введенні і редагуванні тексту користувач працює з активним документом в одному з вікон. Для переходу до іншого вікна необхідно клацнути його назву на лінійці задач або в меню **Окно**, яке містить список відкритих документів.

Рядок меню призначений для виклику команд Word. Він розміщений, як правило, безпосередньо під заголовком вікна (рис. 4.2). Для вибору команди необхідно клацнути ЛКМ на назві потрібного пункту. Після цього з'явиться список тих команд вибраного пункту, що використовуються найчастіше (рис. 4.3). Якщо клацнути на кнопці  наприкінці списку команд, то з'являться всі команди пункту меню.

Панелі інструментів являють собою набори кнопок із малюнками. Інструменти дублюють команди системи меню, які часто використовуються. Для виклику команди, зв'язаної з кнопкою, необхідно клацнути ЛКМ на цій кнопці. При наведенні вказівника миші на кнопку, поруч з нею може з'явитися рамка із назвою команди.

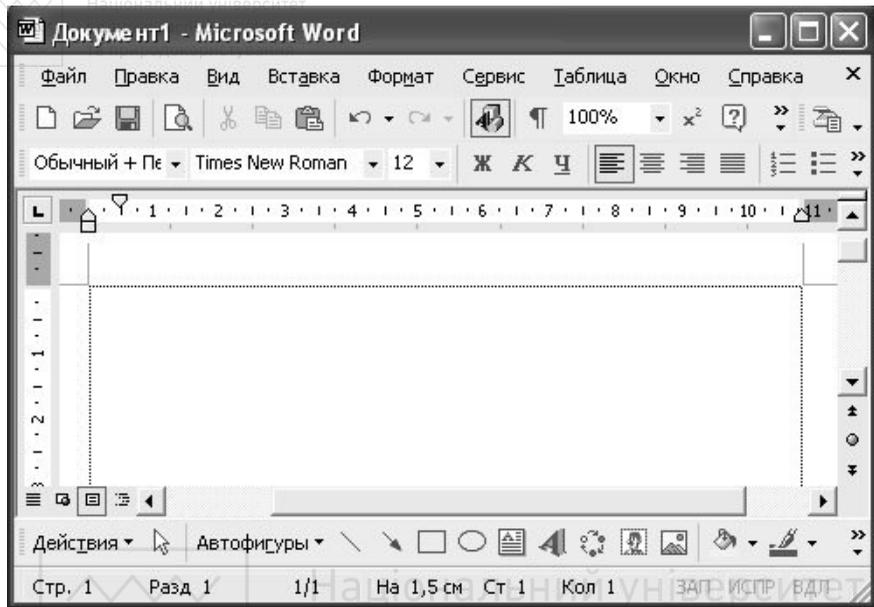


Рис. 4.2. Вікно програми Word

За замовчуванням під рядком меню розміщені дві панелі інструментів: **Стандартная** и **Форматирование**. Ще одна панель **Рисование** виводиться внизу вікна безпосередньо над рядком стану.

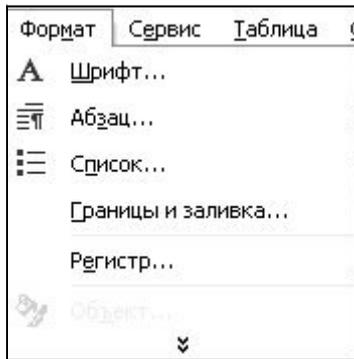


Рис. 4.3. Список деяких команд пункту меню Формат

Для вмикання чи вимикання панелей потрібно вибрати команду меню **Вид / Панели инструментов** і клацнути в рядку з потрібною назвою. Перед назвами ввімкнутих панелей розміщені помітки **✓**.

Якщо для відображення всіх кнопок на панелі недостатньо місця, то спочатку виводяться ті з них, які були використані останніми. При натисканні  наприкінці панелі з'являться інші кнопки, як це показано на рис. 4.4.

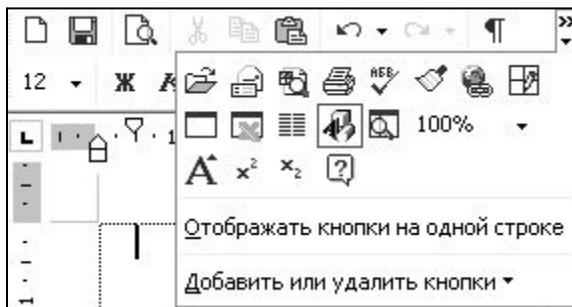


Рис. 4.4. Кнопки панелі інструментів Стандартная

При клацанні на кнопці **Добавить или удалить кнопки** з'явиться меню, в якому можна ввімкнути чи вимкнути будь-яку із кнопок, передбачених для цієї панелі.

Для розташування на одній з панелей інструментів кнопки, що входить до іншого набору, потрібно вибрати в меню **Сервис / Настройка**. У вікні діалогу переходимо на вкладку **Команды**. У списку **Категории** можна вибрати потрібну групу, після чого в списку **Команды** з'являться відповідні кнопки. Щоб доповнити кнопкою панель інструментів, досить транспортувати її з діалогового вікна у потрібну позицію панелі. Процес завершується клацанням кнопки **Закреть**. Для вилучення кнопки з панелі інструментів необхідно перетягнути її мишею в діалогове вікно **Настройка**.

Над робочим полем розташована горизонтальна, а зліва від робочого поля - вертикальна **координатні лінійки**. За їх допомогою зручно встановлювати поля сторінок, абзацні відступи, змінювати ширину стовпців і встановлювати позиції табуляції. За замовчуванням координатна лінійка градуєвана в сантиметрах. Виводяться чи забираються лінійки за допомогою команди **Вид / Линейка**.

Смуги прокручування (вертикальна і горизонтальна) призначені для переміщення по тексту у вертикальному чи горизонтальному напрямку. Переміщення в тексті з використанням смуг прокручування здійснюється за допомогою ЛКМ.



Редактор Word дає можливість переглядати документ у різних режимах:

-  **Обычный** – використовується для виконання більшості операцій;
-  **Web-документ** - відображає документ у вигляді Web-сторінки;
-  **Разметка страницы** – показує документ у точній відповідності з тим, як він буде виведений на друк. У цьому режимі зручно працювати з колонтитулами, фреймами і багатоколонковою версткою документа; тільки в цьому режимі відображається вертикальна координатна лінійка;
-  **Структура** - призначений для роботи із структурою документа: дозволяє показувати і приховувати текст і заголовки різної глибини вкладеності, створювати і коригувати підпорядковані документи.

Зміна режиму перегляду здійснюється за допомогою відповідних команд меню **Вид** або кнопок, розташованих зліва від горизонтальної смуги прокручування (рис.4.2).

Рядок стану розташований у нижній частині вікна Word. В ньому виводяться повідомлення і довідкова інформація (таблиця 4.1).

В процесі виконання тих чи інших дій у текстовому редакторі Word зручно використовувати так звані **контекстні меню**. Для виклику контекстного меню (рис. 4.5) потрібно клацнути правою кнопкою миші (ПКМ) на вибраному об'єкті. Контекстне меню з'являється біля вказівника миші. Воно містить команди для роботи з вибраним об'єктом.

Завершення роботи у Word здійснюється виконанням команди меню **Файл / Выход** або клацанням кнопки закриття вікна  чи натисканням комбінації клавіш **Alt+F4**.



Інформація у рядку стану

Індикатор	Значення
Стр. 1	Порядковий номер видимої сторінки документа
Разд 1	Номер розділу, в якому знаходиться видима сторінка
1/1	Номер видимої сторінки / загальне число сторінок у документі
На 1,5 см	Відстань від курсора до верхнього краю сторінки
Ст 1	Номер рядка, в якому знаходиться курсор
Кол 1	Номер позиції курсора в рядку
ЗАП	Індикатор режиму запису макрокоманди
ИСПР	Індикатор режиму редакторської правки
ВДЛ	Індикатор режиму розширення маркування
ЗАМ	Індикатор режиму заміни
украинский	Індикатор мови

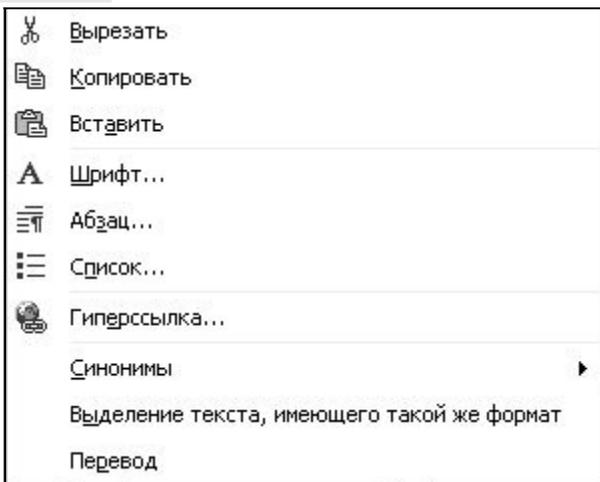


Рис. 4.5. Контекстне меню для обробки тексту



Робота з документами

Для створення нового документа необхідно виконати команду меню **Файл / Создать** чи клацнути кнопку . Після цього у вікні Word справа з'явиться область задач (рис. 4.6). Її можна використати для створення нового звичайного документа, підготовки документа на основі одного із вже існуючих чи розробки документа з використанням шаблонів. В останньому випадку в області задач спочатку треба вибрати команду **Общие шаблоны...**, потім вкладку, наприклад, **Общие**, потім **шаблон** документа, після чого клацнути кнопку **ОК**. Шаблони документів Word мають розширення **.dot**.

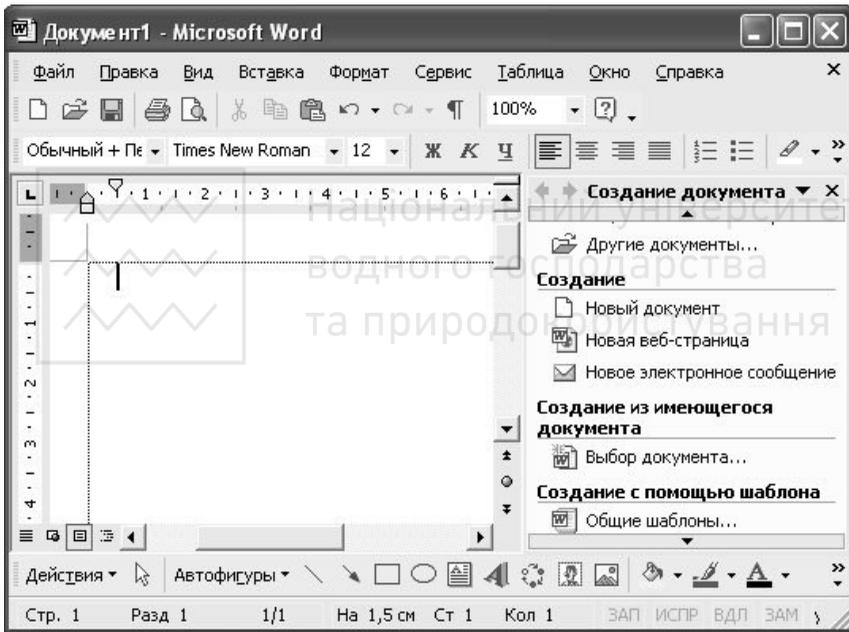


Рис. 4.6. Вікно Word, що містить документ та область задач (справа)

Для відкриття існуючого документа необхідно виконати команду меню **Файл / Открыть** чи клацнути кнопку . У полі **Папка** діалогового вікна **Открытие документа** (рис. 4.7) вибираємо диск, що містить документ. У полі, яке розташоване нижче, можна розгорнути папку, двічі клацнувши на ній ЛКМ, і вибрати документ. Документи Word мають розширення **.doc** і характерний для редактора значок. Зображення одного з них можна побачити на рис. 4.7.

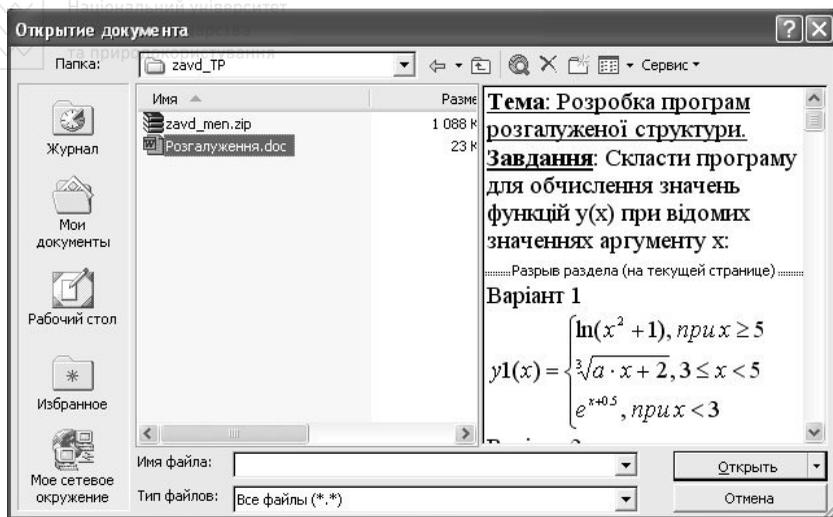


Рис. 4.7. Вікно діалогу Открытие документа

У верхньому рядку діалогового вікна серед інших кнопок є кнопка, що дозволяє подати вміст відкритої папки за одним із кількох варіантів. Вона має вигляд  і називається **Представлення**. При виборі опції **Просмотр** користувач у правій частині вікна може побачити фрагмент документа ще до відкриття відповідного файлу.

Якщо користувач не бачить у своїй папці зареєстрованих там документів, він повинен перевірити, що саме вказано у полі **Тип файлів**, оскільки можлива ситуація виведення даних про файли не відповідного формату.

Для зберегання документа необхідно викликати команду меню **Файл / Сохранить** чи клацнути кнопку . При першому зберіганні даних з'явиться діалогове вікно **Сохранение документа** (рис. 4.8). В полі **Папка** потрібно вибрати диск і розгорнути папку, в якій необхідно зберегти документ. У полі **Тип файла** вказати формат для зберігання документа. За замовчуванням це **Документ Word (.doc)**. У полі **Имя файла** вказати назву файлу і клацнути кнопку **Сохранить**.

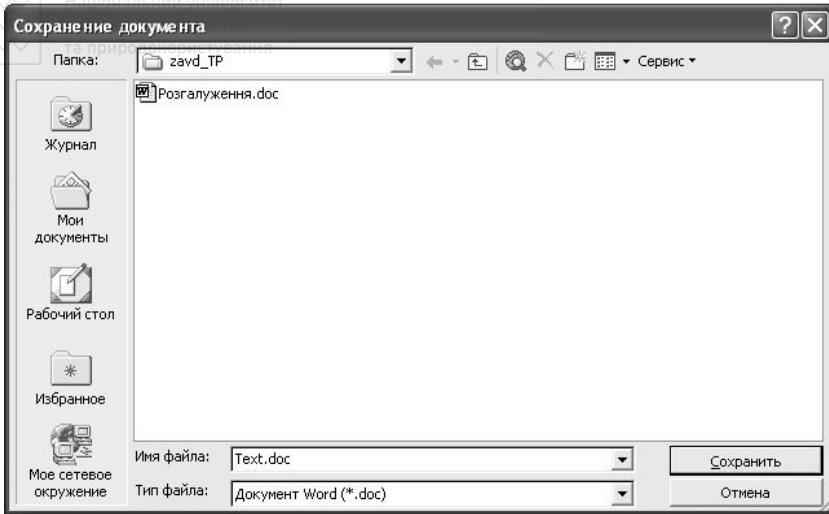


Рис. 4.8. Вікно діалогу Сохранение документа

При повторному зберіганні даних вікно **Сохранение документа** не виводиться, а документ автоматично зберігається у тому ж файлі. Щоб зберегти документ з іншим ім'ям або в іншій папці, потрібно в меню **Файл** вибрати команду **Сохранить как...**

Режими введення тексту

Існує два **режими введення тексту** - вставка і заміна. У режимі **вставки** при введенні нових символів текст, що міститься в документі, переміщується вправо від місця введення. У режимі **заміни** попередній текст замінюється новим. Зміна режимів здійснюється подвійним клацанням на індикаторі **ЗАМ** у рядку стану або натисканням клавіші **Insert** на клавіатурі.

Опрацювання текстів

У процесі роботи користувач постійно орієнтується на текстовий курсор і вказівник миші.

Текстовий курсор - мерехтлива вертикальна риска |, що вказує місце введення чи коригування тексту.

Досягнувши правої границі тексту, курсор автоматично переходить на початок наступного рядка. Для переходу на початок наступного абзацу потрібно натиснути **Enter**.



Для переміщення курсора у тексті використовуються клавіші керування курсором чи їх комбінації (таблиця 4.2) або клацання ЛКМ.

Таблиця 4.2.

Клавіші та їх комбінації для керування курсором

Клавіші та їх комбінації	Переміщення
↑ (↓)	На один рядок вгору (вниз)
← (→)	На одну позицію вліво (вправо)
Ctrl+↑ (Ctrl+↓)	На один абзац вгору (вниз)
Ctrl+← (Ctrl+→)	На одне слово вліво (вправо)
PgUp (PgDn)	На один екран вгору (вниз)
End (Home)	У кінець (на початок) рядка
Ctrl+Home (Ctrl+End)	На початок (у кінець) документа

Вставка символів, відсутніх на клавіатурі

Для вставки у текст символу, відсутнього на клавіатурі, необхідно:

- встановити курсор у позицію, в яку потрібно вставити символ;
- вибрати команду меню **Вставка / Символ**;
- у діалоговому вікні **Символ** (рис. 4.9) вибрати вкладку **Символи**;
- у полі **Шрифт** вибрати тип шрифту;
- клацнути мишею на потрібному символі в таблиці;
- клацнути кнопку **Вставити**;
- для завершення роботи з вікном **Символ** потрібно клацнути кнопку **Закрити**.

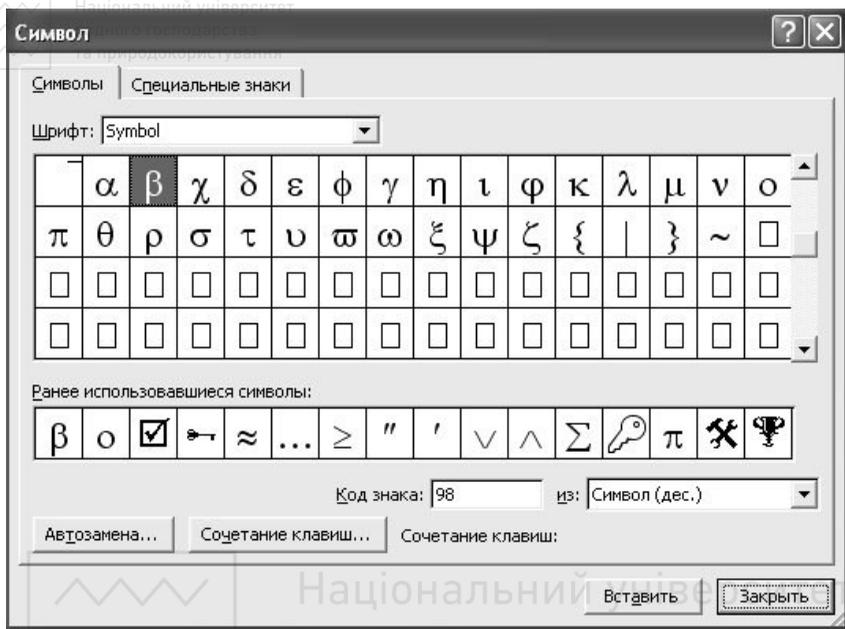


Рис. 4.9. Вкладка Символ вікна діалогу Символ

Виділення фрагмента тексту

Перед тим, як виконати якусь операцію з фрагментом тексту, його необхідно **виділити** одним із таких способів:

- встановити вказівник миші в ліве поле документа (він перетвориться на стрілку, спрямовану вправо). При натисканні ЛКМ виділиться один рядок, при натисканні її двічі - абзац, при натисканні тричі - весь документ;
- встановити вказівник миші в ліве поле навпроти першого рядка фрагмента, натиснути ЛКМ і, не відпускаючи її, розтягнути виділення на весь фрагмент;
- встановити вказівник миші на початку фрагмента, натиснути ЛКМ і, не відпускаючи її, розтягнути виділення на весь фрагмент;
- для виділення одного слова досить двічі клацнути на ньому ЛКМ;
- для виділення одного абзацу можна клацнути у ньому тричі ЛКМ;



• для виділення одного речення можна натиснути клавішу **Ctrl** і клацнути мишею у реченні;

- для виділення всього тексту досить натиснути клавішу **Ctrl** і клацнути мишею на лівому полі;
- щоб виділити фрагмент тексту за допомогою клавіатури, необхідно встановити курсор на початок фрагмента і, натиснувши клавішу **Shift**, клавішами керування курсором розтягнути виділення на весь фрагмент.

Зняти виділення можна клацанням ЛКМ у будь-якому місці тексту. При виділенні нового фрагмента попереднє виділення знімається.

Редагування тексту

Символ справа від курсора вилучається клавішею **Delete**, символ зліва від курсора - клавішею **Backspace**. Якщо виділити фрагмент тексту і набрати з клавіатури новий текст, він вставиться на місце виділеного фрагмента.

Зауваження: Якщо вставка нового тексту замість виділеного не відбувається, потрібно виконати команду **Сервіс / Параметри...** перейти на вкладку **Правка** та поставити відмітку біля опції **замінять виділений фрагмент**.

Щоб розділити абзац на два, необхідно встановити курсор в кінець першого абзацу і натиснути клавішу **Enter**.

Об'єднати два абзаци в один можна двома способами:

- встановити курсор за останнім символом першого абзацу і натиснути клавішу **Delete**;
- встановити курсор перед першим символом другого абзацу і натиснути клавішу **Backspace**.

При натисканні клавіші **Enter** у поточну позицію курсора вставляється невидимий символ ¶. Для вмикання чи вимикання режиму перегляду невидимих символів використовується кнопка . Невидимі символи знищуються, як і звичайні, клавішами **Delete** і **Backspace**.

Для вилучення фрагмента тексту можна виділити його і натиснути клавішу **Delete**.



Копіювання тексту

Для копіювання фрагмента тексту необхідно:

- виділити фрагмент тексту;
- клацнути кнопку  або вибрати команду меню **Правка / Копировать** чи виконати команду **Копировать** контекстного меню;
- поставити курсор у місце, куди потрібно вставити фрагмент;
- клацнути кнопку  або вибрати команду меню **Правка / Вставить** чи виконати команду **Вставить** контекстного меню.

У процесі цієї операції копія виділеного фрагмента тексту поміщається у буфер тимчасового зберігання (**Clipboard**), а потім потрапляє у документ. Вставляти фрагмент із буфера можна декілька раз, але після копіювання в буфер нового фрагмента тексту попередній фрагмент звідти може зникати.

Переміщення тексту

Для переміщення фрагмента тексту необхідно:

- виділити фрагмент тексту;
- клацнути кнопку  або вибрати команду меню **Правка / Вырезать** чи виконати команду **Вырезать** контекстного меню;
- поставити курсор у місце, куди потрібно вставити фрагмент;
- клацнути кнопку  або вибрати команду меню **Правка / Вставить** чи виконати команду **Вставить** контекстного меню.

Перемістити фрагмент тексту можна також наступним чином:

- виділити фрагмент тексту;
- транспортувати його в потрібне місце (місце вставки вказують курсором).

Якщо при цьому тримати натиснутою клавішу **Ctrl** (біля вказівника миші з'явиться знак +), то фрагмент буде скопійовано.

Зауваження: Якщо транспортування виділеного тексту не відбувається, потрібно попередньо виконати команду **Сервис**



Скасування та повтор виконаних дій

Для відміни останньої операції редагування необхідно вибрати команду меню **Правка / Отменить ...** або клацнути кнопку . Якщо клацнути на стрілці, то відкриється список операцій, виконаних у поточному сеансі. Клацнувши на імені однієї операції, можна відмінити її і всі операції, виконані після неї.

Щоб повернути останню відмінену операцію, потрібно в меню **Правка** вибрати команду **Повторить ...** або клацнути кнопку . Для перегляду списку відмінених операцій потрібно клацнути на стрілці.

Відміну дій можна виконувати клавішною комбінацією **Ctrl+Z**.

Використання буфера обміну

У Word існує **буфер обміну** на 24 комірки, за допомогою якого можна копіювати фрагменти тексту не тільки в межах Word, але і в інші програми, наприклад, у Microsoft Excel.

Для виведення буфера обміну (рис. 4.10) необхідно вибрати команду меню **Правка / Буфер обмена Office...**

Щоб скопіювати фрагмент у буфер, його необхідно виділити і клацнути на кнопці .

Для вставки фрагмента з буфера у позицію курсора необхідно клацнути на потрібному значку фрагмента.

Якщо фрагмент скопійовано із Word, то він буде мати значок .

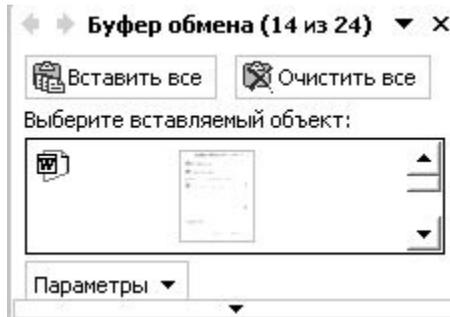


Рис. 4.10. Буфер обміну Word

Пошук і заміна тексту

Для пошуку фрагмента тексту використовується команда меню **Правка / Найти**. В діалоговому вікні **Найти и заменить** (рис. 4.11) у полі **Найти** треба ввести текст для пошуку і клацнути **Найти далее**.

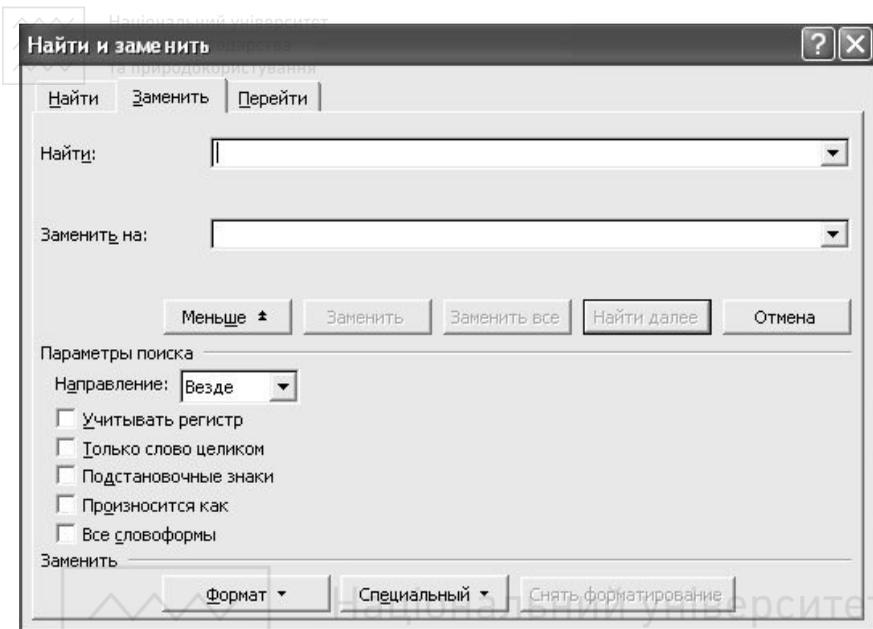


Рис. 4.11. Вкладка Заменить вікна діалогу Найти и заменить

При необхідності можна клацнути кнопку **Больше ▼** і ввести додаткові умови пошуку. Кнопка **Больше ▼** при цьому буде замінена на кнопку **Меньше ▲**. У полях параметрів пошуку, що з'являться, вказують:

- **Направление** - напрям пошуку (**Везде** – шукати у всьому документі; **Вперед** – шукати в тексті після курсора; **Назад** – шукати в тексті до курсора);
- **прапорці режимів пошуку** (**Учитывать регистр** – при пошуку розрізняти великі і малі літери; **Только слово целиком** – пошук тільки тих слів, що цілком збігаються із зазначеним; **Подстановочные знаки** – використовуються символи шаблону, що вибираються після натискання кнопки **Специальный**).

Щоб знайти наступне слово із вибраними умовами, необхідно клацнути кнопку **Найти далее**.

Для заміни одного фрагмента тексту іншим можна вибрати вкладку **Заменить** діалогового вікна **Найти и заменить** або вибрати команду меню **Правка / Заменить**. На вкладці **Заменить** слід ввести умови



- в полі **Найти:** ввести фрагмент тексту, який необхідно замінити;
- в полі **Заменить на:** ввести фрагмент тексту для заміни;
- клацнути кнопку **Найти далее;**
- для заміни знайденого слова клацнути кнопку **Заменить;**
- для заміни усіх фрагментів, що задовольняють умовам, клацнути кнопку **Заменить все.**

Форматування тексту та окремих символів

Перед зміною параметрів тексту його потрібно виділити. Якщо текст не буде виділений, то змінюються поточні параметри, тобто параметри тексту, який буде вводиться далі з поточної позиції.

Форматування тексту - процедура зміни параметрів тексту, що визначають його зовнішній вигляд.

У текстовому редакторі Word існують поняття форматування окремих символів, форматування абзаців і форматування сторінок.

Для зміни параметрів окремих символів виконують команду меню **Формат / Шрифт**, що викликає діалогове вікно **Шрифт** (рис. 4.12). Встановити параметри шрифту можна також за допомогою кнопок на панелі інструментів **Форматирование** (див. рис. 4.2). Вкладка **Шрифт** використовується для встановлення параметрів шрифту, описаних у таблиці 4.3.

Для зміни інтервалу і положення символів використовується вкладка **Інтервал** діалогового вікна **Шрифт**. У полях на цій вкладці вказують наступне:

- **Масштаб** - міра розтягування або стискання символів (за замовчуванням 100%);
- **Інтервал** - відстань між окремими символами, вказана у пунктах: **Обычный, Разреженный, Уплотненный** (за замовчуванням – Обычный);
- **Смещение** - зміщення символів по вертикалі (у пунктах): **Нет, Вверх, Вниз** (за замовчуванням – Нет).

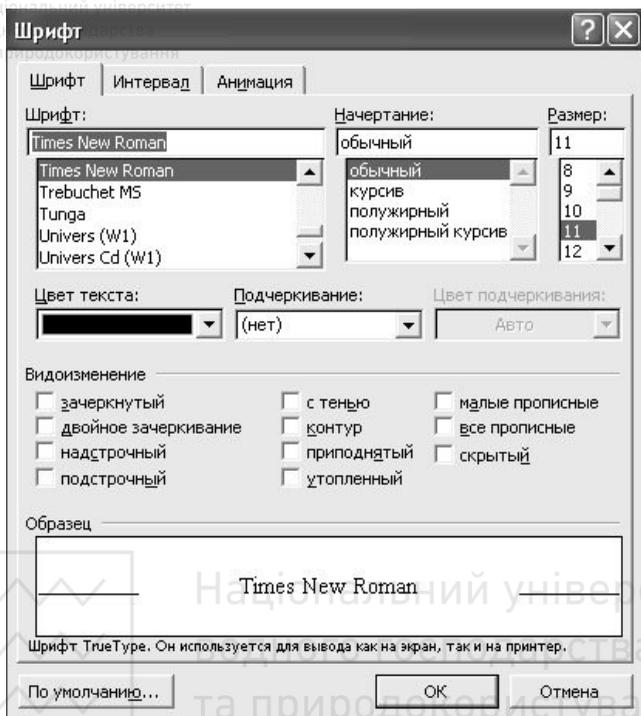


Рис. 4.12. Вкладка Шрифт вікна діалогу Шрифт

Для зміни регістра символів у набраному тексті необхідно виділити фрагмент тексту і вибрати в меню команду **Формат / Регистр...**. У діалоговому вікні можна вибрати один із таких перемикачів:

- **Как в предложениях** – зробити великою першу літеру першого слова речення;
- **все строчные** – перевести всі літери фрагмента в нижній регістр;
- **ВСЕ ПРОПИСНЫЕ** – перевести всі літери фрагмента у верхній регістр;
- **Начинать С Прописных** – перевести перші літери кожного слова у верхній регістр;
- **иЗМЕНИТЬ РЕГИСТР** – замінити літери верхнього регістра літерами нижнього регістра і навпаки.



Об'єкти вкладки Шрифт вікна Шрифт

Об'єкт	Призначення
Шрифт	Вибрати шрифт (інструмент на панелі )
Начертание	Вказати один із варіантів зображення шрифту: обычный , курсив (інструмент  на панелі), полужирный (інструмент  на панелі) чи полужирный курсив
Размер	Вказати розмір шрифту у пунктах (інструмент на панелі ). 1 пт = 0,353мм
Цвет текста	Призначити колір символів
Подчеркивание	Вибрати тип лінії підкреслення
Цвет подчеркивания	Встановити колір лінії підкреслення
Видоизменение	Змінити зовнішній вигляд символів прапорцями
Образец	Показує фрагмент тексту з обраними параметрами

Форматування абзаців

Для встановлення параметрів абзацу виконують команду меню **Формат / Абзац...**. Після її вибору з'являється діалогове вікно **Абзац** (рис. 4.13).

Для встановлення абзацних відступів і інтервалів необхідно вибрати вкладку **Отступы и интервалы**.

У полі **Выравнивание:** встановлюється один із можливих способів вирівнювання тексту абзацу: **По левому краю**, **По центру**, **По правому краю**, **По ширине**. В полях **слева:** и **справа:** встановлюються відстані від лівого і правого полів до меж абзацу.

У полі **первая строка:** вказується вид та розмір у пунктах відступу першого рядка абзацу: **нет**, **Отступ**, **Выступ**.

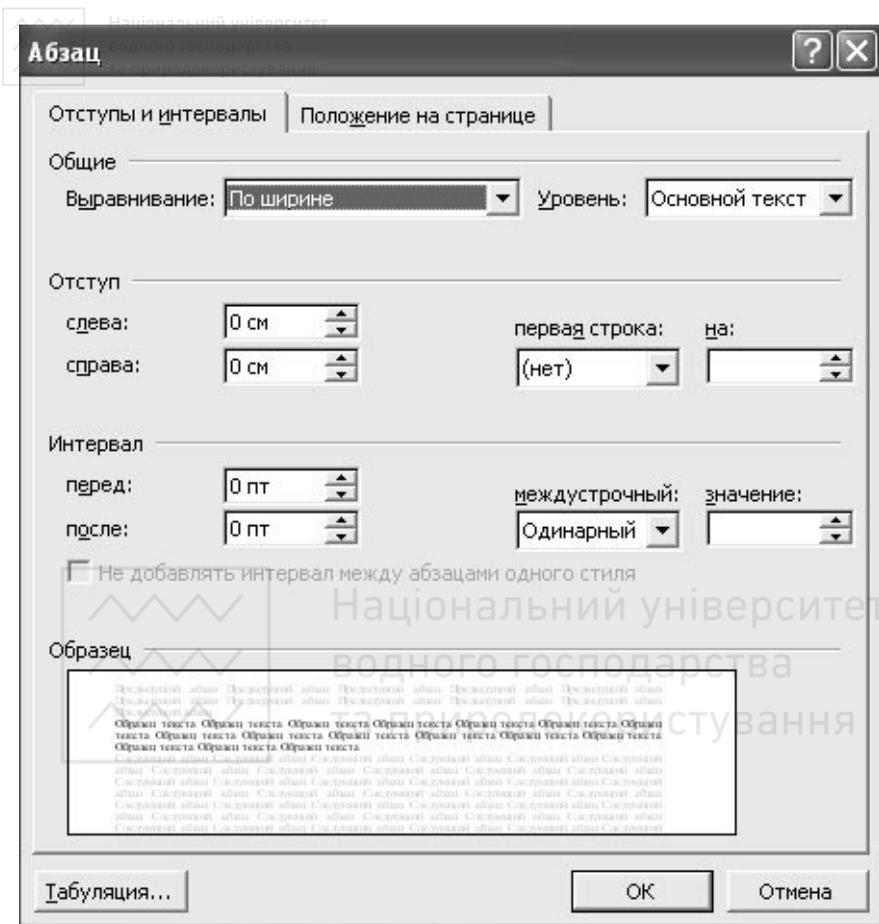


Рис. 4.13. Вкладка Отступы и интервалы вікна діалогу Абзац

В полях **перед:** і **после:** задають відстані відповідно перед першим рядком абзацу і після останнього його рядка.

У полі **междустрочный:** встановлюють інтервал між рядками всередині абзацу: **Одинарный**, **Полуторный**, **Двойной**, **Минимум**, **Точно** чи **Множитель**. При цьому **Полуторный** обчислюється з коефіцієнтом 1,5 відносно одинарного, **Двойной** – з коефіцієнтом 2, а **Множитель** – із вказаним користувачем коефіцієнтом. **Точно** передбачає вказування попередньо підрахованого значення інтервалу, зазначеного у пунктах. **Минимум** означає, що інтервал не буде перевищувати найменшого допустимого.



Встановлювати тип вирівнювання набагато швидше і зручніше за допомогою кнопок  **По левому краю**,  **По центру**,  **По правому краю** та  **По ширине** панелі інструментів **Форматирование**.

Встановлювати відступи усіх та першого рядків абзацу можна за допомогою спеціальних **повзунків**, розташованих на горизонтальній координатній лінійці (їх добре видно на рис. 4.2). **Верхній** повзунок відповідає за відступ першого рядка абзацу, **нижній лівий** повзунок – за ліву границю, а **нижній правий** повзунок – за праву границю тексту абзацу.

Форматування сторінок

Для встановлення параметрів сторінки використовується команда меню **Файл / Параметри сторінки**, яка викликає діалогове вікно **Параметри сторінки**.

Для встановлення полів сторінок призначена вкладка **Поля** (рис. 4.14).

Під полями розуміють не заповнені текстом ділянки на краях сторінки.

Встановлювати можна **Верхнее, Нижнее, Левое** та **Правое** поля.

Призначена орієнтація сторінки **книжная** чи **альбомная** підсвічується синьою рамкою.

При розташуванні кількох сторінок на одному аркуші, у полі **несколько страниц**: замість опції **Обычный** можна вказувати один із таких варіантів: **Зеркальные поля, 2 страницы на листе** і **Брошюра**. При виборі елемента списку **Зеркальные поля** замість полів **Правое**: і **Левое**: з'являться поля **Внутри**: і **Снаружи**:

У полі **Переплет**: встановлюється ширина поля підшивки. Можна вибрати також розташування поля підшивки **Слева** чи **Сверху**.

Щоб зазначити, якої частини документа стосуються обрані параметри, в полі **Применить**: слід обрати потрібне значення: **ко всему документу** чи **до конца документа**.

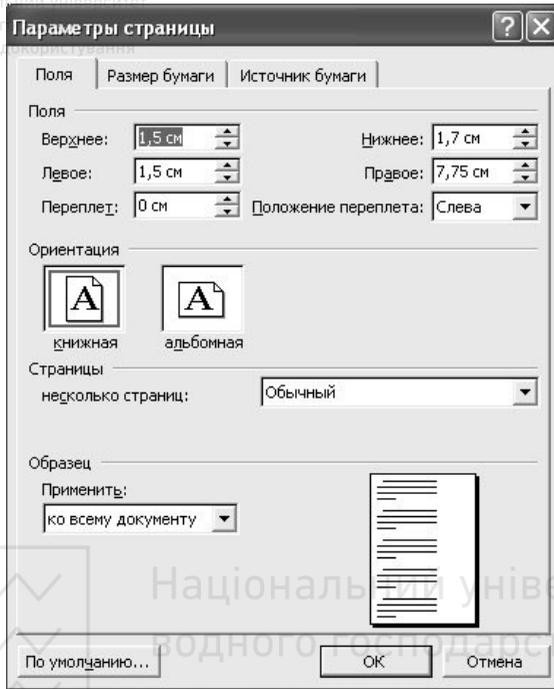


Рис. 4.14. Вкладка Поля вікна Параметри сторінки

У рамці **Образец** схематично показано зовнішній вигляд сторінки з обраними параметрами.

Встановити поля сторінки можна також за допомогою координатних лінійок у режимі **Разметка страниц**. На лінійках поля сторінки позначені сірим кольором. Необхідно встановити вказівник миші на межу сірої і білої ділянок (він буде мати вигляд двонаправленої стрілки) і транспортувати його у потрібне місце.

Вкладка **Размер бумаги** вікна **Параметры страницы** містить поле **Размер бумаги:**, у якому можна вибрати розмір сторінки документа. Якщо необхідні розміри у списку відсутні, то в полях **Ширина:** і **Высота:** можна ввести відповідні значення ширини і висоти сторінки.

Вкладка **Источник бумаги** вікна **Параметры страницы** дозволяє встановити параметри колонтитулів, зокрема, відстань від верхнього чи нижнього колонтитулів до межі сторінки.

Для того, щоб на сторінках із парними і непарними номерами були різні колонтитули, потрібно ввімкнути прапорець **четных и нечетных**



страниці. Щоб колонтитул першої сторінки відрізнявся від інших, необхідно ввімкнути прапорець **первой страницы**.

Спосіб вертикального вирівнювання тексту на сторінці вибирається в полі **Вертикальное выравнивание**. Тут можливі наступні варіанти: **По верхнему краю**, **По центру**, **По высоте** чи **По нижнему краю**.

Організація списків

Word надає широкі можливості щодо зручного оформлення найрізноманітніших списків. Зокрема, можна сформувати списки з **маркерами**, **нумерацією** чи **багаторівневі нумеровані** списки.

|| *Елементом списку вважається абзац тексту.*

Для створення списку необхідно виділити абзаци, які потрібно зробити елементами списку, або встановити курсор у той абзац, із якого буде починатися список. Після цього викликаємо команду меню **Формат / Список**, за якою виводиться вікно **Список** (рис. 4.15).

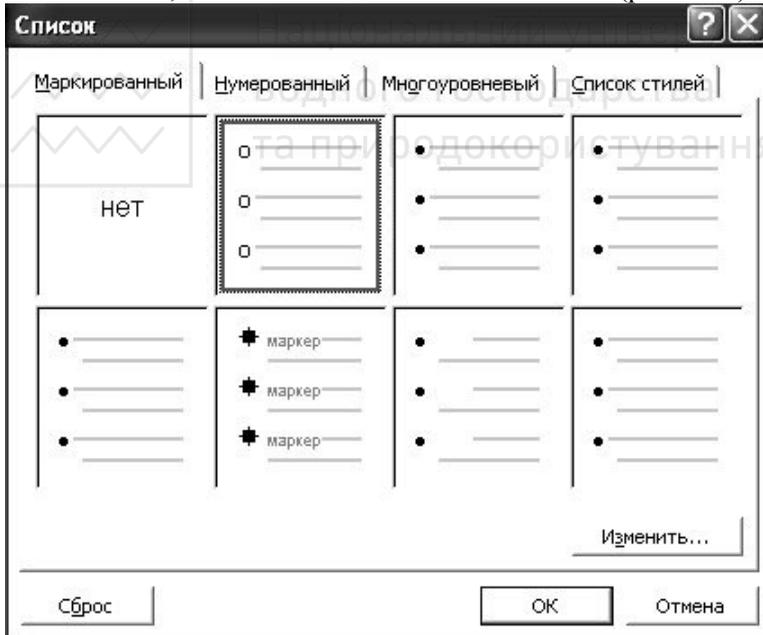


Рис. 4.15. Вкладка Маркированный вікна діалогу Список



Для створення списку з маркерами необхідно вибрати вкладку **Маркирований**. Кожен елемент списку позначається зліва невеликим знаком - маркером. Серед запропонованих варіантів маркерів можна вибрати потрібний, клацнувши на ньому ЛКМ і натиснувши кнопку **ОК**.

Для зміни виду маркерів використовують кнопку **Змінити....** З'явиться вікно **Изменение маркированного списка**, у якому містяться додаткові знаки. При клацанні кнопки **Знак...** відкриється діалогове вікно **Символ**, де можна вибрати будь-який із символів для його використання в якості маркерів списку. У рамці **Положение маркера** задається відстань від лівого краю абзацу до маркера. У рамці **Положение текста** визначається відстань від лівого краю абзацу до лівого краю тексту у списку.

Для створення нумерованих списків використовується вкладка **Нумерований** діалогового вікна **Список**. Серед запропонованих варіантів нумерації списку слід вибрати потрібний і клацнути **ОК**.

Коли курсор введення знаходиться у списку, кожне натискання **Enter** створює новий пронумерований елемент списку. При доповненні новим елементом списку або вилученні елемента, **номери в списку коригуються автоматично**.

Щоб створити свій варіант нумерації, потрібно клацнути кнопку **Змінити**, після чого з'явиться діалогове вікно (рис. 4.16).

У полі **Формат номера** вказується текст перед і після номера елемента списку, наприклад, дужка або крапка. Поле **нумерація**: заповнюється відповідним стилем, а у полі **начать с:** вказують літеру або число, з якого повинен починатися список. Кнопка **Шрифт...** використовується для зміни шрифту номерів елементів списку.

Марковані і нумеровані списки зручніше створювати за допомогою кнопок  **Нумерація** і  **Маркеры** панелі інструментів **Форматирование**.

Виведення тексту у кілька колонок

Word дозволяє розташовувати текст у кілька колонок. Текст вводиться в них послідовно, перехід до наступної колонки відбувається після заповнення попередньої. Для багатоконковий верстки потрібно перейти в режим **Разметка страниц**.

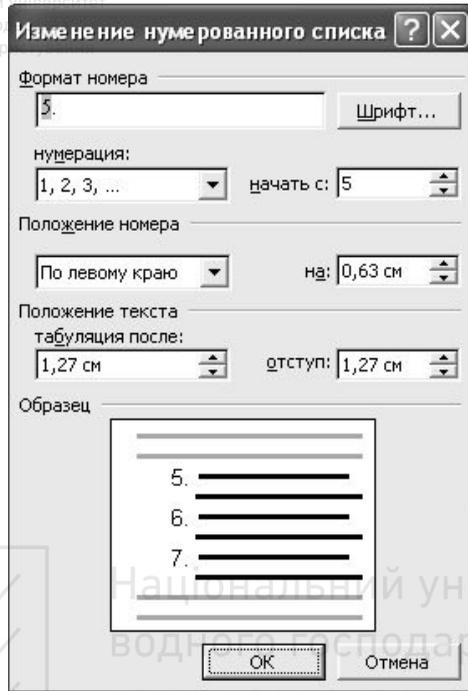


Рис. 4.16. Вікно діалогу Изменение нумерованного списка

Існують два варіанти використання багатоколонкової верстки.

- весь документ розбитий на певну кількість колонок однакової ширини;
- різні частини документа розбиті на різне число колонок або колонки мають різну ширину. У цьому випадку необхідно розбити документ на розділи, кожний із яких буде мати свій поділ на колонки.

Для створення колонок у межах розділу документа потрібно встановити курсор у тексті цього розділу. Якщо весь документ необхідно розбити на колоноки однакової ширини, то курсор може перебувати у будь-якому місці тексту. Потім слід вибрати команду меню **Формат / Колонки...**, що викликає діалогове вікно **Колонки** (рис. 4.17).

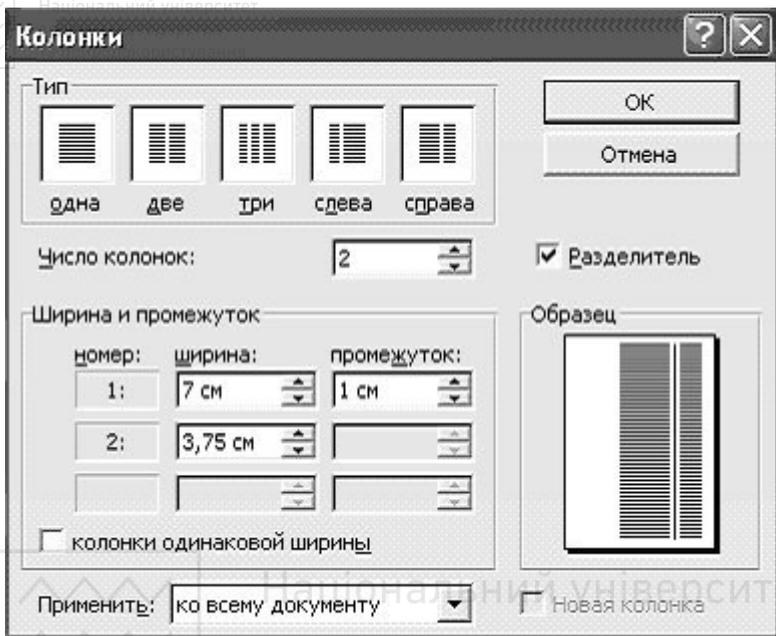


Рис. 4.17. Вікно діалогу Колонки

В полі **Число колонок:** необхідно ввести кількість колонок або вибрати один із малюнків у рамці **Тип**. Прапорець **Разделитель** встановлює лінію між стовпчиками тексту. Якщо включений прапорець **колонки однакової ширини**, то всі колонки будуть мати однакову ширину. Якщо цей прапорець виключити, для кожної колонки можна ввести точне значення її ширини, а також відстань між колонками в полях **ширина:** і **промежуток:**.

У полі **Применить:** вказується частина документа, до якої будуть застосовані обрані режими: **ко всему документу** чи **до кінця документа**.

Змінювати ширину колонок і відстань між ними зручно за допомогою горизонтальної лінійки, на якій відображаються границі колонок. Транспортування границь приводить до зміни параметрів колонок.

Для переходу до наступної колонки можна викликати команду меню **Вставка / Разрыв...**, у діалоговому вікні ввімкнути опцію **нову колонку** і клацнути **ОК**. Після цього курсор і весь текст нижче курсора переміститься на початок наступної колонки.



Знищення колонок виконують встановленням однієї колонки.

Встановлення позицій табуляції

Табуляція використовується для точного вирівнювання колонок тексту або чисел (рис. 4.18).

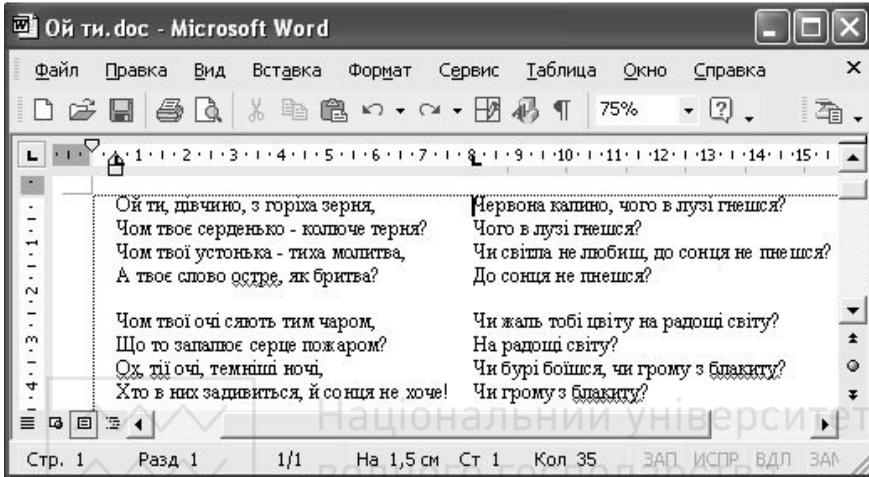


Рис. 4.18. Вирівнювання тексту до лівих країв позицій табуляції

Якщо встановити позиції табуляції, то при кожному натисканні клавіші **Tab** курсор буде переміщуватися до найближчої справа позиції табуляції.

Для встановлення позицій табуляції використовується команда меню **Формат / Табуляція**, яка викликає діалогове вікно **Табуляція** (рис. 4.19). Воно містить такі варіанти вирівнювання: **по лівому краю, по центру, по правому краю, по разделителю, с чертой**.

Для заповнення порожнього місця зліва від знаку табуляції можна використовувати ланцюжок символів, вигляд якого вибирається в групі перемикачів **Заполнитель**.

Встановивши всі необхідні параметри для однієї позиції, потрібно клацнути кнопку **Установить** і нова позиція буде занесена в список **Позиции табуляции**, що містить усі встановлені позиції табуляції.

Щоб змінити тип уже встановленої позиції табуляції, необхідно вибрати потрібний пункт у списку **Позиции табуляции** і встанови-

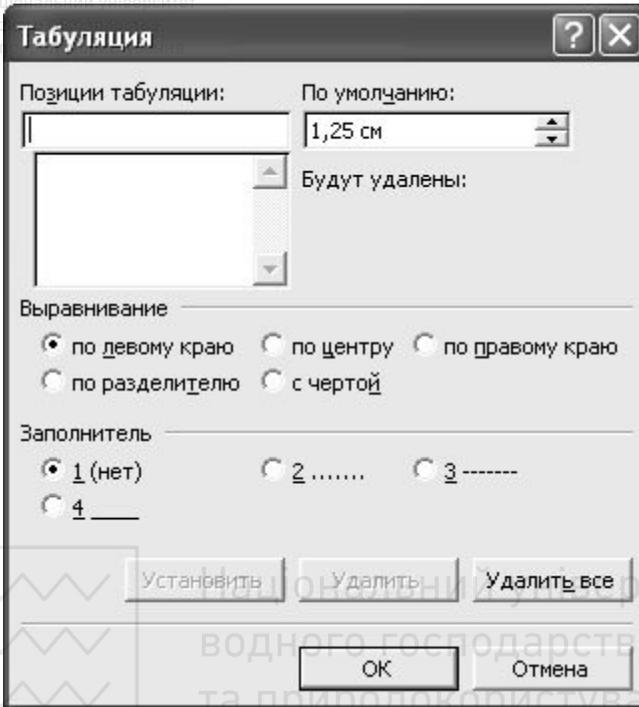


Рис. 4.19. Вікно діалогу Табуляція

ти нові значення режимів.

Для вилучення позиції табуляції досить вибрати її у списку **Позиції табуляції** і клацнути кнопку **Удалить**. Всі наявні позиції табуляції можна вилучити натисканням кнопки **Удалить все**.

Стили

Стиль форматування - набір параметрів (шрифту, абзацу тощо), що має унікальне ім'я.

Вибрати стиль виділеного фрагмента тексту можна кнопкою **текст + 11 pt, ▾** **Стиль** панелі інструментів **Форматирование** або в діалоговому вікні **Формат / Стили и форматирование...**. У полі **Выберите форматирование для применения** діалогового вікна **Стили и форматирование** (рис. 4.20) міститься список доступних стилів. Щоб побачити всі стилі, потрібно у полі **Показать:** вибрати **Все**. Для присвоєння виділеному фрагменту тексту вибраного стилю



потрібно клацнути на цьому стилі.

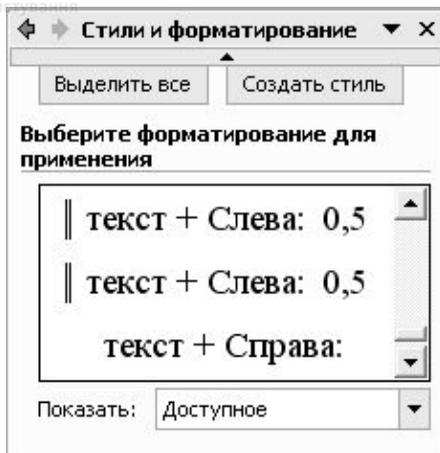


Рис. 4.20. Вікно Стили и форматирование

Створення нового стилю у вікні **Стили и форматирование** можна здійснити кнопкою **Создать стиль**, після клацання якої з'явиться діалогове вікно **Создание стиля** (рис. 4.21). У полі **Имя**: вводиться ім'я нового стилю. У списку **Стиль**: вибирається вид стилю: **Абзаца**, **Знака** чи **Таблицы**. У полі **Основан на стиле**: вибирається існуючий стиль, на базі якого буде створено новий. У полі **Стиль следующего абзаца**: вказують стиль для наступного абзацу.

Якщо встановити прапорець **Добавить в шаблон**, новий стиль буде діяти не тільки в активному вікні, але і у всіх документах, створених на основі цього шаблону.

Для встановлення параметрів шрифту чи абзацу потрібно клацнути кнопку **Формат** і вибрати об'єкт форматування **Шрифт**, **Абзац** тощо. Після натискання кнопки **ОК** новий стиль буде створений.

Щоб змінити існуючий стиль, потрібно навести вказівник миші на його ім'я у вікні **Стили и форматирование** та клацнути кнопку в кінці рядка, що має вигляд направленою вниз трикутника. У розгорнутому списку вибрати команду **Изменить стиль...**. У вікні **Изменение стиля** можна вибрати нові параметри. Для видалення стилю його потрібно виділити і клацнути команду **Удалить...**

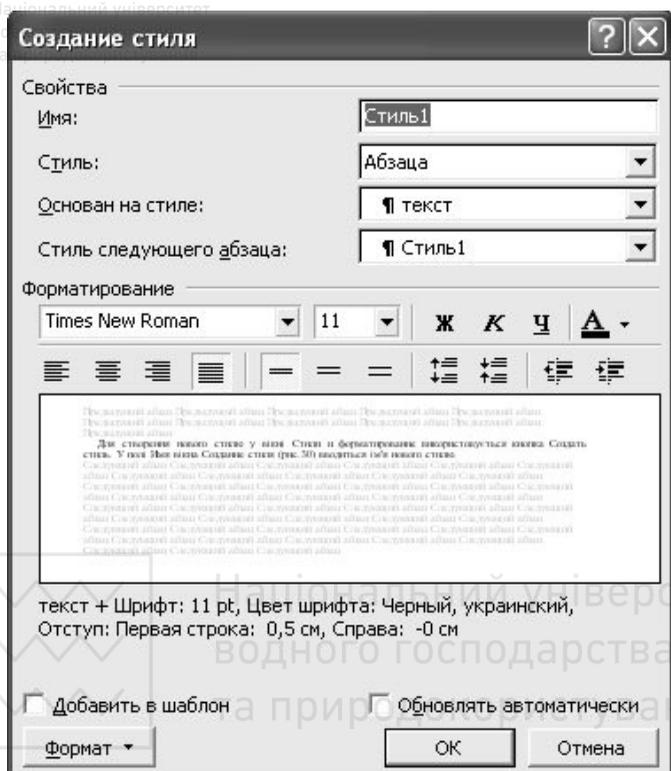


Рис. 4.21. Вікно Создание стиля

Створити стиль можна також за зразком. З цією метою необхідно виділити фрагмент тексту, взятого як зразок, ввести ім'я стилю в полі **Стиль** на панелі **Форматирование** і натиснути **Enter**. Створений стиль буде діяти тільки в активному документі.

Вставка розривів сторінок

Word автоматично розбиває текст на сторінки. Для вставки додаткового розриву сторінки необхідно встановити курсор у рядок, з якого повинна починатися нова сторінка, і викликати команду меню **Вставка / Разрыв...**

У діалоговому вікні **Разрыв** (рис. 4.22) необхідно встановити перемикач **новую страницу** і клацнути **OK**.

Якщо документ складається із сторінок, що мають різні параметри, його слід розділити на декілька розділів. Кожен розділ

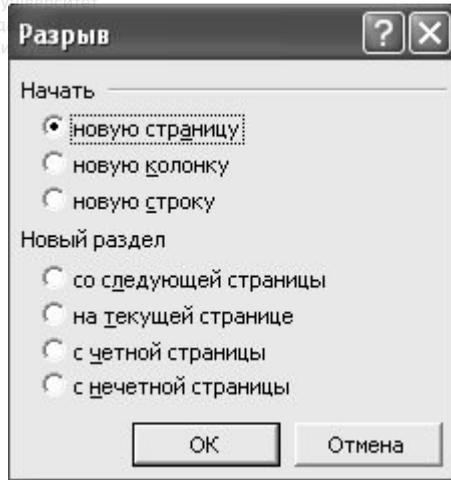


Рис. 4.22. Вікно діалогу Разрыв

має власні параметри сторінки. Для вставки в документ нового розділу, в діалоговому вікні **Разрыв** необхідно вибрати один із перемикачів: **со следующей страницы**, **на текущей странице**, **с четной страницы** чи **с нечетной страницы**.

Щоб вилучити розрив розділу або розрив сторінки, встановлений власноручно, потрібно перейти в режим **Обычный** або ввімкнути режим відображення невидимих символів. У цих режимах розриви сторінок позначені пунктирними лініями, а розриви розділів - подвійними пунктирними лініями. Вилучаються знаки розривів як звичайні символи клавішами **Delete** або **Backspace**.

Нумерація сторінок

Для вставки номерів сторінок необхідно виконати команду меню **Вставка / Номера страниц...**, яка викликає вікно **Номера страниц** (рис. 4.23). У полі **Положение:** можна вибрати розташування номера вгору чи вниз сторінки. У полі **Выравнивание:** - розміщення номера сторінки щодо полів сторінки: **Слева**, **От центра**, **Справа**, **Внутри** чи **Снаружи**. Останні два варіанти доступні, якщо документ має дзеркальні поля. Якщо не встановлений прапорець **Номер на первой странице**, на першій сторінці номера не буде. Кнопка **Формат...** викликає діалогове вікно **Формат номера страницы** (рис. 4.24), у якому задається формат нумерації.

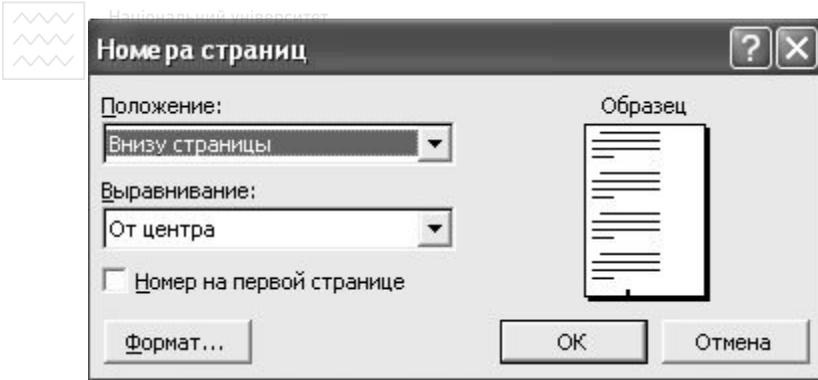


Рис. 4.23. Вікно Номера страниц

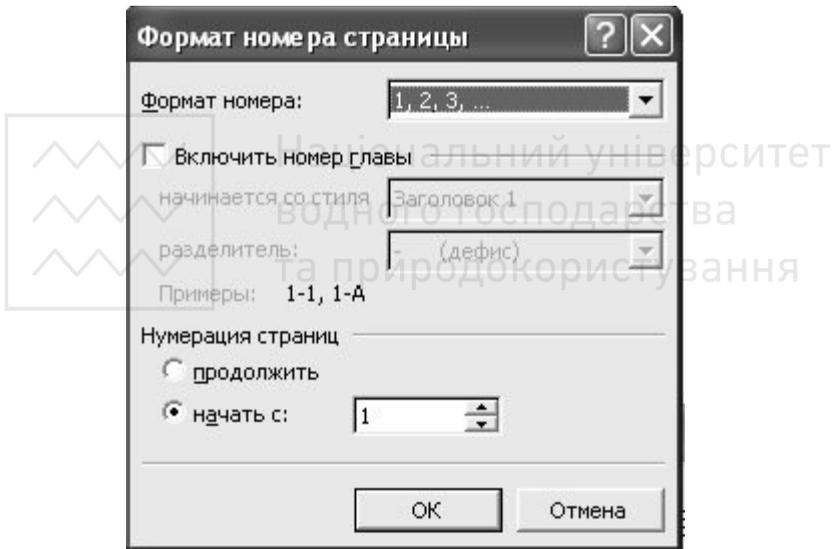


Рис. 4.24. Вікно Формат номера страницы

У полі **Формат номера** вибирається тип нумерації: **арабські** або **римські цифри** чи **літери латинського алфавіту**. У рамці **Нумерація страниц** встановлюється початок нумерації: **продолжить** чи **начать с:**. У випадку вибору останнього варіанту додатково вказують номер початкової сторінки.

Якщо ввімкнути прапорець **Включить номер главы**, до номера сторінки буде доданий номер розділу документа. У полі **начинается**

со стиля необхідно зазначити, який стиль форматування відповідає рівню розділів, номери яких будуть використані. Можна вибрати один із стилів заголовків **Заголовок1... Заголовок9**. У полі **разделитель**: задається роздільник між номером сторінки і номером розділу. Установивши всі параметри, потрібно клацнути **ОК**, після чого відбудеться повернення до вікна **Номера страниц**. Для завершення процесу знову потрібно клацнути **ОК**.

Встановлення колонтитулів

Колонтитул - інформація, що друкується на кожній сторінці документа.

Колонтитули можуть містити номери сторінок, назву книги або розділу. В залежності від розташування, колонтитули поділяють на **верхні** та **нижні**. Текст колонтитула форматується як звичайний текст.

Для створення колонтитулів потрібно вибрати команду меню **Вид / Колонтитулы**. При цьому автоматично здійснюється перехід у режим **Разметка страниц**, оскільки в режимі **Обычный** колонтитули не відображаються. На екрані з'являється поле для введення тексту колонтитула та панель інструментів **Колонтитулы** (рис. 4.25).

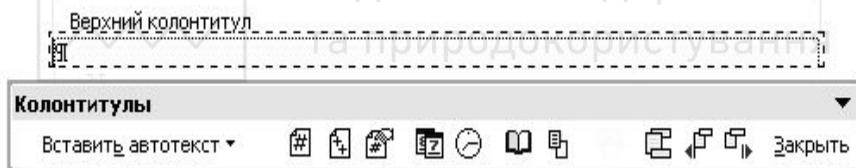


Рис. 4.25. Поле для введення тексту та панель інструментів Колонтитулы

Введений текст розташовується в пунктирній рамці, що вказує межі колонтитула. У режимі відображення колонтитулів основний текст документа редагувати неможливо, тому для переходу до редагування основного тексту потрібно клацнути кнопку **Закреть** на панелі **Колонтитулы**.

Вставка номерів сторінок у колонтитули здійснюється кнопкою

 **Номер страницы**

Для створення на першій сторінці документа колонтитула, відмінного від колонтитулів інших сторінок, необхідно виконати команду меню **Файл / Параметры страницы** або клацнути кнопку

 **Параметры страницы** панелі **Колонтитулы** і на вкладці

Источник бумаги встановити прапорець **первой** сторони. Якщо на цій вкладці встановити прапорець **четных и нечетных страниц**, то можна створити окремі колонтитули для парних і непарних сторінок.

Для переходу з поля верхнього у поле нижнього колонтитула і навпаки використовується кнопка  **Верний/нижний колонтитул**.
Переміщення між колонтитулами парних та непарних сторінок здійснюється кнопкою  **Переход к предыдущему** чи  **Переход к следующему**.

Встановлення прапорця **четных и нечетных страниц** впливає на весь документ, якщо він не розбитий на розділи. В документі із декількох розділів при вставці колонтитула в один розділ цей же колонтитул автоматично додається у всі розділи документа, якщо після набору тексту колонтитула натиснути кнопку  **Как в предыдущем**.

Щоб створити різні колонтитули для декількох частин документа, потрібно вставити в текст розриви між розділами через меню **Вставка / Разрыв...**. Після цього необхідно переходити по черзі в наступні розділи і змінювати існуючі колонтитули або створювати нові. Для знищення колонтитула потрібно вибрати команду меню **Вид / Колонтитулы**, виділити колонтитул, який необхідно вилучити, і клацнути клавішу **Delete**.

Вставка графічних об'єктів

Word дозволяє вставляти в документ графічні об'єкти, створені як в інших програмах, так і за допомогою власної панелі малювання. Об'єкти можна копіювати і вставляти у будь-яке місце документа. Доданий малюнок приєднується до навколишнього тексту документа. Якщо абзац, що містить малюнок, посувається вздовж сторінки, малюнок пересувається разом із ним.

Вставити новий малюнок у текст можна командою меню **Вставка / Рисунок / Создать рисунок** або клацнувши кнопку  панелі інструментів **Стандартная**. При цьому доцільно перейти в режим **Разметка страниц**. За допомогою кнопок панелі інструментів **Рисование** можна зобразити лінії, стрілки, прямокутники та інші геометричні фігури. Графічний об'єкт можна залити кольором чи візерунком, змінити його форму, дзеркально відобразити або



повернути. В ньому можна змінити колір і тип ліній, додати до них стрілки.

Для вставки графічного об'єкта, створеного в іншій програмі, необхідно встановити курсор у позицію, де планується розмістити об'єкт, і вибрати команду меню **Вставка / Рисунок, Из файла**. У вікні **Добавление рисунка** (рис. 4.26) у полі **Папка:** слід вказати

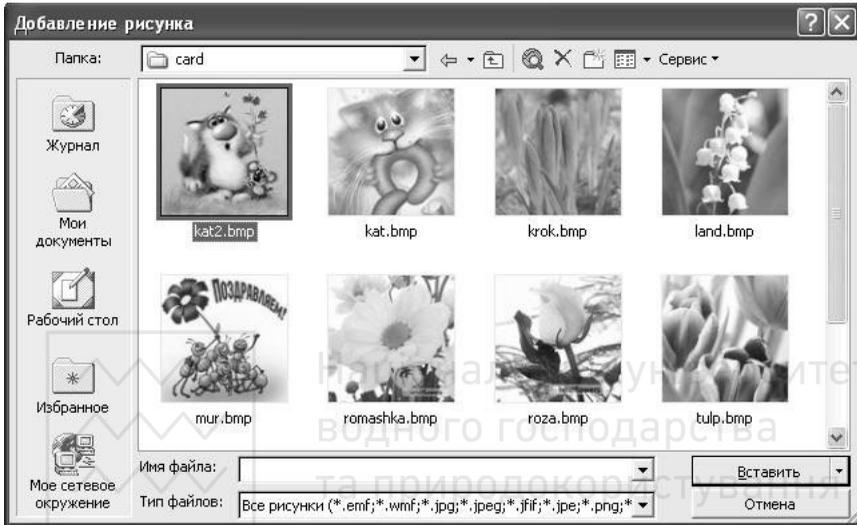


Рис. 4.26. Вікно Добавление рисунка

Диск та розгорнути папку, що містить файл із малюнком, виділити мишею файл і клацнути кнопку **Вставити**. Для вставки малюнків, що поставляються з редактором Word, потрібно після команди **Рисунок** вибрати варіант **Картинки**.

Щоб змінити розміри малюнка, потрібно клацнути на ньому ЛКМ. Після цього навколо об'єкта з'являться маркери. Транспортуючи їх мишею, виконують необхідні зміни розмірів. Перемістити малюнок можна також транспортуванням мишею.

За допомогою контекстного меню малюнка можна ввімкнути панель **Настройка изображения** з кнопками для зміни таких його характеристик, як яскравість, контрастність та ін.

Змінити інші параметри малюнка, що стосуються його розміру, розташування відносно тексту тощо, можна у вікні **Формат рисунка**. Для цього необхідно вибрати відповідну команду меню **Формат /**



Рисунок... або клацнути кнопку  панелі **Настройка изображения**.

Наприклад, щоб текст міг розташовуватися зліва або справа від малюнка, потрібно вибрати вкладку **Положение** і там вказати варіант **вокруг рамки**.

Для знищення малюнка його виділяють і натискають клавішу **Delete**.

Формування таблиць

Для оформлення тексту у вигляді таблиці необхідно встановити курсор у місці, де повинна починатися таблиця, і вибрати команду меню **Таблица / Вставить / Таблица**. Далі у діалоговому вікні потрібно вказати кількість стовпців та рядків таблиці і клацнути **ОК**.

Для вставки таблиці також можна клацнути кнопку  панелі інструментів **Таблицы и границы**. Існує також варіант малювання таблиці вручну, при цьому застосовують інструменти  - для нанесення ліній та  - для їх витирання. Всі команди для роботи з таблицями розміщені в меню **Таблица**.

Кожна клітина таблиці розглядається як абзац, тому і дані в клітинах формуються як абзаци тексту.

При наведенні вказівника миші на верхню лінію таблиці, він перетворюється у чорну стрілку. Якщо в цей момент клацнути ЛКМ, то виділиться один стовпець. Транспортуючи з допомогою миші чорну стрілку, можна виділити відразу кілька стовпців. Рядки таблиці виділяються як рядки звичайного тексту. Для виділення кількох суміжних клітинок необхідно натиснути ЛКМ в одній клітині фрагмента і розтягнути виділення на інші.

Для об'єднання кількох клітин в одну можна виділити їх і викликати команду меню **Таблица / Объединить ячейки** або

клацнути кнопку  на панелі **Таблицы и границы**. Розбити клітину чи блок клітин на вказану кількість колонок можна таким чином: виділити клітину чи блок клітин і вибрати команду меню **Таблица /**

Разбить ячейки або клацнути кнопку .

Коли текстовий курсор розміщений в таблиці, на координатних лінійках з'являються знаки границь стовпців  і рядків . При транспортуванні цих знаків мишею змінюються розміри відповідних



стовпців і рядків. За допомогою кнопки



можна вибрати тип

вирівнювання тексту в клітинах. Кнопка



служить для зміни напрямку тексту у виділених клітинах.

Для вставки елементів таблиці (рядків, стовпців, клітин) необхідно виділити елементи, на місці яких необхідно вставити нові. Вибравши команду меню **Таблиця / Вставити**, вказуємо необхідний варіант: **Столбцы слева, Столбцы справа, Строки выше, Строки ниже** чи **Ячейки**. Щоб вилучити елемент таблиці, треба виділити його і в меню вибрати команду **Таблиця / Удалить**, після чого вказати необхідну опцію: **Таблиця, Столбцы, Строки** чи **Ячейки**.

За замовчуванням лінії сітки таблиці мають товщину 0,5 пункту.

Змінити оформлення таблиці можна таким чином:

- виділити клітини, оформлення яких потрібно змінити;

- у полі  **Тип лінії** вибирати тип лінії, у полі



Толщина лінії – товщину лінії. Кнопка  **Цвет границы** висвічує палітру кольорів, де можна призначити колір оформлення. Вид оформлення вказується за допомогою списку



а для заливки клітин кольором використовується список

Введення формул

Конструювання складних формул у редакторі Word виконують за допомогою спеціальної програми Microsoft Equation, яку запускають командою меню **Вставка / Объект... / Microsoft Equation 3.0** або за

допомогою кнопки  **Редактор формул** на панелі інструментів. Після цього відкриється додаткова панель елементів **Формула**, яка надає доступ до наборів окремих знаків та шаблонів математичних символів (рис. 4.27).

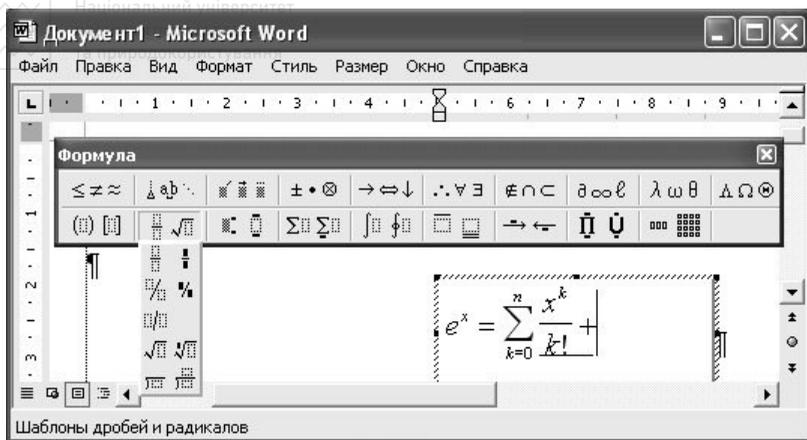


Рис. 4.27. Панель елементів Формула з доступом до шаблонів дробів та радикалів

Формулу конструюють методом введення звичайних символів з клавіатури та вставки спеціальних символів або шаблонів з панелі елементів. Вибрані з панелі шаблони містять позиції-слоти, позначені чорними маркерами, які призначені для введення конкретних значень.

Переміщення між слотами здійснюють за допомогою клавіш із стрілками або клавіші **Tab**. Команди **Формат**, **Стиль**, **Размер** головного меню програми Microsoft Equation дають змогу відрегулювати всі параметри елементів формули так, щоб вони набули потрібного вигляду.

Друкування документів

Перед друкуванням документа доцільно переглянути, як він буде виглядати на папері. Для цього необхідно перейти в режим попереднього перегляду за допомогою команди меню **Файл / Предварительный просмотр** або кнопки . Щоб збільшити зображення, потрібно навести вказівник миші (він буде мати вигляд лупи з плюсом) на потрібний фрагмент і натиснути ЛКМ. При цьому вказівник миші перетвориться в лупу із знаком мінус. Вийти із режиму попереднього перегляду можна за допомогою кнопки **Закрить** або клавіші **Esc**.

Для друкування документа використовується команда меню **Файл / Печать** або комбінація клавіш **Ctrl+P**. В діалоговому вікні **Печать**

(рис. 4.28) в полі **ім'я**: потрібно вибрати принтер, якщо є можливість друкувати на різних принтерах.

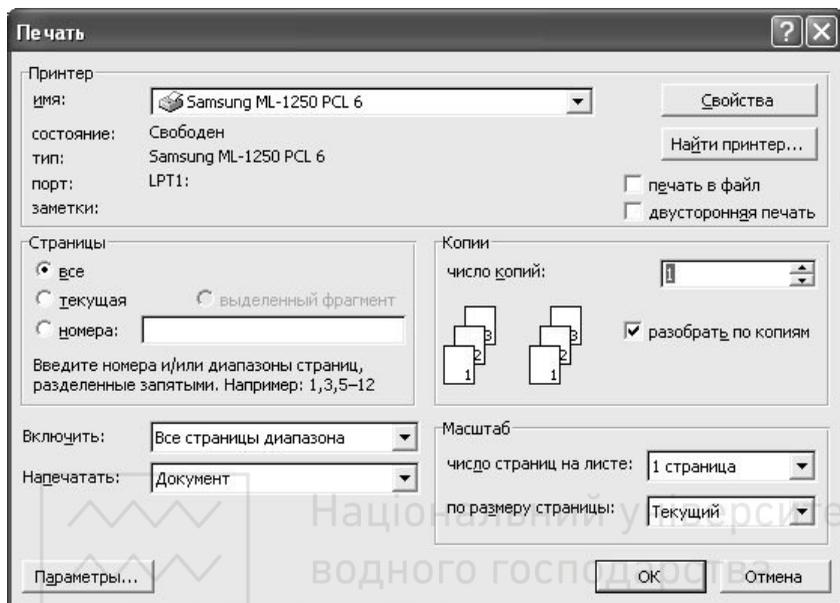


Рис. 4.28. Вікно діалогу Печать

у рамці **Страницы** задається діапазон сторінок, які будуть надруковані: **все**, **текущая**, **выделенный фрагмент** чи **номера**.

Наприклад, щоб надрукувати сторінки 1, 2, 3, 7, 9, необхідно ввести в поле **номера**: 1-3, 7, 9.

У рамці **Копии** вказується кількість копій. Щоб роздрукувати цілком першу копію, потім другу і т.д., потрібно ввімкнути прапорець **разобрать по копиям**.

Для друкування багатосторінкового документа з двох сторін кожного аркуша, можна ввімкнути режим виведення на друк тільки парних або непарних сторінок. У списку **Включить**: можна вибрати одне із значень: **Все страницы диапазона**, **Нечетные страницы** чи **Четные страницы**.

Для друкування однієї копії всього документа досить натиснути кнопку  **Печать**.



Контрольні запитання і завдання

1. Як створити новий документ, відкрити існуючий чи зберегти набраний текст?
2. Яку стандартну назву і тип має файл документа Word?
3. Назвіть панелі інструментів та найбільш широко вживані інструменти редактора Word.
4. Назвіть режими відображення документа на екрані.
5. Перелічіть параметри сторінки документа.
6. Які режими використовують для введення тексту?
7. Як утворити новий абзац у тексті? Опишіть процедуру об'єднання двох абзаців.
8. Поясніть призначення лінійки. Які елементи можуть бути розташовані на горизонтальній лінійці?
9. Що таке форматування тексту?
10. Як можна виділити: весь текст, абзац, речення, рядок, слово?
11. Які види списків існують?
12. Що таке стиль?
13. Поясніть використання буфера обміну.
14. Опишіть послідовність дій при копіюванні чи перенесенні виділеного фрагмента тексту.
15. Як виконують скасування операцій чи їх повторення?
16. Назвіть способи створення таблиць у документі Word.
17. Як об'єднують клітини таблиці в одну?
18. Як забрати окремі лінії в таблиці чи змінити їх вигляд?
19. Як доповнити текст формулою чи малюнком?
20. Які дії можна виконувати із вбудованими об'єктами?
21. Як змінити розміри об'єкта?
22. Як вибрати декілька об'єктів?
23. Для чого виконують команду **Групувати**?
24. Як обгорнути об'єкт текстом?



4.3. MS Excel

Призначення та можливості електронних таблиць

Електронні таблиці (ЕТ) – спеціальний вид прикладного програмного забезпечення. Їх часто називають **табличними процесорами**.

Електронні таблиці використовуються для виконання обчислень, а також подання даних у формі таблиць та їх аналізу з використанням графічних засобів.

Особливістю роботи з ЕТ є те, що вся інформація розміщується в клітинах таблиці. Для опису зв'язку між різними клітинами використовуються формули. Зміна вмісту однієї клітини викликає перерахунок значень усіх клітин, які зв'язані з нею формулами. Розрахунки можуть відбуватися автоматично (режим роботи за замовчуванням) або за командою користувача.

Використання ЕТ дозволяє розв'язувати широкий клас задач інженерної практики без рутинного виконання ручних розрахунків або застосування мов програмування. В операційній системі Windows найчастіше використовуються ЕТ Excel, які включені в пакет програм MS Office.

Порядок розв'язування задачі в ЕТ:

- 1) ввести в клітини таблиці початкову інформацію у вигляді чисел, дат, тексту та ін.;
- 2) записати формули у потрібні клітини, використовуючи стандартні функції та знаки операцій;
- 3) на основі початкових даних і отриманих розрахунків побудувати діаграми та провести аналіз результатів;
- 4) оформити таблицю та роздрукувати результати.

Проектувати електронну таблицю необхідно із врахуванням масовості складеного алгоритму, тобто таким чином, щоб для однотипних задач можна було отримувати результати тільки шляхом заміни початкових даних, не змінюючи структури таблиці.

Можливості сучасних ЕТ базуються на принципах, які відповідають стандартам роботи з графічними оболонками та особливостям прикладного програмного забезпечення такого виду. Основні з них: зручний користувацький інтерфейс; багатовіконний режим роботи; імпорт, експорт даних таблиці у файли інших форматів

Национальный университет
на приклад, баз даних); зручні засоби для створення діаграм за даними таблиці; створення макросів для програмування дій, які часто повторюються; засоби для розв'язку багатьох типів задач з використанням стандартних функцій та пакетів; друк документів; можливе вбудовування (вставка) різноманітних об'єктів та ін.

Вікно Excel

Розглянемо русифіковану версію ET Excel 2003. Для запуску програми використовується ярлик . Вікно Excel показано на рис. 4.29. Крім стандартних елементів, які характерні для інших програм Windows, воно містить власні інструменти. Відмінність від інших програм при роботі з документом полягає в тому, що існують наступні режими роботи: введення інформації в клітину (наявність слова **Ввод** у нижньому лівому куті вікна) або готовність до виконання команд (слово **Готово**).

|| Документ ET Excel називається *робочою книгою*.

Кожна книга зберігається в окремому файлі на диску. Файл із робочою книгою Excel має розширення **.xls**. Робоча книга складається з аркушів (листоків). Будь-який аркуш має назву, яка міститься на його ярлику. Кожна робоча книга має смугу прокрутки аркушів, розташовану внизу вікна. Нова робоча книга за замовчуванням містить найчастіше три аркуші. У будь-який момент часу аркуш робочої книги може бути вилучений, перейменований, переміщений, скопійований. В робочу книгу можна вставляти нові аркуші. Всі дії з аркушами виконуються командами контекстного меню аркуша або за допомогою пунктів меню **Правка** чи **Вставка**. На кожному аркуші можна розмістити таблиці, діаграми, слайди, макрос та інше. Активізація аркуша (перехід з аркуша на аркуш) відбувається клацанням ЛКМ на відповідному ярлику. Назва активного аркуша виділена напівжирним шрифтом, а сам ярлик має білий фон.

Одночасно може бути відкрито декілька робочих книг. Кожну книгу можна згорнути до значка вікна у вигляді смужки із назвою файлу. Дані з різних робочих книг можуть бути зв'язані формулами.



Призначення деяких інструментів програми Excel наведено в таблицях 4.4 та 4.5.

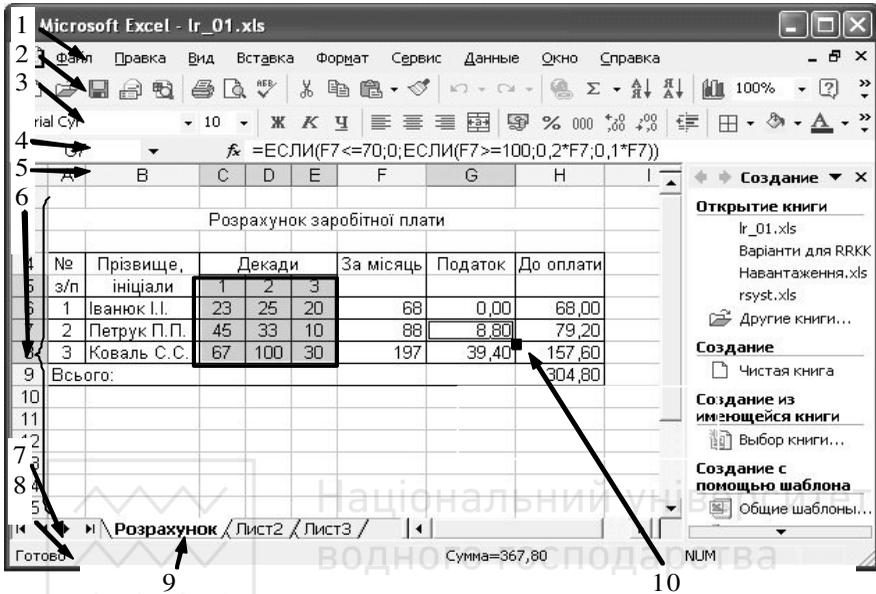


Рис. 4.29. Вікно програми Excel з відкритою робочою книгою Ir_01

- 1) рядок меню;
- 2) панель інструментів Стандартная;
- 3) панель інструментів Форматирование;
- 4) рядок формул (зліва направо: ім'я блоку чи адреса активної клітини, кнопка розкриття списку імен клітин, кнопки внесення або скасування змін у формули (у даному режимі відсутні), кнопка виклику майстра функцій, вміст активної клітини);
- 5) смуга назв стовпців активного листка;
- 6) стовпець номерів рядків активного листка;
- 7) чотири кнопки прокрутки листків;
- 8) рядок стану;
- 9) назва активного листка книги (ярлик);
- 10) маркер автозаповнення клітин.



Деякі інструменти панелі Стандартная

Инструмент	Назва	Призначення
	Формат по образцу	Копіювання формату
	Автосумма	Обчислення суми
	Вставка функции	Вставка функції
	Сортировка по возрастанию	Сортування у зростаючому порядку
	Мастер диаграмм	Виклик майстра для побудови діаграми
	Масштаб	Зміна масштабу

Таблица 4.5.

Деякі інструменти панелі Форматирование

Инструмент	Назва	Призначення
	Объединить и поместить в центре	Об'єднання клітин з вирівнюванням вмісту
	Денежный формат	Грошовий формат
	Процентный формат	Відсотковий формат
	Увеличить разрядность	Збільшити розрядність
	Уменьшить разрядность	Зменшити розрядність
	Уменьшить отступ	Зменшити відступ
	Увеличить отступ	Збільшити відступ
	Границы	Встановити межі



Як і у текстовому редакторі Word, в Excel можна вмикати готові панелі інструментів або самостійно їх конструювати, використовуючи команду меню **Сервіс / Настойка...** Для вмикання існуючої панелі на вкладці **Панели инструментов** потрібно поставити відмітку навпроти назви відповідної панелі. Нову панель створюють клацанням на кнопці **Создать...** Для заповнення створеної панелі інструментами треба клацнути на вкладці **Команды**, вибрати зліва категорію команд, а справа знайти позначення потрібної команди (інструмент) та перетягнути його на панель. У вікні **Настройка** на вкладці **Параметры** вказується, чи потрібно виводити підказки для інструментів та показувати їх збільшені значки.

Основні визначення

Кожен аркуш є окремою таблицею, яка складаються із стовпців та рядків. Рядки позначаються арабськими числами від 1 до 65536. В таблиці використовується 256 стовпців, які позначаються латинськими літерами від A до Z чи їх сполученнями від AA до IV. На рис. 4.29 показано розбиття аркуша книги на стовпці та рядки лініями сітки, яка полегшує роботу з обробки інформації і, як правило, не друкується. Основне поняття ET, як і будь-якої іншої таблиці, – клітина.

Клітина – елемент електронної таблиці, що розміщений на перетині стовпця і рядка, має свою адресу і призначений для зберігання даних.

Адреса клітини складається з назви стовпця та номера рядка.

Наприклад: **B29** - стовець **B**, рядок **29**; **DS108** - стовець **DS**, рядок **108**. При наборі з клавіатури можна використовувати і малі літери, наприклад, **b29**.

Адреси клітин використовують у формулах і при виконанні команд. Наведена вище адресація називається **відносною**. При копіюванні формул відносні адреси клітин у формулі змінюються на величину, що залежить від "відстані" копіювання.

Наприклад, у клітині **B2** міститься формула **=A2*10**. При копіюванні вмісту клітини **B2** у клітину **B4** отримаємо в клітині **B4** формулу **=A4*10**. Тобто, при копіюванні формули на два рядки вниз всі відносні адреси клітин у формулі будуть автоматично замінені на ті, що відповідають двом рядкам нижче.



Інший вид адресації клітин у формулах – **абсолютний**. У випадку абсолютної адресації клітин перед назвою стовпця та номером рядка записуються символи грошової одиниці **\$**. Це вказує на те, що відповідні рядок та стовпець цієї адреси при копіюванні формули змінюватися не будуть.

Наприклад, **\$B\$12** - абсолютна адреса клітини **B12**.

Змішані адреси містять тільки один значок **\$**. При копіюванні змішаної адреси змінюватися може тільки та частина адреси, що “не зафіксована” знаком **\$**.

Наприклад, при копіюванні формули, що містить адресу **\$B12**, стовпець залишатиметься незмінним, а номер рядка може стати іншим. При копіюванні формули, що містить адресу **B\$12**, стовпець може змінюватися, а номер рядка залишатиметься без змін.

На аркуші завжди є **активний елемент**: клітина, блок, діаграма тощо. Активна клітина виділяється **селектором (табличним курсором)**, який має вигляд прямокутної рамки. Змінити активну клітину можна переміщенням селектора клавішами управління курсором або за допомогою клацання ЛКМ. Вміст активної клітини та її адреса завжди відображені у рядку формул (рис. 4.29, позиція 4). На робочому аркуші у клітині, як правило, відображено результат виконання формули.

|| *Прямокутну область клітин називають **блоком**.*

Поряд із словом “блок” використовують і інші назви: фрагмент, діапазон. Блок визначається адресами клітин, розміщених у протилежних кутах прямокутника. Блок у формулах чи командах вказують адресами цих клітин, розділеними двокрапкою.

Наприклад, блок клітин, відмічений на рис. 4.29 контрастним кольором, визначається адресами клітин **C5:E8** (або **E5:C8**).

Клітинам та блокам клітин можна присвоювати **імена**. Це дозволяє швидко переходити до необхідних блоків, використовувати їх імена у формулах як адреси чи посилання. Для присвоєння блоку клітин імені можна виділити потрібний блок, виконати команду меню **Вставка / Імя / Присвоїть...**, вказати ім'я та завершити процес клацанням кнопки **ОК**.

Поряд із розглянутою традиційною адресацією клітин, іноді використовують стиль **R1C1** (англ. Row – рядок, Column - стовпець), в якому і рядки, і стовпці нумеруються. Цей стиль вмикається на вкладці **Общие** команди меню **Сервис / Параметри**. При



застосуванні стилю R1C1 абсолютна адреса, наприклад, клітини \$D\$2 буде записана як R2C4 (перетин другого рядка і четвертого стовпця).

Типи даних та операції

В Excel використовуються наступні типи даних: числа, дати, час, текст, адреси клітин та формули. Excel визначає тип даних клітини за вмістом інформації, яка введена в цю клітину. **Числа** записуються у звичайному або експонентному форматі. Наприклад, 234; 23,21; 34E-12. **Дати** записуються з допомогою цифр і крапки, яка розділяє день місяця, місяць, рік. Наприклад, 22.12.99. **Час** записується цифрами і двокрапкою, що розділяє години, хвилини, секунди. Вказані типи даних автоматично вирівнюються до правого краю клітини.

|| **Текст** - будь-яка послідовність символів.

Вміст клітини автоматично розпізнається як текст у тому випадку, якщо він не визначений в якості іншого типу. Для запису інших даних у якості тексту перед ними ставиться апостроф. Текст автоматично вирівнюється до лівого краю клітини.

|| **Формула** - допустимий вираз, який містить дані та елементи електронної таблиці.

Будь-яка формула, як правило, починається із символу = (дорівнює). При наборі формули її можна розпочинати і із іншого математичного знаку, наприклад, + (плюс). Формула складається з даних, вбудованих функцій, які розділені знаками операцій. Правила запису виразів практично не відрізняються від правил їх запису в математиці.

У формулах використовуються наступні основні операції:

^	піднесення до степеня
* /	множення і ділення
- +	віднімання і додавання
= <> < > <= >=	операції порівняння
&	об'єднання рядків (конкатенація)

Операції перелічені в порядку спадання їх пріоритетів. Множення має вищий пріоритет, ніж додавання, тому: $5 + 1 * 3 = 8$, а не 18. Операції з однаковим пріоритетом виконуються при обчисленні

формули зліва направо. Для зміни порядку виконання операцій використовують круглі дужки.

Дані у формулах подаються: **числами, датами, часом, текстовими константами, адресами клітин, іменами**. Приклади формул: $=134-34^3$; $=3*A2^3-10$; $=СУММ(A2:F8)$; $=(A6/(B6-2))+SIN(A6)+СРЗНАЧ(A2:A8)$; $=СЕГОДНЯ()-ДАТА(2000;1;1)$.

Використання у формулах адрес клітин з даними замість конкретних даних дає змогу гарантувати цілісність та правильність виконання обчислень, виключає необхідність редагування кожної формули при зміні даних.

Формули можуть містити адреси клітин з іншого аркуша або з іншої робочої книги. Наприклад: Лист2!B8 – клітина B8 з аркуша Лист2 (назва аркуша відділяється від адреси клітини символом !); [Книга2.xls]Лист2!B8 (назва книги записується в квадратних дужках).

Якщо введена формула містить помилки, то вона не обчислюється. При цьому в клітині з'являється повідомлення про помилку. Перелік таких повідомлень розміщено в таблиці 4.6.

Таблиця 4.6.

Повідомлення про помилку

Помилка	Пояснення
#####	Стовпець має ширину меншу, ніж потрібно для відображення даного формату даних або дата чи час мають від'ємне значення
#ЗНАЧ!	Використано недопустимий тип аргументу або операції
#ДЕЛ/0!	Спроба ділення на нуль
#ИМЯ?	Не розпізнано ім'я (клітини, блоку тощо), що використовується у формулі
#Н/Д	Значення недоступне у формулі чи функції (клітина порожня, неправильні значення аргументів тощо)



#ССЫЛКА!	Посилання на клітину вказано неправильно
#ЧИСЛО!	Неправильні числові значення у формулах або функціях
#ПУСТО!	Вказано перетин двох областей, які не мають спільних клітин

Для пошуку помилок Excel містить допоміжну функцію, з допомогою якої можна графічно відобразити зв'язки між клітинами у формулі. Найзручніше відслідковувати залежності з допомогою панелі інструментів **Зависимости (Сервис / Зависимости формул / Панель зависимостей)**.

Основні функції

Функції виконують спеціальні обчислення. Опис функцій можна переглянути у предметному вказівнику довідки в алфавітному порядку або за списками груп у кожній з основних **категорій**. Основні категорії функцій: **Инженерные, Финансовые, Дата и время, Математические, Статистические, Ссылки и массивы, Работа с базой данных, Текстовые, Логические, Проверка свойств и значений, Информационные**.

Запис функцій у формулах відбувається за наступними **правилами**:

- назва функції записується літерами нижнього або верхнього регістрів;
- всі літери у назві функції повинні бути з однієї мови (англійська чи російська);
- аргументи функцій записуються в круглих дужках;
- всі аргументи записуються в тому порядку, який визначений синтаксисом функції;
- значення аргументів повинні відповідати області визначення функції;
- аргументи один від одного відділяються крапкою з комою.

У формулах і функціях допускається введення пробілів, але вони будуть знищені після фіксації формули у клітині.

Синтаксис запису функції: **<ім'я функції>[(**аргументи**)]**. Тобто, спочатку записується ім'я функції, а потім – у круглих дужках

функції можна вводити з клавіатури або вказувати в електронній таблиці за допомогою ЛКМ.

Зауваження: аргументи тригонометричних функцій записуються у радіанах, результат обернених тригонометричних функцій також подається в радіанах. Значення виразів повинне відповідати типу функції.

Приклад 1. Для обчислення квадратного кореня із синуса 36° потрібно записати формулу = **КОРЕНЬ(SIN(РАДИАНЫ(36)))**.

Приклад 2. Розглянемо використання функції ЕСЛИ. Нехай в клітині **D3** міститься деяке число, а в клітину **E3** необхідно записати текст: “**перевиконано**”, якщо значення в **D3>100**; “**виконано**”, якщо **D3=100**; “**не виконано**”, якщо **D3<100**. Для цього випадку формула в клітині **E3** матиме вигляд: =ЕСЛИ(D3>100; “**перевиконано**”; ЕСЛИ(D3=100; “**виконано**”; “**не виконано**”))

Таблиця 4.7.

Деякі функції Excel

Математичні

Синтаксис функції	Пояснення (повернені значення)
ABS(<вираз>)	Модуль (абсолютна величина) <виразу>
ACOS(<вираз>)	Арккосинус <виразу>
ASIN(<вираз>)	Арксинус <виразу>
ATAN(<вираз>)	Арктангенс <виразу>
COS(<вираз>)	Косинус <виразу>
EXP(<вираз>)	Основа натурального логарифма в степені <вираз>
LN(<вираз>)	Натуральний логарифм <виразу>
LOG(<вираз>;<основа>)	Логарифм <виразу> за вказаною <основою>
LOG10(<вираз>)	Десятковий логарифм <виразу>



SIN(<вираз>)	Синус <виразу>
TAN(<вираз>)	Тангенс <виразу>
ГРАДУСЫ(<вираз >)	Перетворює радіани <виразу> в градуси
РАДИАНЫ(<вираз >)	Перетворює градуси <виразу> в радіани
КОРЕНЬ(<вираз >)	Квадратний корінь з <виразу>
ПИ()	Число π , заокруглене до 15 знаків після коми
СУММ(<аргументи>)	Сума <аргументів> (<виразів>)

Статистичні

МАКС(<аргументи>)	Максимум із списку <аргументів>
МИН(<аргументи>)	Мінімум із списку <аргументів>
СРЗНАЧ(<аргументи>)	Середнє значення із списку <аргументів>

Логічні

ЕСЛИ(<умова>; <вираз1>; <вираз2>)	Якщо умова істинна, то повертається значення вираз1, в іншому випадку вираз2
И(<аргументи>)	ІСТИНА, якщо всі <аргументи> мають значення ІСТИНА; в іншому випадку - ХИБНІСТЬ
ИЛИ(<аргументи>)	ІСТИНА, якщо хоча б один <аргумент> має значення ІСТИНА; в іншому випадку – ХИБНІСТЬ
ИСТИНА	Логічне значення ІСТИНА
ЛОЖЬ	Логічне значення ХИБНІСТЬ
НЕ(<вираз>)	Протилежне значення логічного <аргументу>



Пошуку та підстановки

ГПР(X ; <блок>; Y)	Горизонтальний пошук в таблиці даних. Шукається вказане значення X в першому рядку вказаного <блоку>. У випадку успішного пошуку функція повертає значення із стовпця, де міститься знайдене значення, з рядка з номером Y вказаного <блоку>
ВПР(X ; <блок>; Y)	Вертикальний пошук в таблиці даних. Шукається вказане значення X в першому стовпці вказаного <блоку>. У випадку успішного пошуку функція повертає значення з рядка, де міститься знайдене значення, із стовпця з номером Y вказаного <блоку>

Вказаний блок - додаткова таблиця, з якої проводиться відбір даних для їх автоматичної підстановки в основну таблицю. Перший рядок (стовпець) блоку функції ГПР (ВПР) обов'язково повинен містити значення, що впорядковані за зростанням. Кожна клітина блоку повинна містити значення, тобто у блоці не повинно бути порожніх клітин.

Пояснимо роботу функції пошуку та підстановки на такому прикладі. Нехай виконується пошук даних у телефонному довіднику. Спочатку в першому стовпці відшукується прізвище потрібної особи – власника телефону. Для знайденого прізвища з того ж рядка, але з другого стовпця, вибирається номер телефону (аналогічно “працює” функція ВПР).

Розглянемо приклад застосування функції вертикального пошуку. Нехай маємо додаткову таблицю даних про коди машин, їх модель (марку) та ціну. Необхідно побудувати основну таблицю, в деяких стовпцях якої будуть розміщені відомості відповідно про код машини, модель (марку), ціну. Відобразимо побудовану таблицю на малюнку із вказаними адресами клітин. Спочатку наведемо додаткову таблицю, потім – основну. В основній таблиці в клітинах запишемо формули, хоча після формування таблиці на листку буде зображено результат

Додаткова таблиця

	В	С	Д
22	<i>Код</i>	<i>Модель</i>	<i>Ціна</i>
23	1	Ford	10000
24	2	Нива	15000
25	3	МТЗ	8000

Фрагмент основної таблиці

	Е	Ф	Г
2	<i>Код</i>	<i>Модель</i>	<i>Ціна</i>
3	2	=ВПР(Е3;\$B\$23:\$D\$25;2)	=ВПР(Е3;\$B\$23:\$D\$25;3)
4	2	=ВПР(Е4;\$B\$23:\$D\$25;2)	=ВПР(Е4;\$B\$23:\$D\$25;3)
5	1	=ВПР(Е5;\$B\$23:\$D\$25;2)	=ВПР(Е5;\$B\$23:\$D\$25;3)

Основні прийоми роботи в Excel

1. Дані починають заноситися в активну клітину після натискання будь-якої алфавітно-цифрової клавіші.
2. Заповнення вмісту клітини може проводитись у рядку формул (клацнути в ньому ЛКМ) або безпосередньо у клітині.
3. Завершення введення даних відбувається після натискання клавіші **Enter** або клавіш управління курсором. Для відміни змін використовують клавішу **Esc**.
4. Якщо Excel готовий до введення формул, то в рядку стану записано **Готово**; при введенні інформації – **Ввод**; при редагуванні - **Правка**.
5. Редагування вмісту клітини можна виконувати всередині клітини (для цього клацнути в ній двічі ЛКМ) або в рядку формул (зробивши активною потрібну клітину, клацнути в ньому ЛКМ).
6. Знищувати вміст клітини чи блоку зручно клавішею **Delete**.
7. Автоматична заміна вмісту клітини відбувається після виділення клітини селектором та набору нової інформації з її фіксацією.
8. Відмітка блоку за допомогою клавіатури здійснюється клавішами управління курсором при натиснутій клавіші **Shift**. Цим способом зручно користуватися при відмітці великих блоків. Для спрощення виконання деяких команд краще спочатку відмітити блок.

9. З метою недопущення помилок, запис адрес клітин чи блоків у формулах можна здійснювати за допомогою ЛКМ. Наприклад, для вказування аргументів функції **СУММ** достатньо виділити мишею потрібний блок.
10. Для зміни властивостей об'єкта (клітини, блока тощо) можна скористатися правою кнопкою миші або пунктом меню **Формат**. У меню, що розкриється, необхідно вибрати ту команду, яка відповідає поставленому завданню.
11. При необхідності, клітини можна об'єднувати, виконавши команду **Формат / Ячейки / Выравнивание / объединение ячеек**. Для розміщення заголовка посередині таблиці, можна набрати його в клітині першого стовпця над сформованою таблицею, відмітити всі клітини над таблицею в рядку заголовка, скористатися інструментом  **Объединить и поместить в центре**.
12. Відмінити останні зміни може команда **Отменить...** пункту меню **Правка** або відповідний інструмент .
13. Для зручного введення числових рядів можна скористатися командою **Правка / Заполнить / Прогрессия...**.
14. Для транспонування введеної таблиці можна спочатку скопіювати її в буфер обміну, а потім вставити з буфера за допомогою команди меню **Правка / Специальная вставка.../ опція транспонировать** або команди контекстного меню.
15. Якщо ширина стовпця менша, ніж видима частина текстової інформації у клітині, і клітина справа не зайнята, то зображення решти інформації продовжується у сусідній справа клітині. Якщо ж клітина справа зайнята, то видима частина інформації з лівої клітини обрізається по ширині клітини, але вміст клітини не змінюється. Ширину стовпця можна збільшити командою **Формат / Столбец / Ширина...** або транспортуванням межі колонки в адресній смугі. Розмістити інформацію в одній клітині у декілька рядків можна командою **Формат / Ячейки... / Выравнивание / опція переносить по словам**.
16. До кожної клітини можна додати текстову примітку командою меню **Вставка / Примечание**. Клітина з приміткою містить червоний трикутничок у верхньому правому куті. Текст примітки з'являється при наведенні вказівника миші на клітину з приміткою. Для зміни тексту примітки активної клітини потрібно



скористатися командою **Вставка / Изменить примечание** або командою **Изменить примечание** контекстного меню клітини. Клацнувши мишею за межами примітки, можна вийти з режиму редагування.

17. Для одночасної роботи з декількома областями одного аркуша великої таблиці у випадку, коли на екрані відображена тільки її частина, можна розділити аркуш на області командою меню **Окно / Разделить**. Переміщення по таблиці в кожній області відбувається незалежно. Відмінити розділення дозволяє команда **Окно / Снять разделение**.

Для зручності виконання певних дій в Excel можна користуватися клавішами та їх комбінаціями, призначення яких подано у таблиці 4.8.

Таблиця 4.8.

Призначення деяких клавіш та їх комбінацій

Клавіша або комбінація клавіш	Призначення
Ctrl+1	Відобразити діалогове вікно формату клітини
Shift+F3	Відобразити діалогове вікно майстра функцій
F4	Перетворити адресу клітини в абсолютну
F5	Перемістити селектор у клітину з вказаною адресою
Alt+Enter	Перейти на новий рядок в активній клітині
Ctrl+Enter	Заповнити відмічений діапазон поточним значенням
Ctrl+F3	Присвоїти ім'я клітині (блоку клітин)
Shift+F5	Відобразити вікно пошуку та заміни
F6 (Shift+F6)	Перейти до наступного (попереднього) вікна у списку



F11	Автоматичне створення діаграми
Shift+F11	Вставка нового аркуша
Ctrl+;	Ввести поточну дату
Ctrl+Z	Відмінити останню виконану дію

Прийоми роботи з мишею

Миша використовується для вибору пунктів меню, відмітки блоків, зміни висоти рядків і ширини стовпців, вибору команд з рядка швидкого виклику команд, швидкого переходу в іншу частину таблиці. Частіше використовується ліва кнопка миші, клацання якою відповідає натисканню на клавішу **Enter**. Переміщення миші по столу викликає переміщення відповідного вказівника миші.

Для вибору певного об'єкта (зробити активним об'єкт, вибрати пункт меню, команду) необхідно навести вказівник миші приблизно на середину потрібного об'єкта і клацнути ЛКМ.

При виконанні різних дій вказівник миші набуває вигляду, зображеного на рис. 4.31.

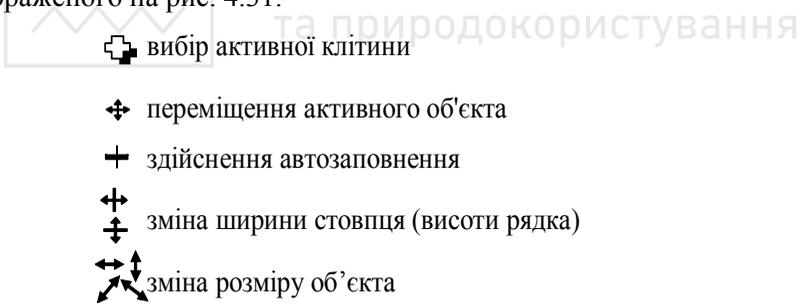


Рис. 4.31. Вигляд вказівника миші та відповідна дія

Щоб перейти на клітину з певною адресою (зробити клітину активною), необхідно навести вказівник миші на потрібну клітину і клацнути ЛКМ.

Для відмітки блоку клітин необхідно помістити вказівник миші на кут блоку (одну з клітин у вершинах уявного прямокутника, в якому містяться всі клітини блоку), натиснути ЛКМ і, не відпускаючи її, протягнути вказівник до протилежного кута блоку (протилежної



вершини прямокутника), після чого відпустити ЛКМ. Клітина, з якої починалася відмітка блоку, не буде зафарбована контрастним кольором.

Щоб відмітити весь рядок або стовпець, потрібно клацнути ЛКМ на відповідному номері рядка або на назві стовпця. Відмітка несуміжних елементів таблиці проводиться при натиснутій клавіші **Ctrl**.

Для зміни висоти рядка або ширини стовпця слід помістити вказівник миші на границю рядка або стовпця у відповідній адресній смузі, натиснути ЛКМ і, не відпускаючи її, перемістити вказівник на потрібну відстань, після чого відпустити кнопку миші.

Автоматичне заповнення блоку клітин інформацією із сусідньої клітини (відміченого блоку) здійснюється протягуванням вказівником миші маркера автозаповнення клітини на необхідну кількість клітин. При цьому інформація копіюється із змінами, які визначаються закономірністю зміни даних у відміченому блоці або в сусідніх клітинах. Якщо потрібно вибрати спосіб автозаповнення самостійно, перетягування вказівника миші виконується при натиснутій ПКМ.

Для **переміщення активної клітини** (блоку клітин) можна навести вказівник миші на нижній край активної клітини (блоку) так, щоб він мав вид стрілки (рис. 4.31, рядок 2) і перетягнути вміст клітини на потрібне місце при натиснутій ЛКМ. Для отримання копії при цьому необхідно тримати натиснутою клавішу **Ctrl**.

Діаграми та графіки в Excel

Діаграма – засіб наочного графічного зображення інформації, який використовується для порівняння декількох величин або декількох значень однієї величини, аналізу зміни їх значень тощо.

Побудова діаграми проводиться на основі рядів даних.

Рядом даних називають діапазон клітин у межах одного рядка або стовпця.

На одній діаграмі, в залежності від її типу, можна відобразити кілька рядів даних. Всі дані, які потрібно відобразити на діаграмі, слід вводити в таблицю без пропусків.

В Excel можна будувати діаграми наступних **типів**: гістограма (**Гистограмма**) – відображає дані таблиці у вигляді стовпчиків; графік (**График**) – відображає розвиток процесів у часі або за категоріями; кругова (**Круговая**) – відображає частку кожного значення в загальній

сумі (як правило, у відсотках); точкова (**Точечная**) – пари значень (x, y) відображаються точками, які можуть бути з'єднані між собою лініями (до цього виду діаграм відносяться звичайні графіки залежності Y від X); поверхнева (**Поверхность**) – відображає зміни значень за двома вимірами у вигляді поверхні. Кожен тип діаграм має декілька видів. Діаграма може розміщуватися на тому ж аркуші, що і дані для неї, або на окремому аркуші чи на будь-якому іншому аркуші.

Побудова діаграм



Якщо вибрати інструмент **Мастер диаграмм**, відкриється діалогове вікно **Мастер диаграмм**. Для продовження роботи тут можна використовувати наступні кнопки: **Отмена**, **<Назад, Далее>**, **Готово**. Процес побудови діаграми (графіка) розбитий на 4 кроки.

Крок 1: вибираємо **тип** (зліва) і **вид** (справа) діаграми (рис. 4.32).

Крок 2: вказуємо **діапазон даних**, а на іншій вкладці цього ж вікна при потребі додатково описуємо **ряди даних**. Діапазон даних можна вказувати за допомогою клавіатури або ЛКМ на листі робочої книги. При необхідності треба уточнити, де саме розміщені ряди даних: **в стовпчиках** чи **в рядках**. Після повідомлення цієї інформації можна побачити загальний вигляд діаграми у діалоговому вікні майстра діаграм.

Крок 3: вказуємо **параметри** діаграми. На вкладці **Заголовки** записуються **назви: діаграми, осі X** (категорій), **осі Y** (значень). Вкладки **Оси**, **Линии Сетки**, **Легенда** дозволяють вимикати чи вмикати відповідні елементи діаграми та керувати їх параметрами. Вкладки **Подписи данных** та **Таблица данных** служать для додаткового оформлення графіка шляхом виведення в якості надписів значень функції або виведення повної таблиці значень функції.

Крок 4: вибираємо **місце розташування** діаграми (на окремому аркуші, чи на активному).

Створену діаграму можна редагувати. Адже вона складається з набору окремих елементів, які можна змінювати, переносити, знищувати. При клацанні на елементі діаграми, він виділяється маркерами. Відкрити діалогове вікно для форматування елемента діаграми найпростіше через контекстне меню цього елемента. Для кардинальної зміни діаграми краще побудувати нову діаграму, а попередню – виділити і знищити з допомогою клавіші **Delete**. Аркуш

діаграми знищується командою **Удалить лист** пункту меню **Правка** або відповідною командою контекстного меню аркуша.

Всі зміни, виконані в клітинах, на базі яких будується діаграма, автоматично відображаються на діаграмі.

Для побудови точкової діаграми потрібно попередньо відсортувати дані зазначеннями, які будуть відкладені на осі абсцис.

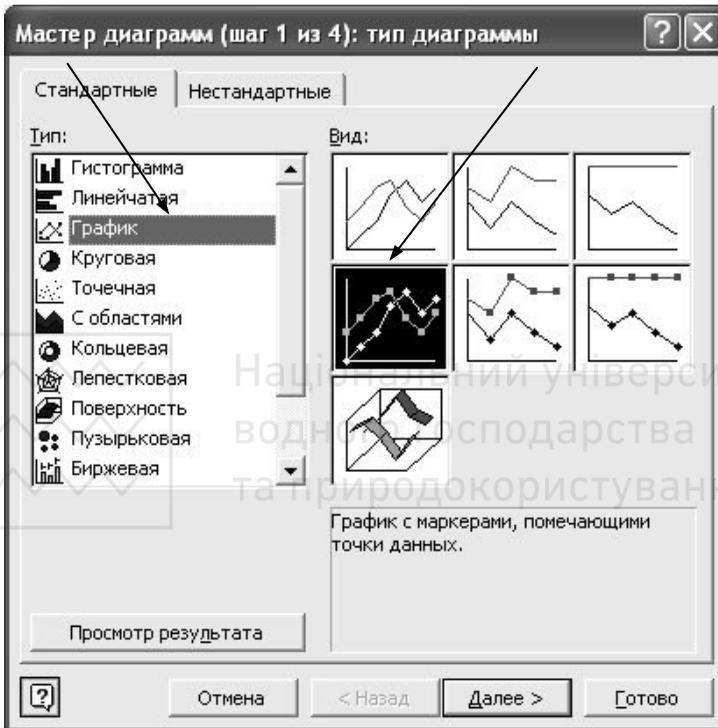


Рис. 4.32. Крок 1 побудови діаграми

Сортування рядків даних таблиці

Порядок виконання роботи:

1. Відмітити блоком всі дані таблиці, які повинні сортуватися.
2. Виконати команду **Сортировка** пункту меню **Данные**.
3. У вікні **Сортировка диапазона**:
 - заповнити поле **Сортировать по**, тобто вказати блок клітин одного стовпця, який сортується в першу чергу;



- при потребі вказати діапазон клітин стовпця, які будуть сортуватися потім і в останню чергу;
 - вибрати порядок сортування (за зростанням чи спаданням).
4. Натиснути **ОК** для виконання сортування.

Математична обробка даних

Пошук кореня рівняння

Порядок розв'язування задачі:

1. Ввести початкове значення змінної, наприклад, у клітину **В3**.
2. В клітину **С3** занести формулу, яка є лівою частиною рівняння, зведеного до вигляду $f(x)=0$. У формулі в якості змінної записується адреса клітини **В3**, де міститься початкове значення змінної.
3. Виконати команду меню **Сервіс / Подбор параметра**.
4. У вікні **Подбор параметра** заповнити поля:
 - **Установить в ячейке** - адреса клітини, в якій записане значення лівої частини рівняння (**С3**);
 - **Значение** - значення, якому повинна дорівнювати ліва частина рівняння (у наведеному прикладі 0);
 - **Изменяя значение ячейки** - адреса клітини, де розміщене початкове значення змінної (**В3**), і буде знайдений розв'язок рівняння;
 - Натиснути **ОК** для відшукування розв'язку рівняння.
5. Натиснути ще раз кнопку **ОК** для збереження результатів в клітинах.

Обчислення визначника матриці

Зауваження: обчислити визначник можна тільки для квадратної матриці.

Порядок розв'язування задачі:

1. Ввести початкову матрицю.
2. Зробити активною клітину, де буде розміщений визначник матриці.
3. Вибрати інструмент **Вставка функції**, вказати категорію функцій **Математические**, вибрати функцію **МОПРЕД**, натиснути кнопку **ОК** або клавішу **Enter**.

4. У вікні функції **МОПРЕД** ввести діапазон клітин (масив), де розміщена початкова матриця, або відмітити цей діапазон за допомогою миші.
5. Натиснути кнопку **ОК** або клавішу **Enter**.

Обчислення оберненої матриці

Зауваження:

- обернена матриця будується тільки для квадратної матриці, визначник якої відмінний від нуля;
- операції роботи з матрицями (масивами) завершуються натисканням комбінації клавіш **Ctrl+Shift+Enter**.

Порядок розв'язування задачі:

1. Ввести початкову матрицю.
2. Відмітити діапазон клітин, де буде розміщена обернена матриця, розміри якого збігаються із розмірами початкової матриці.
3. Вибрати інструмент **Вставка функції**, вказати категорію функцій **Математические**, вибрати функцію **МОБР**, натиснути кнопку **ОК** або клавішу **Enter**.
4. У вікні функції **МОБР** записати діапазон клітин (масив), де розміщена початкова матриця, або відмітити цей діапазон за допомогою миші.
5. Натиснути комбінацію клавіш **Ctrl+Shift+Enter**.

Обчислення добутку матриць

Зауваження: добуток двох матриць можна знайти для таких матриць, в яких кількість стовпців першої матриці дорівнює кількості рядків другої.

Порядок розв'язування задачі:

1. Ввести початкові матриці (кожен елемент у окрему клітину).
2. Відмітити блок клітин, де буде розміщена матриця-добуток, у якій кількість рядків визначається кількістю рядків першої матриці, а кількість стовпців - кількістю стовпців другої матриці.
3. Вибрати інструмент **Вставка функції**, вказати категорію функцій **Математические**, вибрати функцію **МУМНОЖ**, натиснути кнопку **ОК** або клавішу **Enter**.
4. У вікні функції **МУМНОЖ** у полі **Массив1** вказати блок клітин, де міститься перша матриця добутку, у полі **Массив2** вказати діапазон клітин, де розміщена друга матриця добутку.
5. Натиснути комбінацію клавіш **Ctrl+Shift+Enter**.



Пошук оптимальних розв'язків

Задачі оптимізації зустрічаються у повсякденному житті досить часто. Здебільшого це економічні задачі, які виникають в різних сферах діяльності людини. Як правило, в таких задачах при певних обмеженнях необхідно знайти оптимальне (максимальне або мінімальне) значення деякої величини, яка залежить від інших величин. Коректність поставленої задачі та існування розв'язку виходять за межі вивчення ЕТ. Вважатимемо, що розв'язок задачі існує. Розв'язування сформульованої задачі з допомогою ЕТ почнемо з її математичної постановки. Для цього введемо відповідні позначення і з допомогою математичних співвідношень запишемо систему обмежень та функцію мети.

Приклад задачі, яка задана математичною моделлю. Нехай потрібно знайти максимальне значення функції мети $F(x_1, x_2, x_3, x_4)$, де значення змінних x_1, x_2, x_3, x_4 невід'ємні і задовольняють системі нерівностей:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \leq b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \leq b_3 \end{cases}$$

Зауваження: в системі нерівностей не обов'язково повинні бути використані знаки \leq .

Порядок відшукання оптимального розв'язку:

1. Приймемо початкові значення змінних x_1, x_2, x_3, x_4 , наприклад, рівними 0. Занесемо ці значення відповідно у клітини з адресами **A1, B1, C1, D1**.
2. Запишемо матрицю коефіцієнтів $a_{11}, a_{12}, a_{13}, \dots, a_{33}, a_{34}$ при змінних x_1, x_2, x_3, x_4 системи обмежень у клітини **A2, B2, C2, D2, A3, B3, C3, D3, A4, B4, C4, D4**.
3. У клітину **E2** запишемо ліву частину першого обмеження, яке буде мати вигляд **A2*\$A\$1+B2*\$B\$1+C2*\$C\$1+D2*\$D\$1**. Скопіюємо формулу з клітини **E2** у клітини **E3** та **E4** для отримання формул, що відповідають лівим частинам відповідно другого та третього обмеження системи.
4. Запишемо праві частини b_1, b_2, b_3 системи обмежень відповідно у клітини **F2, F3, F4**.

5. У клітину **F1** запишемо формулу, що відповідає функції $F(x_1, x_2, x_3, x_4)$.
6. Виконаємо команду **Сервис / Поиск решения**. У вікні **Поиск решения**: в полі **Установить целевую** вказуємо клітину **F1**. Встановлюємо перемикач **Равной максимальному значению**. У полі **Изменяя ячейки** вказуємо блок **A1:D1**, де містяться початкові значення змінних. У цих клітинах буде розміщений розв'язок задачі. Запишемо обмеження задачі, натиснувши кнопку **Добавить**. Відкриється вікно **Добавление ограничения**. Будемо записувати обмеження: змінні не менші від 0. Для цього в полі **Ссылка на ячейку** записуємо блок **A1:D1**. У середньому полі введення вибираємо зі списку символ \geq . У полі **Ограничение** записуємо число 0. Натискаємо кнопку **ОК**. Для запису системи обмежень знову натискаємо кнопку **Добавить**. В полі **Ссылка на ячейку** записуємо блок **E2:E4**. У середньому полі введення вибираємо зі списку символ \leq . У полі **Ограничение** вказуємо блок **F2:F4**. Натискаємо кнопку **ОК**.
7. Для розв'язання задачі необхідно натиснути кнопку **Выполнить**.
8. Щоб зберегти знайдені значення необхідно вказати перемикач **Сохранить найденное решение** та натиснути кнопку **ОК**.

Дослідження зв'язку між двома змінними

В інженерних дослідженнях часто виникає задача встановлення зв'язку (аналітичної залежності) між показниками, які позначимо змінними X і Y . Цей зв'язок може бути функціональним або стохастичним. Стохастичний (кореляційний) зв'язок показує співвідношення між випадковими величинами, які можуть перебувати під впливом інших величин (наприклад, зв'язок між величиною стоку річки Y і середньодобовою температурою повітря X ; зв'язок між концентрацією іонів важких металів у стічній воді і продуктивністю очисних споруд тощо). Результати спостережень величин X і Y задають у вигляді таблиці.

X	x_1	x_2	x_3	...	x_i	...	x_n
Y	y_1	y_2	y_3	...	y_i	...	y_n

Якщо на координатній площині відкласти точки A_i з координатами $(x_i; y_i)$, то одержимо поле розсіяння величин X і Y .

Національний університет
та державного економічного управління

Наближення табличних даних математичною залежністю $Y=f(X)$ називається **апроксимацією** даних.

Найбільш ефективним методом апроксимації є метод найменших квадратів, який полягає в тому, що необхідно провести таку пряму чи плавну криву лінію через скупчення точок, щоб вона перебувала на мінімальній відстані від більшості з них. При цьому сума квадратів відстаней від кривої до експериментальних точок повинна бути мінімальною. Тісноту зв'язку встановлює кореляційний аналіз з допомогою числового показника - коефіцієнта кореляції для прямої лінії або кореляційного відношення для всіх видів зв'язку.

Побудова графіка, тобто графічне вирівнювання даних, дозволяє виявити характер зв'язку між величинами X і Y . Чим ближче точки до лінії, тим більша кореляція між величинами X і Y . Характер розташування точок може дати уявлення про тісноту зв'язку між величинами X і Y , а також підказати вид формули, що виражає зв'язок цих величин. Найпростішим видом зв'язку між величинами X і Y є лінійний зв'язок. Тут рівняння зв'язку між показниками (рівняння регресії) має вигляд: $Y = a_0 + a_1 \cdot X$. Коефіцієнти рівняння регресії за методом найменших квадратів обчислюють за формулами:

$$a_0 = \frac{\sum y_i * \sum x_i^2 - \sum x_i * \sum x_i y_i}{n * \sum x_i^2 - (\sum x_i)^2}; a_1 = \frac{n * \sum x_i y_i - \sum x_i * \sum y_i}{n * \sum x_i^2 - (\sum x_i)^2}$$

Коефіцієнт кореляції визначається співвідношенням:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 * \sum (y_i - \bar{y})^2}},$$

де \bar{x} , \bar{y} - середньоарифметичні значення відповідно факторів X і Y .

Порядок обчислення коефіцієнтів:

1. Введемо, наприклад, у стовпці А значення фактора X , а у стовпці В – відповідні значення фактора Y .
2. Для визначення коефіцієнта a_0 скористаємося функцією **ОТРЕЗОК**(<відомі значення Y >; <відомі значення X >) або **ЛИНЕЙН**(<відомі значення Y >; <відомі значення X >). Тут **відомі значення** - адреса блоку, де розміщені значення відповідних факторів. При використанні функції **ЛИНЕЙН** потрібно відмітити дві сусідні клітини в одному рядку і



завершити введення функції натисканням комбінації клавіш **Ctrl+Shift+Enter**. В результаті отримаємо два коефіцієнти, перший з яких a_1 , другий - a_0 .

3. Для визначення коефіцієнта a_1 можна використати функцію **НАКЛОН**(<відомі значення Y>; <відомі значення X>) або **ЛИНЕЙН**(<відомі значення Y>; <відомі значення X>).
4. Значення коефіцієнта r можна знайти з допомогою функції **КОРРЕЛ**(<масив1>; <масив2>) або **ПИРСОН**(<масив1>; <масив2>). Тут <масив1>, <масив2> - набори чисел або адреси блоків, в яких розміщені відповідні значення факторів. Порядок запису блоків не впливає на результати обчислень.

При інтерпретації результатів розрахунку необхідно мати на увазі наступне: від'ємне значення коефіцієнта кореляції вказує на те, що **кут нахилу** прямої до осі X тупий (від 90 до 180°); абсолютне значення коефіцієнта кореляції **не повинно перевищувати 1**; якщо абсолютне значення коефіцієнта кореляції більше від $0,8$, то зв'язок між факторами **сильний**, якщо воно знаходиться в межах від $0,6$ до $0,8$ - **середньої** сили, і, нарешті, якщо абсолютне значення менше від $0,6$ - **слабкий**.

Для побудови графіка залежності $y=f(x)$ блок з парами значень факторів потрібно відсортувати за стовпчиком X .

Щоб знайти прогнозовані значення Y для конкретних значень фактора X , необхідно доповнити таблицю порожнім рядком, в якому записати значення фактора X і за рівнянням регресії визначити значення фактора Y .

Прогнозування за межами діапазону спостереження даних можна проводити не більше, ніж на третину ширини діапазону як вліво, так і вправо. В іншому випадку прогнозування буде недостовірним.

Контрольні запитання і завдання

1. Поняття про електронну таблицю та її призначення.
2. Яке розширення має файл з ЕТ Excel?
3. Яка структура документа Excel?
4. Назвіть дії, які можна виконувати з аркушами ЕТ.
5. Як позначаються стовпці і рядки ЕТ і скільки їх може бути?
6. Яким чином змінюють ширину стовпця ЕТ?
7. Перелічіть типи даних, які опрацьовують в Excel.
8. Опишіть процедуру введення даних в клітину та їх редагування.

9. Сформулюйте правила запису формули у клітину.
10. Наведіть приклади повідомлень про помилки.
11. Як виконати копіювання формули?
12. Що таке абсолютна і змішана адреси клітин?
13. Назвіть категорії стандартних функцій Excel.
14. Як обчислити суму значень у стовпці?
15. Дайте визначення блоку клітин.
16. Назвіть типи діаграм Excel та їх призначення.
17. Опишіть послідовність побудови діаграми.
18. Як реалізувати розгалуження в Excel?
19. Які є засоби в Excel для розв'язування нелінійного рівняння?
20. В чому суть задачі апроксимації даних?





4.4. MS PowerPoint

Сучасному фахівцю доводиться часто готувати і проводити публічні виступи як у межах своєї фірми чи організації, так і поза нею. Цьому сприяють засоби програми PowerPoint, що є складовою частиною Microsoft Office. Вони дають можливість досить швидко підготувати супровід виступу з використанням ПК.

Набір слайдів, що супроводжує виступ, називається презентацією.

Кожен із слайдів, які включаються до презентації, може містити: текст, таблицю, діаграму, рисунок, організаційну діаграму, відеокліп, звуковий супровід, гіперпосилання на інші слайди та документи, які розміщені в даному комп'ютері або в Internet. Окремі об'єкти слайдів можуть мати ефекти анімації.

При підготовці презентації можна використовувати фрагменти документів Word, Excel чи Access.

Створені презентації можна демонструвати:

- на моніторі для невеликого кола осіб;
- на екрані за допомогою мультимедійного проєктора;
- на екрані за допомогою кодоскопа із використанням прозорої плівки;
- на екрані за допомогою діапроєктора, використовуючи слайдофільми;
- у друкованому вигляді як роздатковий матеріал.

Вікно PowerPoint

Запуск PowerPoint здійснюється подвійним клацанням ЛКМ на



ярлику програми. Вікно PowerPoint (рис. 4.33) має структуру, аналогічну до вікон інших програм Microsoft Office, тобто має смугу заголовка, рядок меню, панелі інструментів, робочу область і рядок стану. Особливість вікна PowerPoint полягає у тому, що ліворуч від горизонтальної смуги прокручування є кнопки режимів перегляду слайдів. Ці кнопки дублюють відповідні команди з меню **Вид**.

Розглянемо окремі режими перегляду слайдів. **Обычний режим** (кнопка ) використовується для створення нового чи коригування існуючого слайда (вкладка **Слайды** , рис.4.33), а також для

Национальний університет
внесення змін у загальну структуру презентації. Вкладка **Структура**
та природокористування
зокрема, надає можливість формувати **нотатки** для кожного із
слайдів.

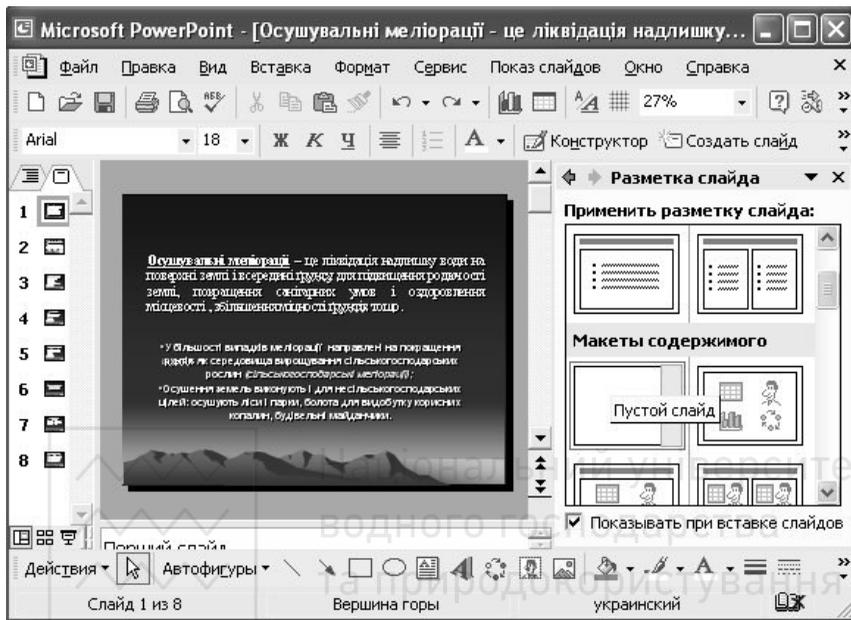


Рис. 4.33. Вікно PowerPoint

***Нотатками** називаються додаткові пояснення у вигляді тексту, які можуть бути створені автором для будь-якого із слайдів і під час демонстрації не відображаються.*

Режим сортировщика слайдов (кнопка ). У робочій області PowerPoint відображається вся презентація, причому кожен слайд виводиться у зменшеному вигляді, що дозволяє швидко орієнтуватися у вмісті кожного слайду. На етапі формування послідовності показу слайдів використовують режим сортувальника. В цьому режимі можна вказати тривалість показу кожного слайда, вибрати анімаційні ефекти переходу від одного слайда до іншого тощо.

Показ слайдов (кнопка ) використовується для демонстрації презентації. В ньому кожен слайд виводиться на весь екран. Вихід із режиму показу здійснюють клацанням клавіші **Esc** або виконанням команди **Завершити показ слайдов** у контекстному меню.



Для доповнення презентації новим слайдом зупиняють курсор між потрібними слайдами та виконують команду меню **Вставка / Создать слайд** або натискають **Ctrl+M**. Перемістити слайд в інше місце презентації можна звичайним перетягуванням при натиснутій ЛКМ. Для коригування будь-якого слайду достатньо на ньому двічі клацнути ЛКМ. Виділений слайд вилучається клавішею **Delete**.

Завершення роботи у програмі PowerPoint здійснюється закриттям її вікна кнопкою  або виконанням команди меню **Файл / Выход**.

Створення презентації

Для створення нової презентації виконують команду **Файл / Создать**, після чого з'являється область задач при запуску. Область задач розташовується у вікні програми в центрі або справа. Вона дозволяє відкривати існуючу презентацію або створювати нову. При створенні нової презентації можна орієнтуватися на вже існуючі презентації або їх шаблони.

Найбільше можливостей при створенні презентацій дає вибір способу **Новая презентация**. Для кожного слайда потрібно спочатку вибрати розмітку, тобто вказати, з яких місцезаповнювачів він буде складатися, потім заповнити його демонстраційним матеріалом та вибрати потрібне оформлення (фоновий рисунок, палітру кольорів, вид шрифту тощо). Як правило, спочатку створюють кілька слайдів, заповнюють їх даними і підбирають відповідне оформлення, а потім продовжують створення презентації на тлі вибраного оформлення.

Під час роботи з окремим слайдом можна змінювати не тільки склад місцезаповнювачів (для цього виконують команду контекстного меню **Разметка слайда**, потім вибирають потрібний макет тексту в області задач), а й розміри та розташування кожного місцезаповнювача. Для зміни розмірів перетягують спеціальні маркери на границях місцезаповнювачів при натиснутій ЛКМ. Переміщують об'єкти захопленням ЛКМ заштрихованої границі місцезаповнювача або ж очікують появи вказівника миші у вигляді перехрестя двонаправлених стрілок, після чого натискають ЛКМ і здійснюють переміщення.

Для вилучення виділеного місцезаповнювача натискають клавішу **Delete**. При цьому у виділеному місцезаповнювачі не повинно бути курсора.



Вставка таблиць і діаграм

Подання інформації у табличному вигляді дає змогу спростити її пошук та аналіз. Тому в презентаціях часто використовують таблиці. У PowerPoint є засоби створення таблиць із нульового рівня, тобто з самого початку. Але оскільки засоби PowerPoint, призначені для побудови таблиць, бідніші за аналогічні засоби Word чи Excel, то при роботі з програмою використовують таблиці, підготовлені заздалегідь в інших додатках.

Таблиці Word застосовують для текстових даних, а також для числових початкових даних. Таблиці Excel фігурують при поданні даних, обчислених за формулами, а також у випадках побудови за цими даними діаграм. Таблиці Access використовуються при виборі даних із кількох зв'язаних таблиць.

Вставка таблиць з інших додатків здійснюється через буфер обміну. Створення нової таблиці на місці виконується командою меню **Вставка / Таблиця...** Далі вказують кількість рядків та стовпців. Заповнюють таблицю за правилами Word.

Організаційні діаграми

Для зображення ієрархічної залежності між елементами системи використовуються організаційні діаграми, що будуються за допомогою додатка Microsoft Organization Chart. Такі діаграми створюють командою меню **Вставка / Організаційна діаграма...** Після цього у діалоговому вікні вказують тип діаграми та клацають ОК.

Під час створення та коригування організаційної діаграми використовують команди, що задаються кнопками панелі інструментів **Організаційна діаграма** (рис. 4.34).

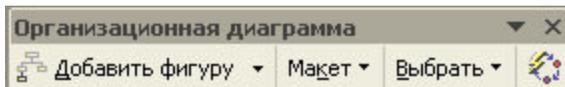


Рис. 4.34 Панель інструментів Організаційна діаграма

*Кожен елемент ієрархії подається прямокутною фігурою, що називається **рамкою**.*

Для заповнення рамки текстом досить клацнути на ній ЛКМ. Фіксація набраного тексту здійснюється клацанням поза межами



рамки. Текстова інформація вводиться та форматується за правилами Word.

Зайві рамки вилучають за допомогою клавіші **Delete**, попередньо їх виділивши. Нові рамки додають відповідними кнопками панелі інструментів.

Вставка рисунків

Рисунки в презентаціях використовують для:

- демонстрації зовнішнього вигляду зразка виробу, товару, будинку, мапи тощо;
- надання емоційного забарвлення презентації (привернути увагу, зняти напругу і т. д.).

У першому випадку використовують рисунки, підготовлені фахівцями конкретної предметної області, які зберігаються у графічних файлах. У другому випадку – рисунки стандартних додатків Microsoft Clip Gallery, Coral Draw, AutoCad та ін.

Для вставки рисунка з файлу можна виконати команду меню **Вставка / Рисунок/ Из файла...** та вибрати потрібне. Після вставки рисунка його можна відформатувати: змінити масштаб, обрізати, змінити контрастність, яскравість тощо.

Вставка звуків і фільмів

Звуковий супровід належить не до якогось конкретного місцезаповнювача, а до всього слайда. Вставку звуку у слайд здійснюють командою меню **Вставка / Фильмы и звук / Звук из**

файла... На слайді після цього з'явиться значок звуку . За замовчуванням звук пролине автоматично. Аналогічно виконують вставку звуку із колекції Clip Gallery.

Слайди можна також доповнювати фільмами (з файлу чи з колекції).

PowerPoint дає можливість самостійно записати мовний супровід слайду. Для цього комп'ютер повинен бути оснащений звуковою картою, мікрофоном і колонками. Запис супроводу здійснюють у такій послідовності: активізують потрібний слайд, виконують команду меню **Вставка / Фильмы и звук / Записать звук**. Для початку запису

клацають мишею на кнопці  **Запись**. Для завершення запису



клацають мишею на кнопки



Остановить. Прослухати запис

можна після клацання мишею кнопки



Воспроизведение. В полі **Название:** вікна **Звукозапись** вказують ім'я запису. Для завершення процесу необхідно клацнути кнопку ОК.

Використання гіперпосилань

PowerPoint надає користувачам засоби для організації нелінійного способу перегляду презентацій. Для цього використовують **гіперпосилання**, які можна подати у вигляді підкресленого тексту, значка або кнопки. Клацання мишею на гіперпосиланні приводить до переходу за вказаною адресою, що дає змогу перейти до того чи іншого слайда. Крім того гіперпосилання дозволяють виводити на екран об'єкти, які не є слайдами, наприклад, додатки.

Для створення гіперпосилання потрібно:

- виділити текст чи інший об'єкт, який буде гіперпосиланням;
- вибрати команду меню **Вставка / Гиперссылка**;
- у вікні діалогу **Добавление гиперссылки** у списку **Связать с:** вказати тип об'єкта і в центральній частині вікна вибрати потрібний об'єкт. Можна клацнути кнопку **Подсказка...** та ввести відповідний текст для супроводу гіперпосилання.
- Перехід між слайдами можна здійснювати гіперпосиланнями у вигляді **кнопок**, які розміщуються на слайдах аналогічно об'єктам **Автофигуры** програми Word.

Надписи на кнопках вказують на їх призначення. Для нанесення надпису об'єкт виділяють, а у контекстному меню вибирають команду **Добавить текстовую строку**. Текст вводять в позицію курсора і фіксують клавішею **Enter**.

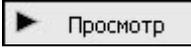
Ефекти анімації

Пожвавити сприйняття матеріалу дозволяють засоби анімації, які створюють ілюзію руху як окремих елементів слайда, так і послідовності слайдів під час зміни одного з них іншим. Анімація окремих елементів слайда встановлюється в режимі слайдів, а послідовності – в режимі сортувальника.

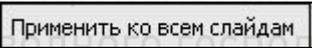
Для встановлення ефектів анімації окремих елементів потрібно:

- виділити потрібний елемент;



- вибрати потрібний ефект;
- клацнути на кнопці  у нижній частині панелі з ефектами;
- у разі необхідності, клацнути кнопку  **Настройка анимации** і виконати виправлення.

Для створення ефектів при переході від одного слайду до іншого потрібно:

- встановити режим сортувальника слайдів;
- виділити послідовність слайдів, клацнувши мишею спочатку на першому, а потім, при натиснутій клавіші **Shift**, – на останньому з них;
- виконати команду меню **Показ слайдов / Смена слайдов...**;
- вибрати відповідний ефект;
- вказати звук, швидкість та спосіб зміни слайдів;
- клацнути кнопку .

Підготовка до демонстрації

Після того, як презентацію створено, потрібно її налаштувати і провести репетицію. Для цього виконують команду **Показ слайдов / Настройка презентации...** Спочатку визначають, яким чином буде здійснено перехід від одного слайда до іншого та які саме слайди будуть включені у демонстрацію, що готується. Якщо зміна слайдів відбувається автоматично, то необхідно задати інтервали часу показу кожного слайда.

Перехід від одного слайда до іншого може відбуватися за вказівкою або автоматично. У PowerPoint є такі види керування переходами:

- **керування доповідачем** (доповідач вказуватиме час переходу до наступного слайда клацанням ЛКМ);
- **керування користувачем** (слайди займатимуть тільки центральну частину екрана; крім них на екрані буде виводитися меню для керування слайдами, панель інструментів Internet та панель задач Windows);
- **автоматичне керування.**

PowerPoint дає змогу зберігати в одній презентації варіанти різних комбінацій послідовностей слайдів, що можуть знадобитися при різних демонстраціях. Для підбору слайдів у демонстрацію потрібно виконати команду меню **Показ слайдов / Произвольный показ...**. Далі у діалоговому вікні необхідно клацнути кнопку **Создать**, у полі **Имя произвольного показа**: вікна **Задание произвольного показа** ввести назву демонстрації та створити список потрібних слайдів, формуючи його із наявних.

Потрібний варіант демонстрації слайдів вказують командою меню **Показ слайдов / Настройка презентации...**. У рамці **Слайды** ставлять помітку біля опції **произвольный показ**: і нижче у списку вибирають ім'я показу.

Вибір способу зміни слайдів здійснюють у вікні діалогу **Настройка презентації**, де у рамці **Смена слайдов** вибирають один із варіантів: **вручную** чи **по времени**. Для варіанту **по времени** спочатку встановлюють інтервал часу, протягом якого буде демонструватися той чи інший слайд. Хронометраж можна задавати автоматично або в ручному режимі.

Демонстрація слайдів

Для запуску презентації з метою показу слайдів найчастіше застосовуються:

- запуск презентації у PowerPoint (при цьому можна коригувати слайди, зупиняти показ);
- запуск презентації з робочого столу (після показу на екрані відновлюється робочий стіл);
- запуск файлу показу слайдів (демонстрація презентації можлива на ПК, де не встановлено PowerPoint);
- запуск довільного показу слайдів (можна демонструвати не всі слайди презентації, причому в обраному порядку).

Якщо у PowerPoint завантажено потрібну презентацію, то розпочати її показ можна одним із способів:

- клацнути ЛКМ на кнопці **Показ слайдов** у лівому нижньому кутку вікна PowerPoint;
- вибрати команду меню **Показ слайдов / Начать показ**;
- вибрати команду меню **Вид / Показ**;
- натиснути клавішу **F5**.

Для запуску презентації з робочого столу потрібно:



створити ярлик презентації;

• клацнути на значку презентації ПКМ і у контекстному меню вибрати команду **Показать**.

Підготовка файлу показу слайдів здійснюється таким чином:

- вибрати команду меню **Файл / Сохранить как**;
- у вікні діалогу **Сохранение документа** встановити **Тип файла** як **Демонстрация PowerPoint (*.pps)**;
- клацнути кнопку **Сохранить**;
- створити ярлик для файлу показу слайдів;
- клацнути двічі на ярлику для запуску файлу показу.

Приклад створення презентації

Опишемо послідовність дій для підготовки презентації на тему “Осушувальні меліорації”:

1. Завантажити програму **PowerPoint** і при потребі виконати команду **Файл / Создать / Новая презентация**.
2. Виконати команду **Вставка/ Создать слайд** (або клацнути **Ctrl+M**) стільки разів, скільки це потрібно, наприклад, 7. Зайві слайди можна знищувати клавішею **Delete**.
3. Клацнути на першому слайді та вибрати для нього потрібну розмітку **Формат / Разметка слайда / Титульный слайд**.
4. Клацнути у першому місцезаповнювачі та ввести відповідний текст. Аналогічно заповнити другий місцезаповнювач, оформивши його як маркований список.
5. Клацнути послідовно на першому та другому місцезаповнювачах та підібрати для них зручне місцерозташування.
6. Виділити послідовно набрані тексти та вказати для них відповідні шрифт, розмір шрифту та колір літер. Наприклад: для першого – **Times New Roman, 24, синій**; для другого - **Arial, 20, жовтий**.
7. Клацнути на другому слайді та вибрати для нього потрібну розмітку **Формат / Разметка слайда / Заголовок, схема или организационная диаграмма**.
8. Виконати дії аналогічно до п. 4.
9. Клацнути двічі на значку діаграми та вибрати тип **организационная диаграмма**. Заповнити фігури діаграми



текстами. Відразу виконати коригування розмірів діаграм та вибрати її місцезнаходження.

10. Розмітка 3, 4 і 5-го слайдів однакова: **Заголовок и текст**.
11. Розмітки трьох наступних слайдів такі: слайд 6 – **Заголовок и объект**; слайд 7 - **Заголовок и текст в две колонки**; слайд 8 – **Заголовок, текст и объект**.
12. Спочатку всі слайди оформимо однаково. Для цього застосуємо команду **Формат / Оформление слайда** та вкажемо один із шаблонів, наприклад, **Вершина горы.pot**.
13. При потребі у подальшому можна окремі слайди супроводжувати рисунками чи фотографіями. У прикладі це виконано для 1, 6 та 7-го слайдів з використанням команди меню **Вставка / Рисунок / Из файла...**
14. На слайді 1 подаються базові визначення, які можна відразу доповнювати додатковою інформацією з інших слайдів, тому кожне з них можна оформити у вигляді гіперпосилання. Для створення гіперпосилання потрібно: виділити текст, що повинен бути гіперпосиланням; вибрати команду меню **Вставка / Гиперссылка**; у вікні діалогу **Добавление гиперссылки** у списку **Связать с:** вказати тип об'єкта і в центральній частині вікна вибрати потрібний об'єкт.
15. Організаційна діаграма, розташована на слайді 2, передбачає можливу деталізацію наведених понять при демонстрації презентації. Оскільки користувач може виявляти бажання переглядати слайди щоразу в іншій послідовності, то для зручності варто передбачити відповідні переходи від слайду 2 до одного з наступних (3, 4 чи 5-го), а також повернення до попереднього слайду. Варто передбачити також обхід деталізації, тобто перехід від діаграми до наступного за змістом матеріалу (слайд 6). Перехід між слайдами можна здійснити гіперпосиланнями у вигляді **кнопок**, які розміщуються на слайдах аналогічно об'єктам **Автофигуры** програми Word і налаштовуються таким чином: у контекстному меню вибирають команду **Настройка действия...**; у вікні діалогу встановлюють перемикач **Действие по щелчку мыши** на опцію **Перейти по гиперссылке**; нижче розгортають список можливих об'єктів та



Розглянута презентація наведена у додатку 3.1.

Контрольні запитання і завдання

1. Яке призначення програми PowerPoint?
2. Перелічіть команди головного меню програми PowerPoint.
3. Що являє собою презентація?
4. Що може містити слайд?
5. Як можна створити, коригувати, вилучити слайд?
6. Опишіть різні режими перегляду слайдів.
7. Як доповнити слайд новим об'єктом?
8. Для чого використовують гіперпосилання?
9. Для чого використовують ефекти анімації?
10. Як підготувати презентацію до демонстрації?





4.5. MS Access

Поняття бази даних

База даних (БД) (англ. *database*) – сукупність інформації, яка зв’язана між собою і зберігається у структурованому вигляді.

Приклади баз даних: телефонний довідник, інформація про працівників установи тощо.

Система баз даних (СБД) (англ. *database system*) – комп’ютеризована система опрацювання даних, основна мета якої – зберігати інформацію та видавати її за вимогою.

Системи баз даних поділяються на **однокористувацькі** (англ. *single-user system*) та **багатокористувацькі** (англ. *multi-user system*). Користувачу системи надається можливість виконувати багато різноманітних **операцій з даними**, зокрема, додавати нові дані чи змінювати існуючі, виконувати пошук даних та їх відбір за певним критерієм, вилучати дані.

Використання систем баз даних надає користувачам певні **переваги** порівняно з класичними методами їх зберігання та опрацювання: швидкість обробки даних, компактність зберігання, захист інформації, централізоване управління даними.

За способом встановлення зв’язків між даними бази даних поділяються на **ієрархічні, сіткові та реляційні**.

Ієрархічна база даних будується за принципом підпорядкування об’єктів. Один об’єкт є **головним**, а решта, які перебувають на нижчих рівнях ієрархії, - **підлеглими**. Між головним і підлеглими об’єктами встановлюється взаємозв’язок **“один до багатьох”**. Інакше кажучи, для даного головного об’єкта може існувати декілька підлеглих. Прикладом ієрархічної структури даних може бути організація каталогів на диску. Схема ієрархічної моделі даних показана на рис. 4.35 а.

У **сітковій моделі** даних поняття головного і підлеглих об’єктів дещо розширені. Один і той же об’єкт може одночасно виступати як головним, так і підлеглим елементом. Це означає, що кожен об’єкт може брати участь у довільній кількості зв’язків. Теоретично повинен існувати зв’язок кожного елемента даних з усіма іншими (зв’язок **“багато до багатьох”**). За схемою сіткової моделі організована, наприклад, структура автобусних маршрутів (рис. 4.35 б).



У **реляційній моделі даних** (рис. 4.35 в) об'єкти і зв'язки між ними подаються у вигляді прямокутних таблиць, які називають **відношеннями** (англ. relation).

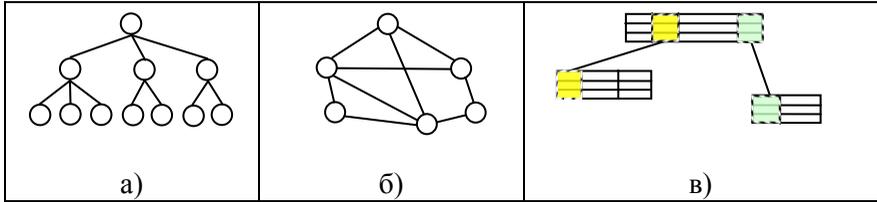


Рис. 4.35. Моделі даних: а) ієрархічна, б) сіткова, в) реляційна

Більшість баз даних для зберігання інформації використовують таблиці. Кожна таблиця у базі даних має унікальне ім'я. База даних – це сукупність таблиць, зв'язки між якими встановлюються за допомогою стовпців із даними однакового типу, проте немає вимоги, щоб ці стовпці мали однакові назви. Встановлення зв'язків між таблицями гарантує отримання правильних результатів при розробці складних форм і звітів. При наявності зв'язків програма автоматично вибирає зв'язані дані з таблиць у звіті, запити чи форми.

Для багатотабличних БД можна встановлювати **зв'язки таких типів**: “один до одного”, “один до багатьох”, “багато до багатьох”. У більшості випадків будь-які дві таблиці зв'язуються відношенням “один до багатьох”. Часом виникає необхідність розбити одну таблицю на кілька дрібніших, оскільки деякі відомості із неї використовуються не часто або деякі дані мають конфіденційний характер. Зв'язок між такими таблицями має вигляд “один до одного” (це означає, що один запис однієї таблиці зв'язаний лише з одним записом іншої). Між двома таблицями може бути заданий лише один зв'язок, тип якого можна при потребі змінити.

Реляційна модель найчастіше використовується при організації систем баз даних завдяки своїй простоті і природності подання інформації.

Рядки таблиці бази даних називають **записами** (англ. record), а стовпці - **полями** (англ. field).



Для роботи з базами даних використовують спеціальне програмне забезпечення.

Система керування базами даних (СКБД) - програмне забезпечення, яке необхідне для створення, використання та модифікації баз даних.

Будь-який пакет програм СКБД містить: середовище користувача, алгоритмічну мову програмування, транслятор, програми-утиліти. Нині налічується понад п'ятдесят різних СКБД. Серед них: Access, FoxPro, Paradox, Oracle, Informix та ін.

Однією з простих та популярних систем є СКБД Access, розроблена фірмою Microsoft. Access є першою програмою, яка підтримує технологію ODBC (Open DataBase Connectivity). Це дозволяє, використовуючи відповідні драйвери та фільтри, працювати із даними різних форматів, що відкриває доступ до БД інших прикладних програм.

Поняття бази даних Access

База даних в Access являє собою сукупність об'єктів таких типів: таблиць, запитів, форм, звітів, сторінок, макросів та модулів.

Access зберігає усі об'єкти бази даних в одному файлі з розширенням *.mdb. Склад бази, дату створення, розмір та ім'я власника (тобто її властивості) можна переглянути в окремому діалоговому вікні **Файл / Свойства бази даних**.

В залежності від потреб користувача, **об'єкти БД можуть створюватися:**

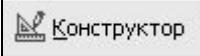
- у режимі конструктора;
- за допомогою програм-майстрів (у діалоговому режимі);
- на основі вже існуючих об'єктів.

Існують певні **вимоги до імен об'єктів**: довжина назви об'єкта не повинна перевищувати 64 символи; в імені об'єкта дозволяється використовувати пропуски (тільки не починати з них), але не можна вживати крапки, знаки оклику, квадратні дужки, символи * і ?.

Програма Access має **два режими роботи з об'єктами**:

- **конструктор** – перегляд структури об'єкта бази даних або її коригування;
- **перегляд** – відображення вмісту об'єкта бази даних.

Для перегляду або модифікації структури об'єкта можна активізувати його, клацнувши ЛКМ, та вибрати кнопку

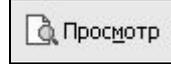


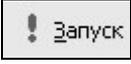
панелі інструментів вікна бази даних.

Щоб переглянути вміст об'єкта бази даних, можна у лівій частині вікна БД вибрати ту вкладку, яка містить перелік потрібних об'єктів, наприклад, **Таблицы** і двічі клацнути на об'єкті ЛКМ або виділити об'єкт і активізувати першу зліва кнопку панелі інструментів вікна бази даних. Залежно від типу об'єкта, вона буде мати вигляд



– для таблиць, запитів, форм та сторінок;



для звітів;  – для макросів.

Перед створенням бази даних важливо визначити мету її розробки та перелік завдань, які вона повинна виконувати. Тому перш за все необхідно продумати структуру та підібрати формати записів, а саме:

- визначити, яким чином дані будуть згруповані в таблиці;
- встановити **перелік полів** кожної таблиці;
- призначити **ключові поля** для кожної таблиці;
- визначити **схему зв'язків** між таблицями.

У таблицях баз даних потрібно зберігати лише **первинну** інформацію. Наприклад, у базі варто накопичувати лише відомості про кількість виробів і ціну, але зовсім необов'язково тримати інформацію про вартість замовлених товарів, оскільки її можна завжди обчислити у формі, запиті чи звіті.

Для аналізу даних використовують запити, які створюють **динамічні** (тимчасові) таблиці.

Запит призначений для відбору даних за встановленими критеріями.

Розрізняють два типи запитів: **QBE-запити** (англ. Query By Example – запит за зразком) і **SQL-запити** (англ. Structured Query Language – запити, створені за допомогою структурованої мови програмування). При роботі з QBE-запитами користувач визначає їх у вікні проектування, використовуючи при цьому підказки. При створенні SQL-запитів розробник повинен для формування опису використовувати спеціальні інструкції та функції.

QBE-запит програма Access легко трансліює у відповідний SQL-запит і навпаки. За допомогою SQL користувач може створити запити, набагато складніші від QBE-запитів.

QBE-запити називають також запитами на вибірку.

Доповнення записів та їх редагування рекомендується здійснювати через **форми**, а не безпосередньо в таблицях.

Форма – це бланк, який потрібно заповнити для занесення в таблицю нових даних або маска, яка накладається на набір даних для їх відображення.

Форма-бланк дозволяє спростити процес заповнення бази. За допомогою **форми-маски** можна обмежити обсяг інформації, доступної для користувача, тим самим забезпечити захист даних на певному рівні.

Звіт – засіб відображення підсумкових даних з таблиць та запитів у вигляді, зручному для перегляду, аналізу чи друку.

Сторінка доступу до даних – спеціальний тип Web-сторінки, призначеної для перегляду даних та роботи з ними в зовнішній або внутрішній мережі.

Макрос – засіб автоматизації процесу заповнення бази інформацією, відбору інформації тощо.

Модуль – написана мовою Visual Basic for Applications (VBA) процедура обробки подій чи виконання обчислень, призначена для виконання операцій, які неможливо реалізувати за допомогою команд чи макрокоманд Access.

Запуск програми Access здійснюється вибором команди у головному меню **Пуск / Програми/ Microsoft Access** або використанням відповідного ярлика (рис. 4.36).



Рис. 4.36. Ярлик для запуску Microsoft Access

Після запуску Access з'являється **головне вікно програми**, яке має будову, аналогічну до вікон інших програм пакету Microsoft Office. Воно містить заголовок, розмірні кнопки та кнопку виклику системного меню, головне меню програми, панелі інструментів (за

умови їх ввімкнення), а також робочу область, де можуть розташовуватися інші вікна (рис. 4.37).

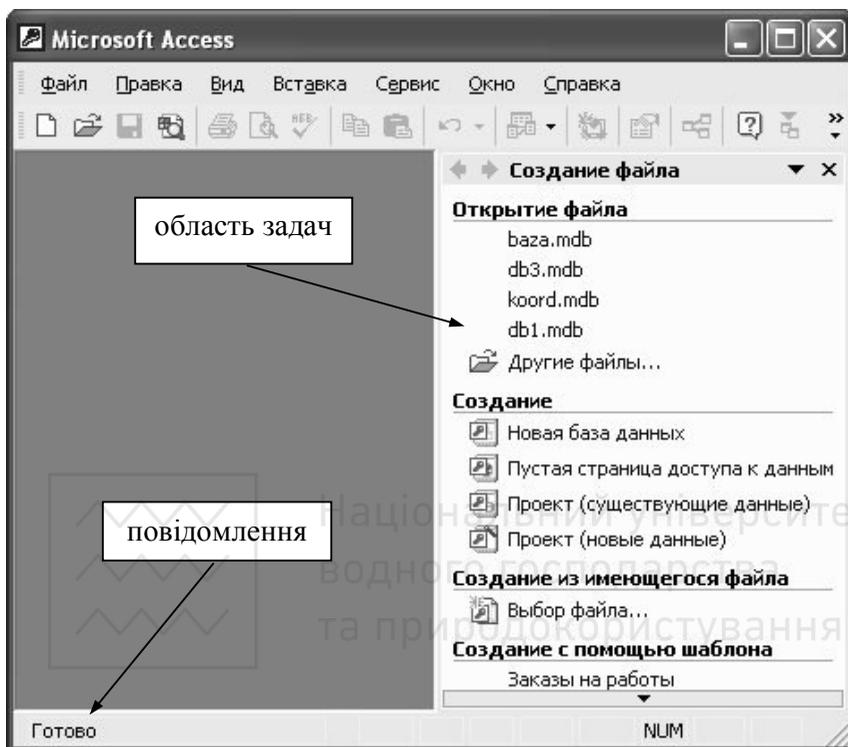


Рис. 4.37. Головне вікно програми Access

Відмітимо, що кількість та доступність пунктів меню залежить від режиму роботи Access. **Рядок стану** надає користувачеві інформацію про поточний режим роботи програми. Якщо система готова прийняти команду користувача, то у рядку стану зліва міститься повідомлення **Готово** (рис. 4.37).

У різних режимах роботи в рядку стану можуть з'являтися інші повідомлення. Наприклад: ФЛТР - включено фільтр; MOV - встановлено режим переміщення; EXT - встановлено режим розширення; ЗМІ - встановлено режим заміни при введенні даних.

Вказати спосіб розгортання списків команд для пунктів меню можна таким чином: **Сервис / Настройка...** / вкладка **Параметры** / відмітити потрібне. Наприклад, для опції **Эффект при выводе меню**



існують наступні варіанти: **По умолчанию, Развертывание, Случайный выбор, Соскальзывание, Угасание.**

Вмикання необхідних панелей інструментів Access виконується за допомогою команди меню **Вид / Панели інструментов / Настройка...** / вкладка **Панели інструментов**, де необхідно встановити відмітку біля потрібних назв.

У головному вікні Access може з'явитися **область задач при запуску** (зазвичай вона розташовується у правій частині робочої області вікна Access, тобто так, як показано на рис. 4.37). Область задач дозволяє: відкрити існуючу базу; створити нову БД, сторінку доступу до даних або проект; створити нову базу на основі однієї з існуючих або створити БД з використанням набору шаблонів. Для налаштування вмикання області задач при запуску Access потрібно у **Сервис / Параметры...** розгорнути вкладку **Вид** та вказати опцію **Область задач при запуске.**

Для створення нової бази необхідно в області задач у групі **Создание** вибрати ярлик **Новая база данных**. У вікні **Файл новой базы данных** вказати диск, папку і назву файлу для зберігання даних та клацнути кнопку **Создать**. Відразу після цього у робочій зоні з'явиться вікно бази даних (рис. 4.38).

Для створення нової БД за відсутності у вікні Access області задач виконують команду **Файл / Создать**. Після цього з'явиться **Область задач при запуске**. Всі наступні дії необхідно виконувати аналогічно попередньому опису.

Вікно бази даних дозволяє отримувати доступ до всіх об'єктів бази та вибирати режим роботи з ними. Це вікно, окрім стандартних компонент, має певну кількість ввімкнутих власних **панелей інструментів, панель об'єктів бази та панель групування**.

Розробку нової бази слід починати із **таблиць**, які у програмі Access можна створювати **одним із 5 способів**:

- у режимі таблиці;
- у режимі конструктора;
- за допомогою майстра таблиць;
- шляхом імпорту готової таблиці;
- за допомогою встановлення зв'язку з іншою БД.

Будь-який із перших трьох способів наведеного списку можна реалізувати, використавши один із **ярликів**, що розміщені, як правило, у верхній частині вікна бази даних (рис. 4.38).

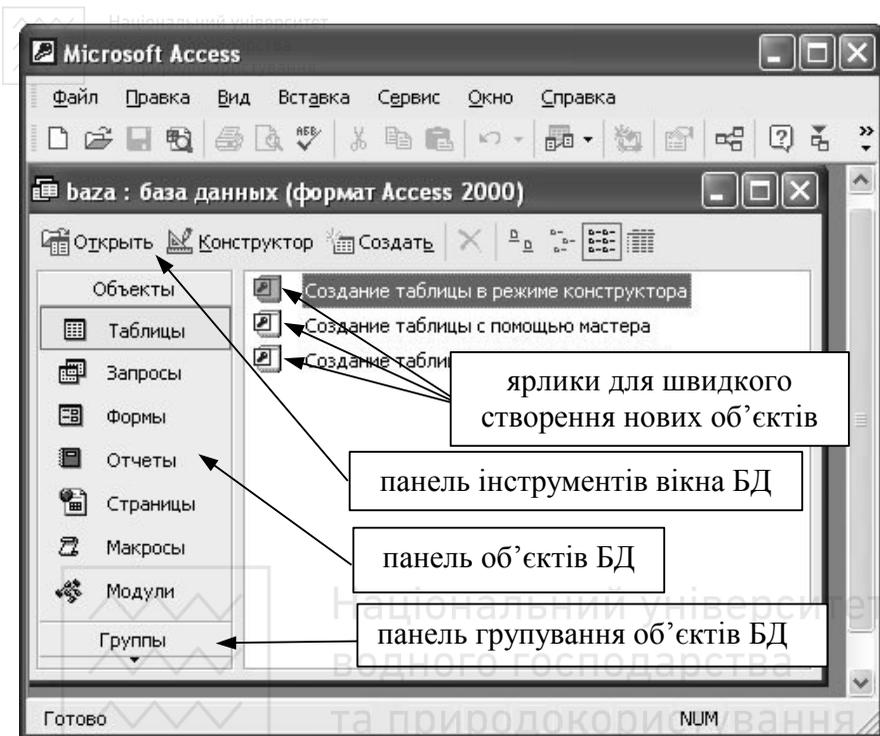


Рис. 4.38. Головне вікно програми Access з вікном нової бази даних

Якщо ярлики для швидкого створення нових об'єктів у вікні БД відсутні, їх можна ввімкнути через меню: **Сервіс / Параметри** / вкладка **Вид** / опція **Новые ярлыки объектов**.

Поля таблиць Access

Кожне поле запису має *назву (ім'я)* і характеризується *типом даних та певними властивостями*.

Тип даних вказується для кожного поля при його визначенні. Перелік типів даних наведений у таблиці 4.9. За замовчуванням встановлюється **Текстовий** тип даних.

Зауваження: Змінити тип поля, який встановлюється за замовчуванням, можна в діалоговому вікні **Параметри / Режим таблиць**.



Типи даних Access

Тип даних	Опис вмісту поля
Текстовый	Текст обсягом не більше ніж 255 символів
Поле МЕМО	Текст обсягом не більше ніж 65 535 символів
Числовой	Числове значення, діапазон якого визначається властивістю Размер поля
Дата / время	Значення дати та часу (8 байтів)
Денежный	Число, яке містить 15 розрядів зліва від десяткової крапки та 4 розряди справа від неї (8 байтів)
Счетчик	Унікальне значення (автоматично змінюється, коли до таблиці додають новий запис)
Логический	Значення Істина (1) або Хибність (0)
Поле объекта OLE	Документ, таблиця, графік, малюнок, звуковий сигнал, відео- та інша інформація, створена Windows-додатками, що підтримують технологію OLE
Гиперссылка	Шлях до файлу на диску або адреса в зовнішній чи внутрішній мережі

Тип даних визначає множину допустимих значень та перелік операцій, що можуть бути застосовані до них.

У режимі конструктора для будь-якого поля можна вказувати **опис**. В цьому випадку текст відповідного повідомлення з'являтиметься у рядку стану при роботі з даними поля.

Опис поля – пояснення у вигляді тексту, яке надає користувачеві додаткову інформацію про поле.

Кожне поле таблиці має **властивості**, кількість яких залежить від типу даних. Властивості полів дозволяють визначати: максимально можливу довжину текстового або діапазон значень числового поля (**Размер поля**), формат для відображення даних (**Формат поля**),



шаблон при введенні даних (**Маска вводу**) тощо. Деякі властивості призначені для полегшення роботи користувача при заповненні таблиці даними. До них належать значення для підстановки у поле за замовчуванням при додаванні нових записів (**Значення по умовчанию**) та **Підстановка**. Частина властивостей дозволяє уникати помилок при введенні даних в таблицю. Серед них: умова для визначення множини допустимих значень поля (**Умовне на значення**), обов'язковість заповнення поля значеннями (**Обязательное поле**), допустимість порожніх рядків (**Пустые строки**). Властивість **Підпись** спрощує надалі розробку форм та звітів, оскільки передбачає автоматичне виведення текстових повідомлень, що супроводжують відповідні поля.

Необхідною умовою для роботи з БД є однозначна ідентифікація кожного запису.

***Ключовий елемент даних або ключ** (англ. key) - це елемент запису, за яким можна однозначно встановити значення інших елементів цього запису.*

***Первинний ключ** (англ. primary key) – поле або комбінація полів, які однозначно ідентифікують запис в таблиці.*

За замовчуванням ключові поля у списку даних таблиці відображаються першими. В якості ключових можна використовувати лише ті поля, дані в яких ніколи не будуть повторюватися. Найчастіше ключовими встановлюють номери, шифри або коди клієнтів, рахунків, накладних тощо. **Щоб поле стало ключовим**, його потрібно зробити активним (або виділити, якщо це сукупність полів) в режимі конструктора таблиці і клацнути інструмент  **Ключевое поле** на панелі інструментів вікна Access. Відміна призначення поля ключовим здійснюється аналогічно.

Якщо користувач при створенні таблиці БД не встановить первинний ключ самостійно, то MS Access запропонує ввести і зробити ключовим додаткове поле з типом даних **Счетчик**, значеннями якого можуть бути порядкові номери записів.

***Індексованими полями** називають поля, в яких інформація впорядкована за певною ознакою.*

Індекси використовуються також для збільшення швидкості виконання багатьох операцій з даними бази.



Створення таблиць БД

Таблиці даних можна створювати одним із таких способів.

- 1. За допомогою майстра.** Перейти на вкладку **Таблицы**, клацнути ярлик **Создание таблицы с помощью мастера**, вибрати категорію таблиць (**Деловые** або **Личные**), вказати послідовно **зразки таблиць** (наприклад, **Сотрудники**), кожного разу **формуючи список полів** нової таблиці на основі зразків полів. В процесі складання списку полів нової таблиці використовують кнопки:  - вибір поточного поля,  - вибір всіх полів,  - відміна вибору поточного поля,  - відміна вибору всіх полів. Якщо двічі клацнути ЛКМ у списку зразків полів, буде здійснено вибір поля. Для відміни вибору можна виконати те саме у списку вибраних полів. Кнопка **Переименовать поле...** дозволяє змінювати назву поля. Після переходу на наступний крок кнопкою  стає можливим **вибір імені таблиці** (за замовчуванням відповідає зразку) та **способу визначення ключа** (автоматично чи самостійно). Ще один крок дозволить **встановити зв'язок** таблиці, яка формується, з існуючими таблицями БД. Останній крок дає змогу вибрати **варіант завершення роботи з майстром**: **Изменить структуру таблицы**, **Ввести данные непосредственно в таблицу** або **Ввести данные в таблицу с помощью формы, создаваемой мастером**.
- 2. В режимі конструктора.** Перейти на вкладку **Таблицы**, клацнути ярлик **Создание таблицы с помощью конструктора**. Після цього з'явиться вікно конструктора таблиць, яке розділене горизонтально на дві частини. У верхній частині потрібно вказати **назви усіх полів**, підбираючи для кожного поля **тип даних** із запропонованого списку та формуючи, якщо потрібно, **опис**. У нижній частині вікна розташовані **властивості активного поля**. Усі властивості подаються на двох вкладках: **Общие** і **Подстановка**.
- 3. Шляхом введення даних.** Перейти на вкладку **Таблицы**, клацнути ярлик **Создание таблицы путем ввода данных**, після чого з'явиться вікно таблиці. Для вибору назви поля потрібно клацнути на відповідному стовпці двічі ЛКМ (при цьому заготовка імені поля, яка спочатку має вигляд **Поле1**, буде виділена контрастним кольором). Типи даних для різних полів таблиці будуть підбиратися автоматично шляхом аналізу даних,



що вводяться. Цей спосіб створення таблиць зручний для використання користувачами-початківцями або як засіб пришвидшення роботи.

Приклад проектування структури таблиць

Процес створення таблиці починається із стадії її проектування, коли розробник бази даних аналізує дані, які необхідно опрацювати. Проаналізуємо діяльність підприємств-користувачів водних ресурсів, кожне з яких забирає чисту та скидає забруднену воду у деяку річку. Заміри обсягів скинутої води здійснюються періодично. Цю інформацію у реляційних СКБД необхідно подати сукупністю двовимірних таблиць. Нехай база даних, що створюється, називається **база** та спочатку складається з однієї таблиці **СкидиВоди**. Визначимо структуру таблиці, в якій розміщена така інформація:

- повна назва підприємства;
- місце розташування (адреса) підприємства;
- дата скиду води;
- назва водного об'єкта;
- обсяг скинутої забрудненої води.

Кожен описаний вище пункт таблиці **СкидиВоди** – це майбутнє поле таблиці. Визначимо тепер назву, тип даних, розмір та необхідну множину значень цих полів і зведемо все у таблицю 4.10.

Як бачимо, деякі клітини цієї таблиці не заповнені. Зокрема, розмір поля з типом **Дата** визначається автоматично, а можливі значення даних для полів **Дата**, **НаселенийПункт** та **Об'ємСкиду** таблиці скидів можуть бути різноманітними.

Для уникнення зайвих повторів даних, які можуть бути причиною помилок при введенні даних та нераціонального використання дискового простору комп'ютера, проектування таблиць завершується їх **нормалізацією** – процесом зменшення



Поля таблиці СкидиВоди

Назва поля	Тип даних	Розмір, байт	Множина значень
НазваПідприємства	Текстовий	16	Перелік підприємств
НаселенийПункт	Текстовий	16	
Дата	Дата		
НазваВодногоОб'єкта	Текстовий	16	Перелік річок
Об'ємСкиду	Числовий	4	

надлишкової інформації. Згідно із принципами нормалізації, таблицю **СкидиВоди** слід редагувати, замінивши в ній поле **Назва-**

Підприємства на **КодПідприємства** (тип даних **Текстовий**, розмір **5**), а крім цього ввести ще нову таблицю **СписокПідприємств** з полями **КодПідприємства**, **НазваПідприємства**, **НазваВодногоОб'єкта** та **НаселенийПункт**, які повністю відповідають аналогічним полям таблиці **СкидиВоди**. Структура таблиці **СкидиВоди** модифікується наступним чином: всі поля, що стосуються підприємства і визначаються за його кодом, крім самого коду (а це однакові на даний момент часу поля таблиць **СкидиВоди** і **СписокПідприємств**), у таблиці вилучаються.

Аналогічно здійснюється поділ таблиці **СписокПідприємств** на дві таблиці: **СписокПідприємств** та **ПерелікВоднихОб'єктів**. У таблиці **СписокПідприємств** поле **НазваВодногоОб'єкта** замінюється на **КодВодногоОб'єкта**. Новостворена таблиця **ПерелікВоднихОб'єктів** містить поля **КодВодногоОб'єкта** та **НазваВодногоОб'єкта**.

Таблиці **СписокПідприємств** та **ПерелікВоднихОб'єктів** відіграють роль **таблиць-довідників**.

У загальному випадку таблиці створюються таким чином, щоб задовольнити трьом **нормальним формам**.

Перша нормальна форма (1НФ)

- Таблиця не має записів, що повторюються.
- У таблиці відсутні групи полів, що повторюються.



- Рядки повинні бути не впорядковані.
- Стовпці повинні бути не впорядковані.

Зрозуміло, щоб задовольнити умові 1НФ, кожна таблиця повинна мати первинний ключ.

Друга нормальна форма (2НФ)

- Таблиця задовольняє умовам першої нормальної форми.
- Будь-яке неключове поле однозначно ідентифікується повним набором ключових полів.

Очевидно, що таблиці, які мають один первинний ключ, автоматично задовольняють умові 2НФ.

Третя нормальна форма (3НФ)

- Таблиця задовольняє умовам другої нормальної форми.
- Жодне з неключових полів таблиці не ідентифікується за допомогою іншого неключового поля.

Процес нормалізації таблиць, як правило, супроводжується створенням окремих додаткових таблиць, які зв'язані між собою з використанням введених первинних ключів. Повертаючись до таблиці **СкидиВоди** відмітимо, що її первинним ключем буде сукупність полів **Дата** та **КодПідприємства** (це пов'язано з тим, що скиди одних і тих же підприємств реєструються періодично, але жодне з підприємств за логікою завдання не буде реєструватися кілька разів протягом одного дня). Очевидно первинним ключем таблиці **ПерелікВоднихОб'єктів** буде поле **Код ВодногоОб'єкта**, а первинним ключем таблиці **СписокПідприємств** – поле **КодПідприємства**. Тепер настає черга побудови таблиць (знак \Rightarrow означає первинний ключ).

Реалізувати структуру спроектованих таблиць найзручніше в режимі конструктора. При цьому слід дотримуватися сформульованих вимог до імен об'єктів Access.

Таблиця 4.11.

Структура таблиці СкидиВоди

Назва поля	Тип даних	Розмір, байт	Множина значень
\Rightarrow Дата	Дата		



↔ КодПідприємства	Текстовий	5	
Об'ємСкиду	Числовий	4	

Таблиця 4.12.

Структура таблиці СписокПідприємств

Назва поля	Тип даних	Розмір, байт	Множина значень
↔ КодПідприємства	Текстовий	5	
КодВодногоОб'єкта	Текстовий	5	
Назва	Текстовий	20	Перелік підприємств
НаселенийПункт	Текстовий	20	

Таблиця 4.13.

Структура таблиці ПерелікВоднихОб'єктів

Назва поля	Тип даних	Розмір, байт	Множина значень
↔ КодВодногоОб'єкта	Текстовий	5	
Назва	Текстовий	20	Перелік річок

Підбір типів даних та описи можна виконати, орієнтуючись на зразки, подані на рис. 4.39, 4.40 і 4.41. Розміри полів необхідно вказувати у нижній частині вікна конструктора згідно з проектами, наведеними у таблицях 4.11, 4.12 і 4.13.

При введенні нових даних або коригуванні існуючих можна здійснювати **автоматичний вибір нового значення поля із списку**, що пропонується.

Масив підстановки (список значень для вибору) може бути створений користувачем з фіксованих значень або із значень таблиці чи запиту. Для реалізації цього інколи використовують спеціальну програму, яка викликається у вікні конструктора при виборі типу поля **Мастер подстановок**.

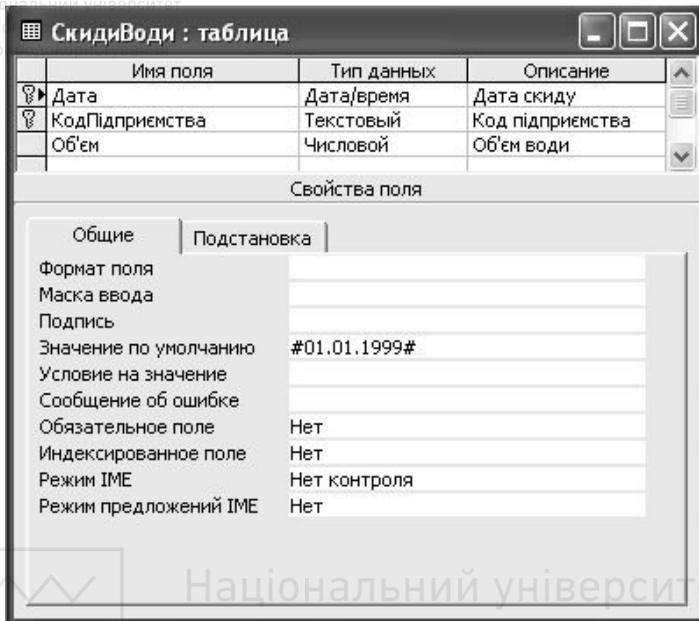


Рис. 4.39. Таблица СкидиВоди в режиме Конструктор

У вікні діалогу **Создание подстановки** потрібно буде на першому кроці вибрати одну з опцій: **Объект “столбец подстановки” будет использовать значения из таблицы или запроса** чи **Будет введен фиксированный набор значений**.

Далі після вибору опції **Объект “столбец подстановки” будет использовать значения из таблицы или запроса** необхідно вказати **об’єкт** (таблицю чи запит) та **поля** (поле), що містять потрібні дані. Для поля також вказується підпис.

На тому ж кроці, але при виборі опції **Будет введен фиксированный набор значений**, після клацання **Далее>** необхідно сформулювати один або кілька стовпців з даними, які братимуть участь у заповненні таблиці.

На третьому кроці слід вказати стовпець із значеннями, що повинні бути збережені у таблиці бази при виборі рядка, а на четвертому - підпис для створеного поля, який буде відображатися при перегляді таблиці.

Підстановку значень можна реалізувати інакше. Розглянемо це більш детально на прикладі таблиць **СкидиВоди** і **Список-**

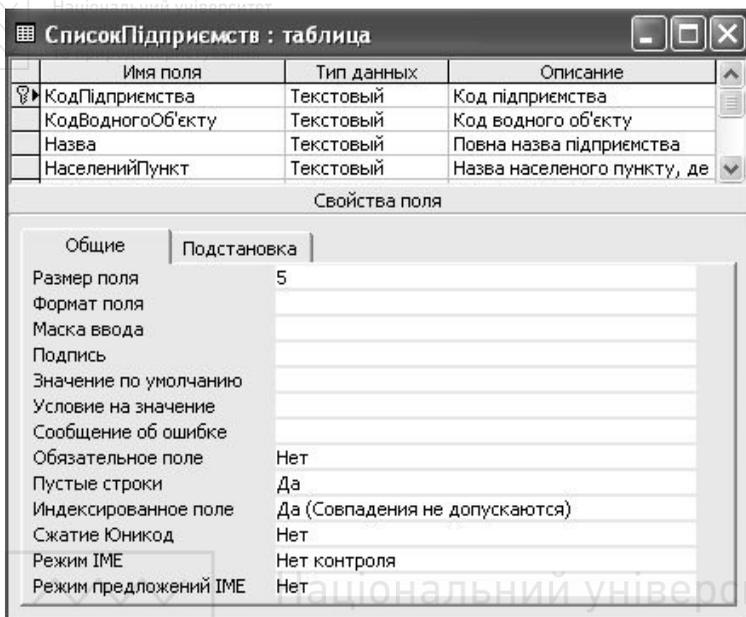


Рис. 4.40. Таблица СписокПідприємств в режиме Конструктор

Підприємств. Організуємо автоматичний вибір та підстановку даних **КодПідприємства** з таблиці **СписокПідприємств** у таблицю **СкидиВоди**. Для цього у властивості **Подстановка** поля **КодПідприємства** таблиці **СкидиВоди** потрібно вказати значення згідно рис. 4.42.

Значення параметрів підстановки **Тип элементов управления** і **Тип источника строк** зручно не набирати з клавіатури, а вибирати з коротких списків, які розгортаються при клацанні ЛКМ в кінці рядка поля введення.

Параметр **Источник строк** заповнимо командою:

SELECT СписокПідприємств.КодПідприємства, СписокПідприємств.
Назва, * **FROM** СписокПідприємств;

Команду можна набрати вручну з клавіатури або сформувати в режимі діалогу, що розпочинається з виклику **Построителя запросов** кнопкою , яку можна знайти, клацнувши ЛКМ в кінці рядка **Источник строк**. Діалог передбачає **вибір таблиці-джерела даних (СписокПідприємств)** та транспортування полів, за якими здійснюється вибір коду з таблиці-джерела (**КодПідприємства**,

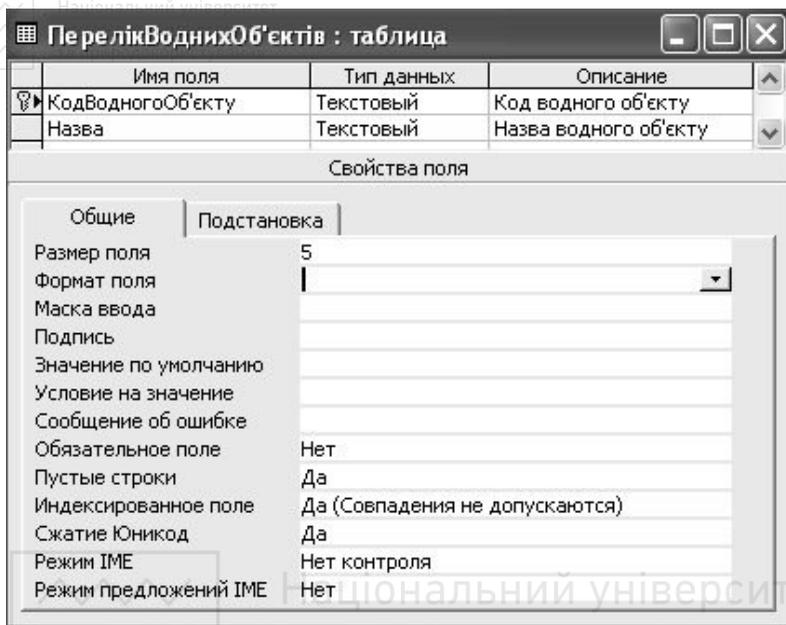


Рис. 4.41. Таблица ПерелікВоднихОб'єктів в режиме Конструктор (Назва) з верхньої частини вікна **Построителя запросов** у нижню таким чином, щоб їх назви з'явилися правіше надпису **Поле:**.

Присоединенный столбец повинен містити номер стовпця, з якого вибиратимуться дані для підстановки (наприклад, 1). Параметр **Число столбцов** керує кількістю полів, що будуть відображатися для вибору потрібного значення при підстановці (наприклад, 2).

Аналогічно можна забезпечити підстановку значень з таблиці **ПерелікВоднихОб'єктів** у таблицю **СписокПідприємств**.

Майстер підстановок у загальному випадку забезпечує вибір даних серед елементів **фіксованих значень** або серед **значень поля таблиці-довідника**. У розглянутому прикладі щоразу використовуються довідники. Процес вибору, як правило, посилюється візуалізацією інших полів таблиці-довідника (в даному випадку при виборі значень поля **Код...** висвічується ще й поле **Назва...**).



Приклад створення схеми даних

Готуючись до подальшої обробки інформації, можна для

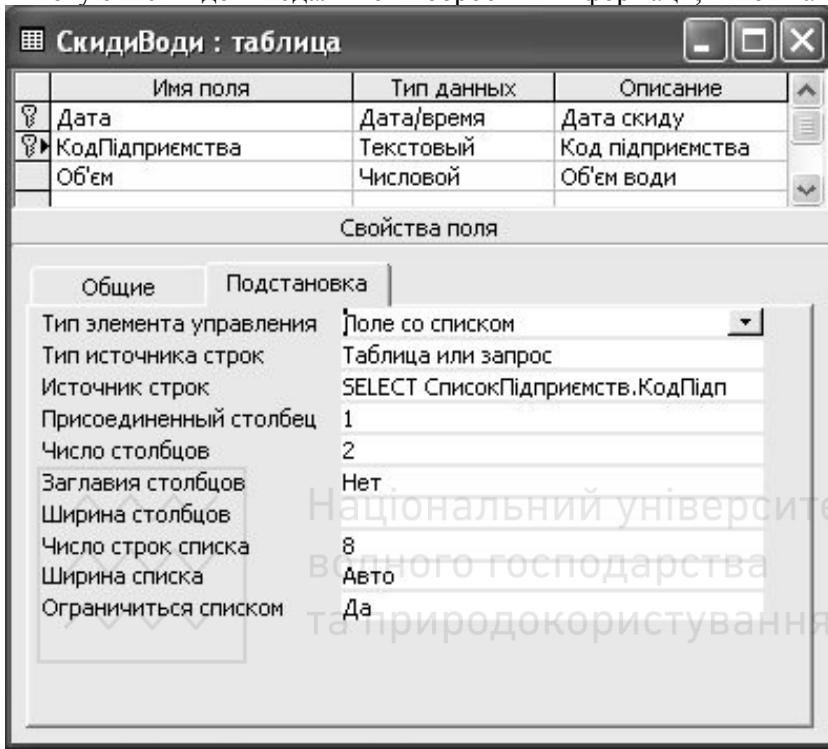


Рис. 4.42. Параметры влаивости Подстановка поля КодПідприємства
таблиці СкидиВоди

зр
учності створити схему даних, що міститиме опис зв'язків між
таблицями. Це виконується командою меню **Сервис / Схема даних**
або кнопкою  панелі інструментів Access. У вікні діалогу
Добавление таблицы необхідно виділити по черзі кожен з таблиць
списку та помістити їх у вікно **Схема даних** кнопкою **Добавить**.
Таблиці також можна додавати у вікно **Схема даних**, двічі
клацаючи ЛКМ на їх назвах. Коли вікно **Схема даних** буде містити
зображення кожної з таблиць БД, вікно **Добавление таблицы** можна
закрити. При потребі вилучення зайвих зображень таблиць виконують
клавішею **Delete**.

Зображення таблиць можна розтягнути, транспортуючи ЛКМ їх границі таким чином, щоб було видно усі наявні поля. Зв'язки встановлюються транспортуванням ЛКМ поля **КодВодногоОб'єкта** таблиці **ПерелікВоднихОб'єктів** до відповідного поля таблиці **СпискиПідприємств**, а також транспортуванням ЛКМ поля **КодПідприємства** таблиці **СпискиПідприємств** до відповідного поля таблиці **СкидиВоди**. Встановлені зв'язки відображаються на схемі у вигляді ліній. За логікою задачі обидва зв'язки будуть мати тип **“один до багатьох”**. Знищення зв'язку здійснюється клавішею **Delete** при виокремленій лінії зв'язку. Для зміни зв'язку клацають двічі ЛКМ на його лінії. У вікні діалогу **Изменение связей** можна також вмикати опцію **Обеспечение целостности данных**, а кнопка **Объединение** надає можливість змінювати параметри об'єднання записів у зведену таблицю. Схема даних подана на рис. 4.43.

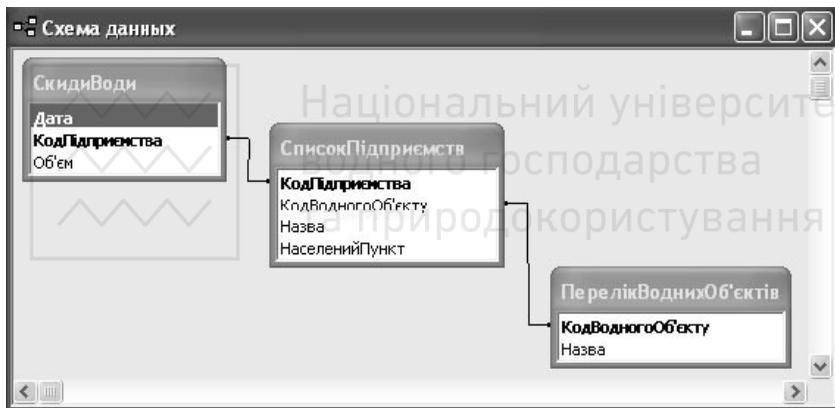


Рис. 4.43. Приклад схеми даних

Р

робота з даними в режимі таблиці

Для заповнення таблиць даними їх відкривають у режимі таблиці кнопкою **Открыть** вікна БД або двічі клацнувши ЛКМ на імені таблиці. Набір даних зручно виконувати з клавіатури, фіксуючи їх у полях за допомогою клавіші **Enter** або клавіш із стрілками.

Відмітимо, що при використанні в таблицях **МЕМО-полів**, для перегляду або редагування вмісту поля слід використовувати комбінацію клавіш **Shift+F2**, яка відкриває для роботи спеціальне вікно **Область вводу**.

В Access неможливо редагувати:



- обчислювані поля (віртуальні дані);
- поля в деяких типах запитів, зокрема, такі, які беруть участь у встановленні зв'язків або поля з підсумковими значеннями.

Зауваження: В Access процес вилучення записів незворотній.

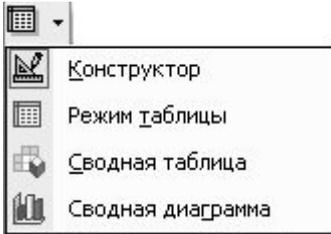


Рис.4.44. Меню кнопки Вид в режимі Конструктор

Для зміни режиму відображення таблиці можна використовувати кнопку **Вид** панелі інструментів вікна Access.

Вона з'являється в режимах **Конструктор і Режим таблиць**.

Вигляд кнопки змінюється залежно від поточного режиму (рис. 4.44).

Швидке переміщення по таблиці здійснюють використанням клавіш та їх комбінацій, зокрема: **Shift+Tab (Tab)** - на попереднє (наступне) поле, **Ctrl+Home (Ctrl+End)** - на перше поле першого запису (на останнє поле останнього запису).

Підбір оптимальної ширини стовпця виконується встановленням вказівника миші на правій межі його заголовка та клацанням двічі ЛКМ. Те саме може виконати команда **Формат / Ширина столбця** або відповідна команда контекстного меню виділеного стовпця.

Щоб перемістити поле на інше місце, його потрібно виділити і транспортувати (при натиснутій ЛКМ) так, щоб контрастна лінія між стовпцями розмістилася на новому місці.

Зафіксовані поля завжди відображаються на екрані в лівій частині таблиці і при переміщенні по ній вони не зміщуються. Для фіксації необхідно виділити потрібне поле, вибрати команду меню **Формат / Закрепить столбцы** або відповідну команду контекстного меню виділеного стовпця. Після виконання цієї команди зазначений стовпець переміститься в ліву частину таблиці і постійно залишатиметься видимим. Відмінити фіксацію дозволяє команда **Формат / Освободить все столбцы**. Після відміни фіксації повернення стовпця на старе місце не відбувається.

Приховування полів (наприклад, з конфіденційною інформацією) можна виконати командою **Формат / Скрыть столбцы**

або відповідною командою контекстного меню виділеного стовпця. Для відновлення відображення прихованих стовпців використовують команду **Формат / Отобразити столбцы**.

Зауваження: приховані поля не беруть участі у пошуку та опрацюванні даних.

Робота з даними передбачає можливість пошуку потрібної інформації за зразком у конкретному полі або в усій таблиці. Для

активізації пошуку потрібно натиснути кнопку  **Найти** панелі інструментів чи виконати команду меню **Правка / Найти** (або комбінацію клавіш **Ctrl+F**). У діалоговому вікні **Поиск и замена** на вкладці **Поиск** у полі **Образец:** потрібно ввести зміст шуканих даних. За замовчуванням у полі **Поиск в:** встановлюється назва поточного поля. Якщо потрібно здійснити пошук даних в усій таблиці, то в цьому полі потрібно вибрати назву таблиці. За допомогою списку **Совпадение:** встановлюють, яку частину поля бази даних повинен складати шуканий текст (**С любой частью поля, Поля целиком, С начала поля**).

Інколи потрібно не просто знайти певну інформацію в таблиці, а замінити її на іншу. Для цього використовують команду **Правка / Заменить** (або комбінацію клавіш **Ctrl+H**). Після того як відкриється діалогове вікно **Поиск и замена**, де активною буде вкладка з додатковим полем **Заменить на:**, вказують текст заміни.

Зауваження: параметри пошуку, які використовуються за замовчуванням, можна змінити у пункті **Параметры / Правка и поиск**.

Інформацію у таблиці можна сортувати за одним чи декількома полями. Спочатку слід встановити мову, якою ведеться БД і за алфавітом якої виконуватиметься впорядкування. Для цього виконуємо команду **Сервис / Параметры / Общие** і в полі **Порядок сортировки базы данных:** вибираємо мову. **При простому сортуванні** (за одним полем) необхідно відкрити таблицю, зробити активним потрібне поле, вибрати команду меню **Записи / Сортировка** та визначити порядок сортування чи виконати те саме з

використанням кнопок  панелі інструментів чи відповідних команд контекстного меню виділеного стовпця.

Зауваження: сортувати за декількома полями можна за допомогою розширеного фільтра чи запиту.

Програма Access дозволяє відобразити дані зв'язаних таблиць у вікні головної таблиці. Переглядаючи головну таблицю в режимі таблиці можна побачити, що між ділянкою виділення і першим стовпцем таблиці є вузький стовпець, у якому навпроти кожного запису розміщений знак . Клацання на ньому ЛКМ дозволяє відобразити частину зв'язаної з ним таблиці, де розміщені лише ті записи, які зв'язані з даним записом головної таблиці. При цьому знак  буде замінено на .

Параметри шрифту записів таблиці можна задати лише для усієї таблиці, а не для окремих записів чи полів. Для цього використовують команду **Формат / Шрифт**.

Для зміни параметрів таблиці БД її потрібно відкрити в режимі таблиці та виконати команду **Формат / Режим таблиць**. Потім можна вибрати потрібні варіанти форматування (оформлення, лінії сітки, колір ліній, фон комірок тощо).

Параметри перегляду усіх таблиць БД на екрані встановлюють командою **Сервіс / Параметри / Режим таблиць**.

При друкуванні таблиць баз даних у програмі Access слід враховувати наступні особливості. Параметри, які задано у вікні діалогу **Параметри сторінки** на вкладці **Поля**, поширюються лише на активний документ. Програма Access зберігає інформацію про поля сторінок лише для форм і звітів. Параметри сторінок таблиць, запитів та модулів потрібно задавати щоразу при виведенні на друк. Якщо для друкування форм чи звітів використовуються готові бланки, у вікні **Печать** клацають кнопку **Свойства** і у новому вікні на вкладці **Бумага** заповнюють список **Тип бумаги** варіантом **Печатные бланки**. Якщо потрібно друкувати лише деякі записи таблиці, то їх **виділяють**, після чого вибирають команду **Файл / Печать** і у діалоговому вікні **Печать** задають опцію **Выделенные записи**.

Зауваження: параметри друкування, які використовуються за замовчуванням, можна змінити командою **Сервіс / Параметри... / Общие**.

Для копіювання таблиці потрібно на вкладці **Таблицы** виділити значок потрібного об'єкта і виконати команду **Правка / Копировать**. Після цього виконати **Правка / Вставить** і у вікні діалогу **Вставка таблицы** вибрати спосіб копіювання (**только структура** або **структура и данные** або **добавление данных в таблицу**) та вказати назву нової таблиці.



Для групування об'єктів у папці **Избранное** потрібний об'єкт БД перетягують мишкою у цю папку або використовують команду **Добавить в группу** з контекстного меню виділеного об'єкта. При цьому у папку поміщається тільки ярлик об'єкта, а сам об'єкт залишається на попередньому місці.

Створення форм

Якщо інформація поновлюється нечасто або для її перегляду потрібно мати повний огляд записів масиву даних, варто використовувати **таблиці**. Якщо ж дані часто поновлюються, а при їх введенні потрібно зосередити увагу на окремому записі, доцільно використовувати **форми**.

У програмі Access можна створювати **прості** або **складені** форми, форми з **діаграмами** чи із **зведеними таблицями**, а також **кнопкові** форми та **заставки**.

Будь-яка форма для роботи з даними (крім кнопкових та заставок) будується на основі **таблиці** або **запиту**. Якщо форма використовує дані кількох таблиць, її бажано створювати на основі **запиту**. **Кнопкові форми** використовуються для вибору інших форм, попереднього перегляду чи друкування звітів, тобто служать засобами інтерфейсу для автоматизації роботи з БД. **Форми-заставки** також сприяють покращенню інтерфейсу.

Існують наступні режими відображення форм: **Конструктор**, **Режим форми**, **Режим таблиць**.

Конструктор форм містить елементи, використання яких дає змогу проектувати форми, а саме: вікно проекту форми, вікно джерела даних, панель елементів керування, вікно властивостей об'єкта.

За замовчуванням у **вікні проекту форми** (рис. 4.45) розміщені **горизонтальна та вертикальна лінійки**. На вертикальній лінійці містяться маркери областей проекту. Весь проект форми розмічений **лініями сітки**, які полегшують форматування окремих

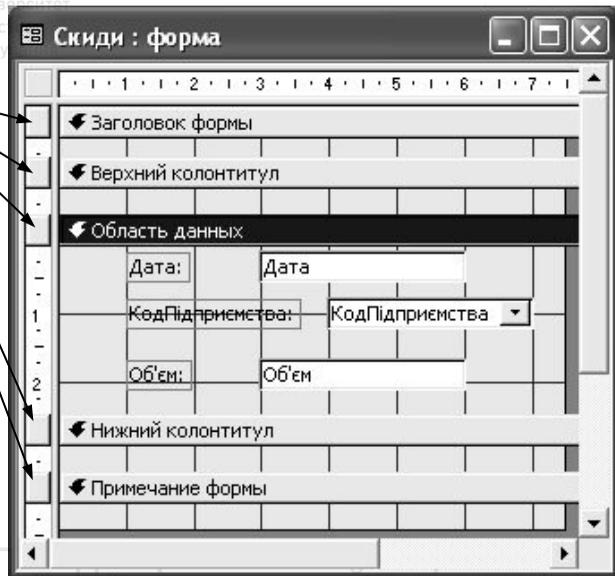


Рис. 4.45. Вікно проекту форми

об'єктів. **Вмикати** або **вимикати** зображення сітки чи лінійки можна за допомогою меню **Вид**.

Вікно проекту форми може містити такі області: **заголовок** (наприклад, назва фірми), **примітку** (інструкції для роботи з формою), **верхній та нижній колонтитули** (назви стовпців на початку кожної сторінки або номер сторінки) та **область даних**. Якщо будь-яка із областей у проекті форми відсутня, ввімкнути її можна через меню **Вид** / назва області.

Область даних - обов'язкова частина форми. В ній розміщуються **елементи керування**, за допомогою яких відображаються дані базових об'єктів, **надписи**, **підсумкові поля**, **списки значень**. Елементи керування можуть бути **зв'язаними**, **вільними** чи **обчислювальними**.

Зв'язані елементи приєднані до полів базової таблиці або запиту. Вони використовуються для відображення, введення чи виведення даних, а також для оновлення значень поля БД. **Вільні** елементи керування використовуються для виведення на екран ліній, малюнків, написів тощо. **Обчислювальні** елементи керування в якості джерела даних використовують вирази, до яких можуть входити дані з базової таблиці чи запиту, а також дані іншого елемента керування. Елементи

керування набувають свого реального вигляду лише в режимі заповнення чи перегляду. У режимі конструктора вони подаються у вигляді умовних позначень, які дуже схожі на реальні.

Вікно джерела даних (рис.4.46), яке вмикається командою **Вид / Список полей**, використовують для швидкого вибору полів при конструюванні форм із даними таблиць.

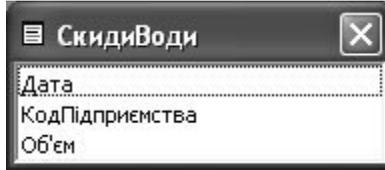


Рис. 4.46. Вікно джерела даних

Форма може містити **елементи керування** - графічні об'єкти, що використовуються для відображення даних, поновлення та перегляду інформації. Для розміщення кожного із елементів на макеті майбутньої форми використовують значки елементів керування, які розташовані на спеціальній панелі (рис. 4.47). Якщо панель відсутня, її можна ввімкнути кнопкою  **Панель елементів**.

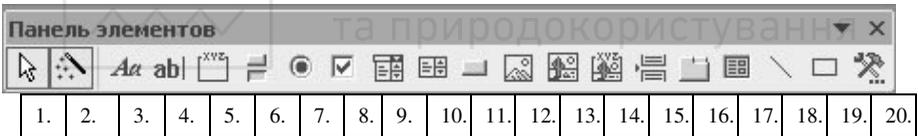


Рис 4.47. Панель елементів

Для виведення чи введення даних найчастіше використовують елемент керування **4.Поле**, а для створення заголовків або пояснювального тексту – **3.Надпись**. При проектуванні кнопочких форм звертаються до елемента **11.Кнопка**.

Після побудови форми виникають різноманітні причини її коригування: покращення дизайну, підвищення ефективності роботи з формою тощо. Редагуванню форми допомагають вікна властивостей самої форми та її окремих елементів. Набір властивостей залежить від вибраного елемента.

Для відображення властивостей об'єкта в окремому вікні потрібно активувати цей об'єкт та вибрати команду **Вид / Свойства** або команду **Свойства** контекстного меню.

Вікно властивостей (рис.4.48) має багатосторінкову структуру, в якій згруповані характеристики за подібними ознаками. У вікні розміщені вкладки: **Макет, Данные, События, Другие, Все**.

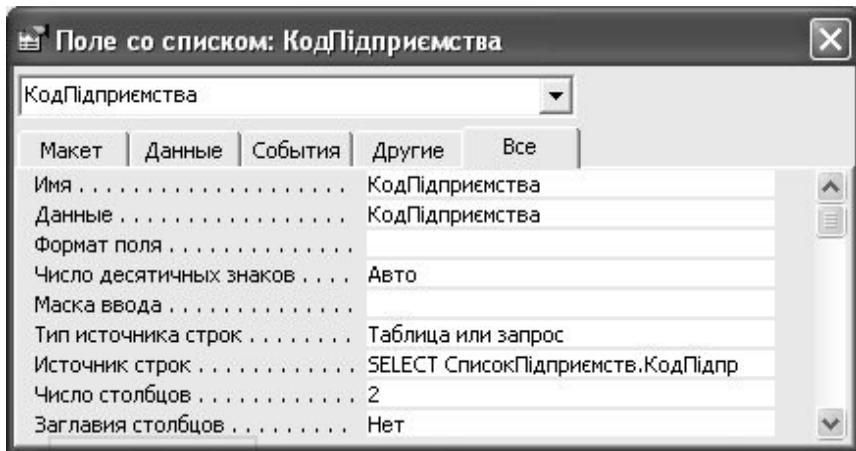


Рис. 4.48. Вікно властивостей об'єкта форми поле КодПідприємства

Для створення форми в режимі **Конструктор** потрібно перейти на вкладку **Формы**, клацнути кнопку **Создать** вікна БД, у вікні **Новая форма** вибрати **Конструктор**, вказати **таблицю** – джерело даних (наприклад, **СкидиВоди**), клацнути **ОК**. Потім необхідно транспортувати поля з вікна джерела даних в область даних вікна конструктора форми. Щоб підібрати розміри окремих елементів форми, використовують маркери. При потребі переміщують об'єкт, використовуючи маркер верхнього лівого кута. Після цього форму можна закрити, вказавши для неї **ім'я** (**Скиди**). Результат розміщено на рис. 4.49.

Програма Access дозволяє створювати форми на основі існуючих шаблонів – **автоформ**. Використовуються автоформи таких типів: **в столбец, ленточная, табличная, сводная таблица і сводная диаграмма**. Для створення автоформи у вікні БД переходимо на

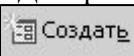
вкладку **Формы**, після чого клацаємо інструмент  **Создать**. У вікні діалогу вибираємо довільний **зразок** автоформи (наприклад, **Автоформа: в столбец**), у списку, що розгортається, виділяємо **таблицю** (**СписокПідприємств**). Форма відразу стане активною. Її



Рис. 4.49. Форма Скиди

назву вказуємо при закритті вікна форми. Результат описаних дій подано на рис. 4.50.

Рис. 4.50. Форма СписокПідприємств

Форма з діаграмою створюється для графічної ілюстрації числових значень одного чи декількох полів таблиці. Перед початком побудови діаграми потрібно вибрати поля таблиці чи форми, значення яких будуть відображатися графічно. За замовчуванням значення стовпця утворюють на діаграмі **ряд даних**, а значення рядка – **категорію**. Діаграма може складатися з декількох рядів чи категорій. У **легенді** можна подати пояснення до позначень.

Access дозволяє створювати такі **типи діаграм**:

- гістограми;
- лінійчасті;
- з областями;
- кругові.

Для створення форми з діаграмою можна у вікні БД перейти на

Национальный университет
для природокористування

вкладку **Форми**, клацнути кнопку **Создать** і у вікні діалогу **Новая форма** вибрати елемент **Диаграмма**, вказавши **таблицю чи запит** – основу майбутньої діаграми (наприклад, **СкидиВоди**). Далі майстер запропонує **сформуванати список полів**, що містять дані для побудови діаграми. На наступному кроці потрібно **вибрати тип** діаграми (наприклад, **Круговая**). Після цього необхідно **організувати розташування полів** у відповідних областях діаграми, транспортуючи мишею кнопки з назвами полів з правої частини вікна у відповідні області діаграми (рис. 4.51).

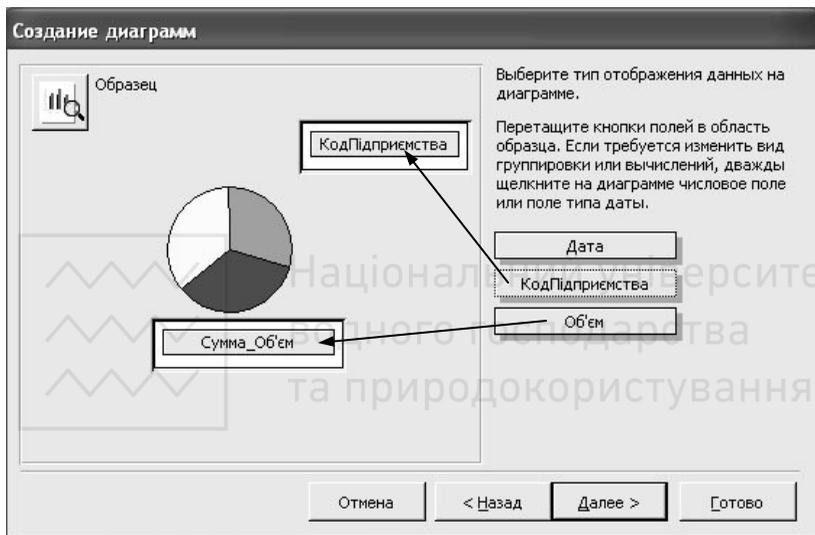


Рис. 4.51. Створення форми з діаграмою

При транспортуванні назви поля в область **Данные** перед іменем поля з'явиться **назва функції**, що буде використана при проведенні обчислень. За замовчуванням це функція сумування. Для вибору іншої функції потрібно клацнути двічі ЛКМ на назві поля в області **Данные**.

У області **Данные** можуть знаходитися декілька обчислювальних полів, інформація з яких опрацьовується з використанням різних функцій. Якщо область **Ряды** залишити незаповненою, то підписами у діаграмі будуть назви обчислювальних полів. Кнопка **Образец** дозволяє переглядати результат побудови діаграми. На останньому кроці роботи з майстром слід вказати **назву** діаграми, передбачити



виведення **легенди**, та вибрати **спосіб завершення роботи з майстром**.

Для оформлення діаграм краще використовувати програму **Microsoft Graph**, яка завантажується після клацання двічі ЛКМ на діаграмі у режимі **Конструктор**. Наприклад, для виведення значень сум скидів на діаграмі (рис.4.52) можна вибрати **Диаграмма / Параметры диаграммы / Подписи данных / значения**.

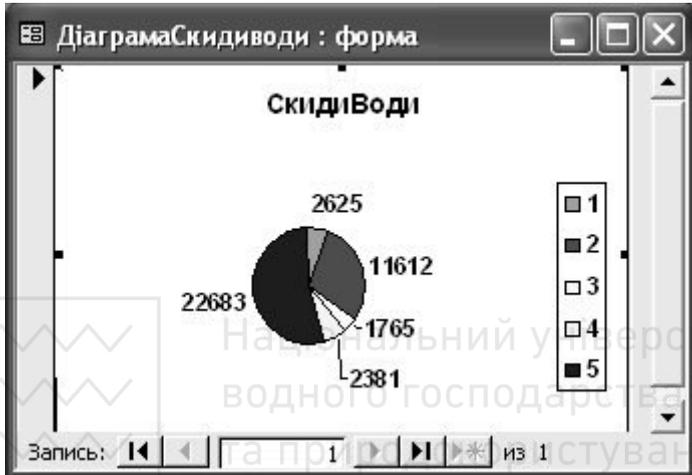


Рис. 4.52. Форма з діаграмою Скиди Води

Форма із зведеною таблицею дозволяє продемонструвати підсумкові дані, які отримані за допомогою майстра зведених таблиць програми Microsoft Excel. Така форма – це інтерактивна таблиця, яка виконує задані користувачем обчислення в залежності від способу розміщення даних у зведеній таблиці. Наприклад, значення поля у зведеній формі можуть відображатися вертикально або горизонтально, підсумкові значення при цьому обчислюються за рядками чи за стовпцями. Значення поля також можуть використовуватися у ролі заголовків рядків чи стовпців, при цьому зведена форма обчислюватиме окремі значення для кожної комірки, а потім проміжні та загальні підсумки.

Відкриваючи зведену форму, користувач бачить відображення даних у вигляді, що відповідає останній активізації форми. Для відображення змінених даних таблиці потрібно оновити дані.

На основі зведеної таблиці можна побудувати діаграму, яка

відображатиме лише дані зведеної таблиці. При зміні даних зведеної таблиці діаграма оновлюється автоматично.

Для створення форми із зведеною таблицею можна заздалегідь підготувати запит (це варто виконувати у випадку, коли поля зведеної таблиці розташовуються у різних таблицях бази). Нехай запит має назву **ЗведенаТаблиця** і містить наступний список полів: **СкидиВоди.Дата**, **СписокПідприємств. НаселенийПункт**, **СписокПідприємств.Назва**, **СкидиВоди.Об'єм** і **ПерелікВоднихОб'єктів.Назва** (рис. 4.53).

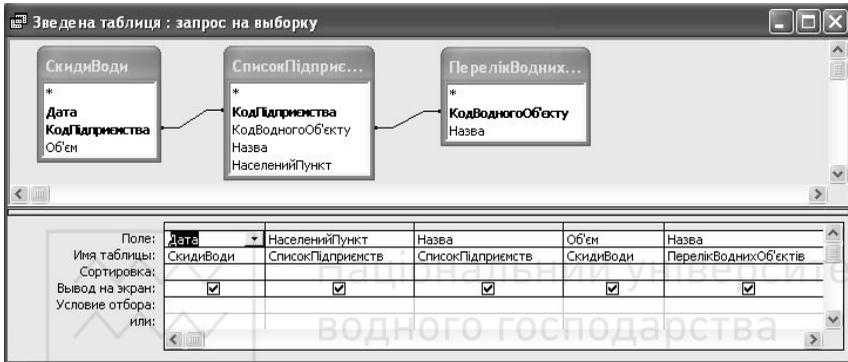


Рис. 4.53. Запит ЗведенаТаблиця в режимі Конструктор

Створюючи форму, у вікні діалогу **Новая форма** слід вибрати елемент **Сводная таблица** та вказати джерело даних (наприклад, запит **ЗведенаТаблиця**). На екрані з'явиться інформаційне вікно **Создание сводных таблиц**. Після натискання кнопки **Далее>** з'явиться нове вікно для формування списку необхідних полів. Натиснувши кнопку **Готово**, потрібно у вікні **макету** підготувати **структуру зведеної таблиці**, вказуючи у ролі стовпців та рядків назви відповідних полів, а також поле, яке буде брати участь у обчисленнях значень за рядками та стовпцями. Для цього поля із списку, що виводиться в окремому вікні справа, **транспортують** у відповідні області в залежності від їх подальших функцій. Реалізувати це можна наступним чином: у **Строка** - **Населений пункт**, у **Столбец** - **Годы**, у **Фильтр** – **Дата**, в **Итоги** – **Об'єм** (рис. 4.54).

Для організації підсумків слід клацнути кнопку  у стовпці **Об-**

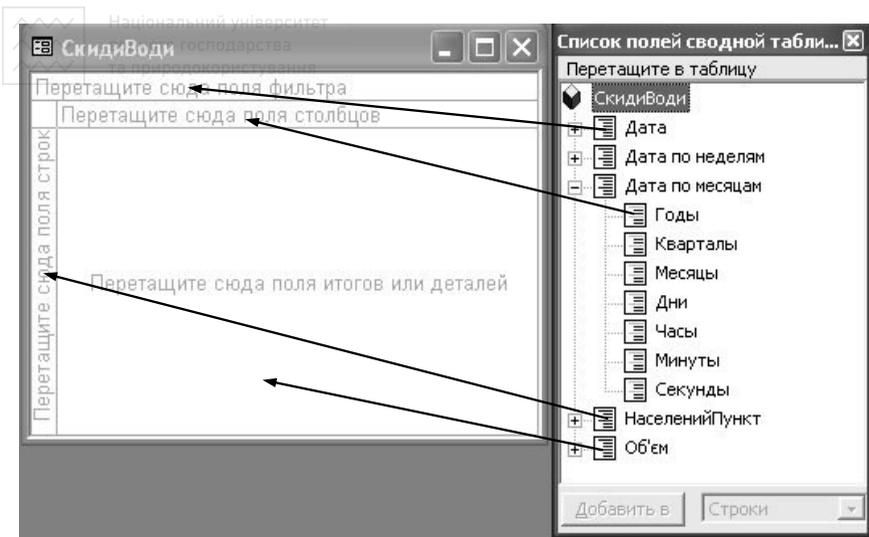


Рис. 4.54. Вікно макету форми Сводная таблица та вікно списку полів **Итоги** та виконати команду **Сводная таблица** / **Автовычисления** / **Сумма**, після чого клацнути  у стовпці **Общие Итоги**. Можна також поклати  біля назв населених пунктів, щоб не допустити повторного виведення даних. При потребі можна **фільтрувати** дані, наприклад, вивести скиди води з 2001 по 2003 рік. Описані дії дозволять отримати форму, показану на рис. 4.55.

		Годы			Общие итоги
		2001	2002	2003	
		+	+	+	+
НаселенныйПункт		Сумма "Объем"	Сумма "Объем"	Сумма "Объем"	Сумма "Объем"
Дубно	+	361	692	300	1353
Здолбунів	+	281	1004		1285
Костопіль	+	2710	1359	673	4742
Рівне	+	313	22160		22473
Общие итоги	+	3665	25215	973	29853

Рис. 4.55. Форма із зведеною таблицею Excel

У Access можна створювати складену форму, яка відображує записи зв'язаних таблиць. Така форма може бути подана у вигляді однієї

форми або у вигляді декількох зв'язаних форм, кожна наступна з яких відкривається за допомогою вбудованої у форму кнопки чи вимикача. У програмі передбачена можливість створення ланцюжка підлеглих форм.

Складені форми, як правило, створюються програмою-майстром. Їх будують на основі зв'язаних таблиць або існуючого запиту. Після вибору у вікні **Новая форма** варіанту **Мастер форм** формується **перелік полів** та вказується **один із можливих виглядів подання даних**: **Подчиненные формы** (передбачається перенесення інформації з однієї таблиці у іншу при заповненні їх даними) чи **Связанные формы** (використовують стандартно для перегляду даних). Далі для варіанту **Подчиненные формы** вказують **зовнішній вигляд** підлеглої форми (у рядок чи у вигляді таблиці), вибирають **стиль, імена** форм та **спосіб завершення** поточної дії. Для варіанту **Связанные формы** також вказують **стиль та імена** форм. Кожна із створених форм утворює окремий об'єкт у базі даних, який, до того ж, може виступати окремою незалежною формою. Приклад зв'язаних форм подано на рис. 4.56.

Використання фільтрів

Фільтрація даних - засіб відображення на екрані лише тих записів таблиці, які задовольняють певним умовам.

Фільтри виступають у ролі **масок** (шаблонів), які виводять на екран лише потрібні записи таблиці.

Access передбачає використання **трьох видів фільтрації**:

- **фільтрація згідно з виділенням** - відбір даних, які збігаються із виділеним фрагментом;
- **звичайна фільтрація** - пошук інформації відбувається відповідно до вмісту поля (чи полів) з використанням критерію;
- **фільтрація за допомогою розширеного фільтра** - відбір інформації відбувається за допомогою побудови складних критеріїв пошуку.

Перед створенням будь-якого фільтра потрібно пересвідчитися у тому, що на дані не накладено жодних умов, тобто всі фільтри знищені. В цьому випадку кнопка  **Применение фильтра** неактивна.

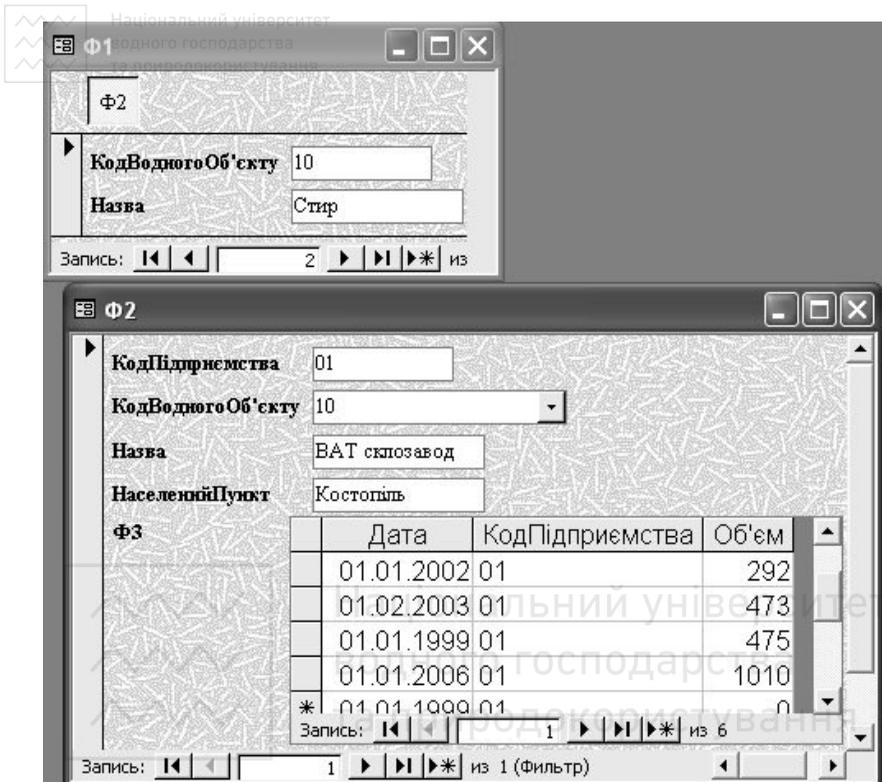


Рис. 4.56. Зв'язані форми у режимі їх перегляду

Зауваження: При активному фільтрі кнопка **Применение фильтра** замінюється на **Удалить фильтр**. Для відображення всіх записів потрібно повторно натиснути кнопку або виконати команду меню **Записи / Фильтр / Удалить фильтр**.

Для створення фільтра використовують команду **Записи/ Фильтр**. Цей пункт меню з'явиться при відкриванні або запуску об'єкта БД, тобто таблиці або форми.

Створення звичайного фільтра чи фільтра за виділеним фрагментом відбувається безпосередньо у вікні об'єкта (таблиці або форми). Створення розширеного фільтра здійснюється у вікні конструктора, яке складається з двох частин: **верхньої**, що містить вкладені вікна таблиць, та **нижньої** з бланком фільтра, де вказуються назви полів і умови, які на них накладаються, та параметри



Для створення фільтра згідно з виділеним потрібно, відкривши таблицю в режимі таблиці, встановити курсор у клітинку з даними, що виступають критерієм пошуку. Якщо пошук здійснюється лише відповідно до деякої частини вмісту клітинки, фрагмент необхідно



виділити. Далі клацаємо на кнопку **Фільтр по виділенному** панелі інструментів або вибираємо команду меню **Записи / Фільтр / Фільтр по виділенному**. На екрані будуть відображені лише ті записи, у яких є відповідність виділеному фрагменту.

Іноді потрібно вибрати дані, які не містять виділеного. У цьому випадку, клацнувши по клітинці із зазначеним фрагментом, потрібно викликати контекстне меню та вибрати команду **Исключить**



выделенное. Для коригування фільтра користуються кнопкою **Изменить фильтр**.

Критерії пошуку для одного фільтра при аналізі вмісту тільки одного поля формуються виключно за допомогою логічного оператора **Или (Або, Or)**. При здійсненні пошуку даних відразу в кількох полях критерій формується з використанням логічного оператора **И (I, And)**. Для виділення даних, які задовольняють декілька умов, зв'язаних логічним **И**, потрібно по черзі використати фільтри для кожного виділеного фрагмента.

Звичайний фільтр дозволяє здійснювати фільтрування за декількома умовами. Якщо умови зв'язані логічним оператором **I**, то їх потрібно ввести в один рядок фільтра, якщо ж умови зв'язані оператором **Або**, то потрібно клацнути на ярлику **Или** внизу вікна **фільтра** та ввести потрібну умову у відповідне поле.

Зауваження: При записі умов для фільтрування можна використовувати оператор **Like** (подібне до) та символи шаблонів * і ?.

Розширений фільтр – найпоширеніший спосіб фільтрування даних. Для його застосування виконують команду: **Записи / Фільтр / Расширенный фильтр**.

Сортування даних із застосуванням фільтрів

Програма Access дозволяє сортувати дані за десятьма полями одночасно. Сортування за одним полем було описано раніше. Для сортування записів одночасно за декількома полями застосовують

розширений фільтр. У нижній частині вікна конструктора розширеного фільтра потрібно розмістити поля в послідовності, за якою відбуватиметься сортування. Порядок сортування встановлюють на перетині рядка **Сортування**: та кожного з полів, за даними яких здійснюється сортування.

При закритті таблиці із збереженням змін, незалежно від способу фільтрації чи сортування, зберігається і сам фільтр. Щоб переконатися у цьому, можна після закривання таблиці із збереженням змін відкрити її знову і включити фільтр кнопкою **Применить** | **Удалить фильтр** панелі інструментів або відповідною командою меню. При цьому діятиме фільтр, встановлений перед останнім збереженням таблиці.

Для збереження таблиці без фільтра необхідно відмінити його дію та закрити таблицю із збереженням змін. Для відміни дії фільтра потрібно знищити всі накладені на поле умови у вікні **Фільтр**, яке відкриється після натискання кнопки **Изменить фильтр** панелі інструментів чи виконання відповідної команди меню.

Розробка запитів

Для пошуку та відбору інформації, а також для зміни та аналізу даних використовуються спеціальні об'єкти – **запити**. Вони можуть служити також джерелами записів для форм та звітів. Результатом виконання запитів є **динамічний** (тимчасовий) масив даних (англ. recordset). У такій таблиці можна додавати, змінювати чи знищувати записи, проте такі зміни не відображаються на даних основної таблиці. Запит можна зберегти як окремий об'єкт БД, але дані, отримані при його виконанні, не зберігаються (якщо не вказана спеціальна опція для збереження динамічної таблиці).

За своїм призначенням та результатом відбору запити поділяються на вибірки, запити-дії (створення нової таблиці, оновлення, доповнення, вилучення), параметричні, перехресні.

Програма Microsoft Access дозволяє проектувати запити вручну в режимі **Конструктор** або автоматично створювати їх за допомогою спеціальної програми **Мастер**. **Мастер** використовується при створенні простого запиту на вибірку, перехресного запиту, запиту на пошук записів, які повторюються, та запиту на пошук записів, не зв'язаних із записами інших таблиць. Створений майстром запит можна змінити, відкривши його в режимі **Конструктор**.



При створенні будь-якого запиту за зразком (QBE-запит) довільним способом (**Конструктор** чи **Мастер**), Access автоматично буде відповідний SQL-запит. Для перегляду SQL-запиту потрібно в режимі **Конструктор** виконати команду меню **Вид / Режим SQL** або скористатися кнопкою **Вид**.

Запит на вибірку дозволяє вибрати з бази даних інформацію, яка цікавить користувача в даний момент. Для створення запиту на вибірку треба перейти на вкладку **Запросы** і клацнути двічі ЛКМ на ярлику **Создание запроса с помощью мастера**. У вікні **Создание простых запросов** можна вибрати таблицю та сформувати список полів. На наступному кроці відмічаємо тип запиту (**подробный** чи **итоговый**) і для типу **итоговый** клацаємо кнопку **Итоги...**, вибравши в окремому вікні функцію для проведення обчислень. Це може бути сума (**Sum**), середнє (**Avg**), мінімальне (**Min**) або максимальне (**Max**) значення. Для типу **итоговый** потрібно додатково вказувати інтервал групування дат. Запит на вибірку буде показаний на вкладці **Запросы**

значком .

За допомогою **запитів-дій** користувач може створювати нові набори динамічних даних, змінювати або переносити дані в таблиці, додавати, знищувати групу записів. Відмінність цих запитів від запитів на вибірку полягає в тому, що вони не просто фільтрують дані, а й виконують певні дії над відфільтрованими даними.

Запит на створення нової таблиці дозволяє з динамічної (віртуальної) таблиці створити новий об'єкт - таблицю БД. Часто такі запити використовують для архівування записів або створення резервних копій таблиць. Запит на створення спочатку готується як звичайний запит на вибірку, а потім відкривається в режимі **Конструктор** і трансформується у потрібний за допомогою кнопки **Тип запроса**, у меню якої вибирається команда **Создание таблицы...** (рис. 4.57).

Запит на створення буде показаний на вкладці **Запросы** значком



Для створення цілком нової таблиці на базі існуючих спочатку створюють нову таблицю, яка складається з полів, властивості яких такі ж, як і властивості полів початкових таблиць.

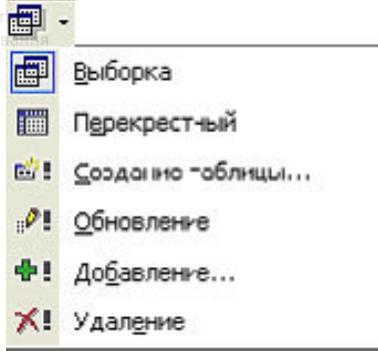


Рис 4.57. Меню кнопки Тип запроса

Після цього для кожної з таблиць, дані з яких увійдуть в нову таблицю, запускають на виконання запит на доповнення, а у діалоговому вікні **Добавление** вибирають назву новоствореної таблиці.

До запитів-дій належать **запити на доповнення** однієї таблиці записами іншої. Вони можуть працювати з таблицями як однієї бази, так і декількох. Таблиці при цьому не обов'язково повинні мати ідентичну структуру, але типи полів повинні бути однаковими. Для того, щоб додати записи однієї таблиці до іншої, потрібно на основі однієї з таблиць створити запит на вибірку, у який увійдуть всі поля таблиці, та змінити в режимі **Конструктор** тип запити командою **Запрос / Добавление**. Останню дію можна виконати також клацнувши в режимі **Конструктор** на панелі інструментів вікна БД кнопку **Тип запроса** та вибравши команду **Добавление...** (рис. 4.57).

У діалоговому вікні, яке після цього відкриється, вибирають назву таблиці, до якої потрібно долучити записи, та встановлюють місце розташування вибраної таблиці. При перегляді запити такого типу в режимі **Конструктор** можна помітити новий рядок **Добавление:** та назви полів таблиці, в які будуть переноситися дані з кожного поля.

Новий запит буде показаний на вкладці **Запросы** значком .

Запит на вилучення використовують тоді, коли з існуючої таблиці потрібно вилучити записи, що задовольняють певним критеріям. Для створення цього запити спочатку готують запит на вибірку, а потім у вікні **Конструктор** змінюють його тип командою **Запрос / Удаление** або виконують це відповідною командою меню кнопки **Тип запроса**. При цьому у бланку **Конструктора** запити з'явиться рядок **Удаление:**

з однаковим значенням у всіх колонках бланка. Умову відбору записів вказують у рядку **Условие отбора:**. Запит на вилучення має позначку



яка розміщується на вкладці **Запросы**.

Для архівування даних слід виконати послідовно:

- запит на створення нової таблиці;
- запит на доповнення записами нової таблиці;
- запит на вилучення тих записів основної таблиці, які збігаються із записами нової таблиці.

При об'єднанні декількох таблиць, які містять подібну інформацію, може трапитись так, що в об'єднаній таблиці опиняться записи, в яких повторюються дані деяких полів. Програма Access передбачає можливість:

- пошуку та вибору записів, які повторюються (утворення з них окремої таблиці);
- вилучення всіх груп записів, які повторюються;
- вилучення всіх повторів, залишаючи при цьому по одному запису з групи подібних.

Для пошуку та вибору усіх записів, які повторюються, зручно створювати запит за допомогою програми **Мастер**. Починають розробку запиту переходом на вкладку **Запросы** і клацанням кнопки **Создать**. У вікні діалогу **Новый запрос** вибирають елемент **Повторяющиеся записи**. Далі виділяють **таблицю або запит**, серед даних якої буде здійснюватися пошук повторів, і формують **список полів**. Причому, на другому кроці слід обов'язково задати хоча б одне поле, а на третьому кроці (вибір додаткових полів для виведення даних) їх можна не вказувати. На вкладці запитів буде відображено

новий об'єкт із значком  запиту на вибірку.

Для вилучення усіх груп записів, які повторюються, спочатку створюють запит на пошук записів, що повторюються. Після цього в режимі **Конструктор** змінюють тип запиту на **запит на вилучення**. Виконання такого запиту знищить усі групи записів таблиці, в яких є повторення.

Для того, щоб залишити у таблиці лише один запис з групи повторень, спочатку потрібно **створити нову таблицю** із структурою, ідентичною до даної. Відкривши новостворену порожню таблицю в режимі **Конструктор**, **ключовим** необхідно вибрати поле, де не може бути повторень (ключ можна вибрати складеним). Далі на



основі таблиці, з якої потрібно вилучити записи, треба **створити запит на доповнення записами**, у який ввійдуть усі поля таблиці. В ролі таблиці, до якої повинні додаватися записи, виступатиме новостворена порожня таблиця з новим ключовим полем. Попередню таблицю з повторами після створення нової можна знищити. Ключове поле при необхідності можна змінити.

Перехресні запити створюють тоді, коли є потреба об'єднати дані у форматі рядків-стовпців. Таблиці з результатами виконання перехресного запиту зручні для створення діаграм та графіків.

Перехресний запит буде показаний на вкладці **Запросы** значком . Створити перехресний запит можна за допомогою програми **Мастер** або в режимі **Конструктор**.

Для використання **Мастера** потрібно у діалоговому вікні **Новый запрос** вибрати елемент **Перекрестный запрос**, вказати **таблицю або запит** (наприклад, запит ЗведенаТаблица), **поле-заголовок рядка** (ПерелікВоднихОб'єктів.Назва), **поле-заголовок стовпця** (СписокПідприємств.Назва), **поле з даними для обчислень** (Об'єм), **функцію обчислень** (Сумма). Для проведення підсумкових обчислень за вибраною функцією для кожного рядка включають опцію **Вычислить итоговое значение для каждой строки** (при цьому у запит додається стовпчик з обчисленими значеннями). Результат виконання описаного запиту показаний на рис. 4.58.



	Назва	Итоговое	ВАТ "Воли"	ВАТ "Сми"	ВАТ сирз	ВАТ склс	ВДКП Рів
▶	Замчисько	11612			11612		
	Любомирка	1765		1765			
	Стир	2625				2625	
	Устя	25064	2381				22683

Запись: 1 из 4

Рис. 4.58. Перехресний запит в режимі таблиці

Для створення перехресного запиту у режимі **Конструктор** на бланку запиту вибирають потрібні **поля** з таблиць чи запитів, змінюють **тип запиту** за допомогою команди **Запрос / Перекрестный** або команди **Перекрестный** меню кнопки **Тип запроса** панелі інструментів Access. Після цього на бланку запиту з'являться два рядки – **Групповая операция** та **Перекрестная таблица** (рис. 4.59).

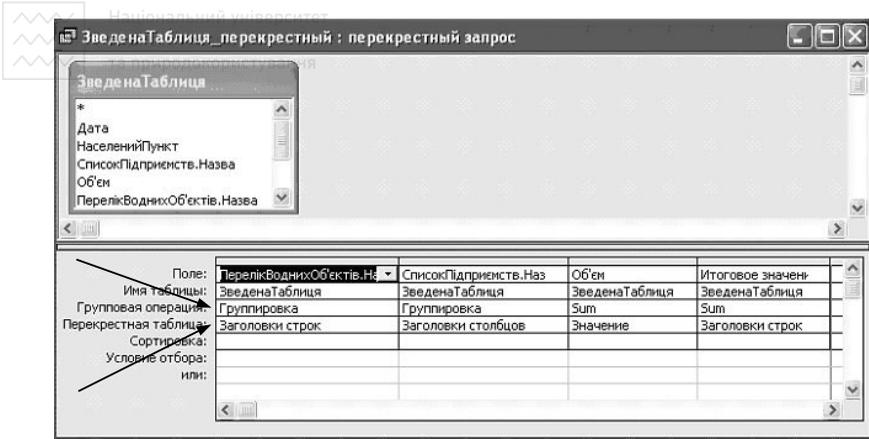


Рис. 4.59. Перехресний запит в режимі Конструктор

У рядку **Перекрестная таблица** потрібно вибрати із списку, що розгортається, один з можливих елементів: **Заголовки столбцов**, **Заголовки строк** або **Значение** від того, значення якого поля виступатиме у ролі заголовків стовпців чи рядків, а також дані якого поля будуть опрацьовуватися. У рядку **Групповая операция** для поля, значення якого аналізуються, потрібно вибрати функцію для проведення обчислень. Назви деяких функцій наведені в таблиці 4.14.

*Запити, які відрізняються між собою лише різними значеннями параметрів, називаються **параметричними**.*

Таблиця 4.14.

Приклади функцій групових операцій

Назва	Опис призначення функції
SUM	Сума значень поля
AVG	Середнє арифметичне значення
MIN	Мінімальне значення
MAX	Максимальне значення
COUNT	Кількість записів
STDEV	Середньоквадратичне відхилення значень даного поля
VAR	Дисперсія значень даного поля

Підготовку параметричного запиту спочатку можна виконувати довільним чином, наприклад, на вкладці **Запросы** клацнути кнопку **Создать** та вибрати **Простой запрос**. В якості джерела даних можна вибрати запит **ЗведенаТаблица**, сформуванати **список полів**, відкинувши всі поля з кодами та назву річки, встановити варіант **подробный** і завершити роботу з майстром.

Якщо відкрити запит в режимі **Конструктор**, то у нижній частині вікна в рядку **Вывод на экран**: можна помітити позначки у вигляді . Відмітку у стовпці **НаселенийПункт** для даного запиту доцільно зняти. Це відмінить виведення даних з цього стовпця, адже вони все одно будуть однаковими. У цьому ж стовпці у рядку **Условие отбора**: потрібно набрати у квадратних дужках текст-підказку. Текст буде використано пізніше при роботі із запитом для організації діалогу з користувачем. Після цього вікно **Конструктор** можна закрити. У випадку подальшого перегляду запиту в режимі **Конструктор** можна помітити, що стовпці, не відмічені позначкою , завжди виводяться останніми незалежно від того, де саме їх розташовували в процесі підготовки даного об'єкта. Взагалі забрати поле **НаселенийПункт** з вікна **Конструктор** не можна, оскільки воно задіяне в організації діалогу. Наведеному опису відповідає рис. 4.60. При виконанні параметричного запиту на екрані з'явиться вікно діалогу, зображене на рис. 4.61. Після введення значення параметра, тобто даних для відбору, з'явиться окреме вікно з результатами запиту (рис. 4.62).

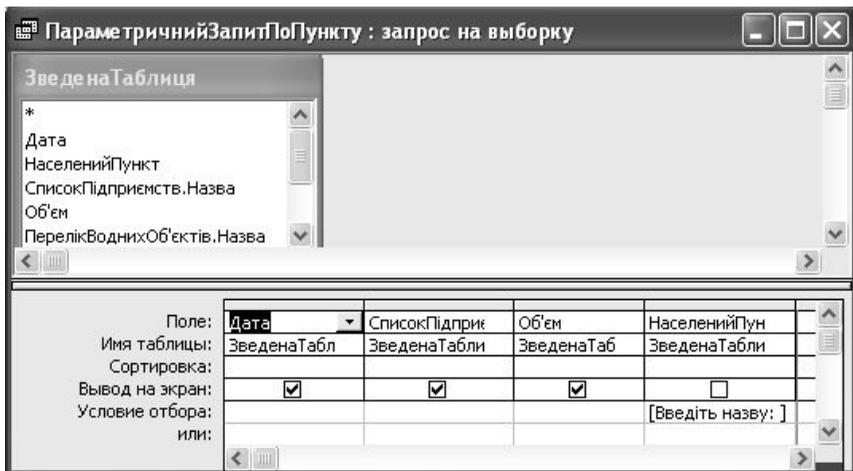


Рис. 4.60. Параметричний запит в режимі Конструктор

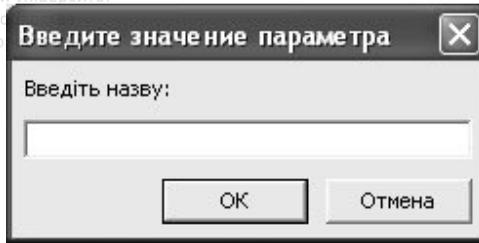


Рис. 4.61. Вікно діалогу параметричного запиту

	Дата	Назва	Об'єм
▶	01.01.1999	ВАТ "Смига"	130
	01.01.2000	ВАТ "Смига"	282
	01.01.2001	ВАТ "Смига"	361
	01.01.2002	ВАТ "Смига"	692
	01.02.2003	ВАТ "Смига"	300
	01.01.2006	ВАТ "Смига"	3030
*			

Запись: 1 из 6

Рис. 4.62. Результат виконання параметричного запиту при введенні значення параметра “Дубно”

Обчислювальні поля у запитах

Access не дозволяє виконувати обчислення безпосередньо у таблицях. Тому для проведення розрахунків та відображення отриманих результатів разом з основними даними у вигляді таблиць чи форм використовують запити. Обчислення виконують, як правило, перед формуванням звітів, які часто базуються не на таблицях, а саме на підготовлених запитах. Перерахунок застосовують також для автоматизації внесення однакових змін у велику кількість записів у випадку, коли нові значення обчислюються за відомою формулою. Формули можуть містити дані як з однієї, так і з кількох таблиць.

Вираз – комбінація операторів та операндів (константи, функції, імена полів, елементи керування та властивості), результатом виконання якої є конкретне значення.



Імена таблиць, звітів, запитів, полів та інших об'єктів при записі виразів вказуються у квадратних дужках, якщо вони містять пробіли. Для констант типу дата використовують символи #, а тексти беруть у апострофи. Наприклад: #01.01.07#, 'Рівне'.

Зауваження: Назви полів у виразах повинні збігатися з відповідними назвами в структурах таблиць і запитів, тому довгі імена доцільно не набирати з клавіатури, а копіювати.

При записі виразів у Access часто використовують **стандартні функції**, яких загалом налічується більше двохсот. Усі функції об'єднані в 16 категорій (груп).

Розглянемо приклади використання обчислювальних полів, продовживши роботу із створеною раніше базою даних **baza**. Нехай в таблиці **СкидиВоди** дані про кожне підприємство реєструються за підсумками року. Припустимо, що річний ліміт скиду забрудненої води для будь-якого підприємства становить 1000 м³. Потрібно встановити назви підприємств, які допустили перевищення ліміту протягом 2001 року, та обчислити величину відхилення від норми для кожного такого підприємства.

Для розв'язування задачі підготуємо простий запит на вибірку. Основою для цього запиту слід вибрати розроблений раніше запит **ЗведенаТаблиця**, оскільки він містить потрібні нам дані з різних таблиць: **Дата**, **НазваПідприємства**, **Об'єм**. Далі у вікні **Конструктор** необхідно на перетині рядка **Умовие отбора:** та стовпця **Дата** вказати умову відбору записів: **Year ([ЗведенаТаблиця]![Дата])=2007**. Тут Year() – функція, що виокремлює з дати рік. При цьому для формування умови доцільно



скористатися інструментом **Побудувати**. Після цього вводимо обчислювальний вираз у першу вільну клітину рядка **Поле:** **[ЗведенаТаблиця]![Об'єм]-1000**. Назву **Виразення1** можна відразу замінити на іншу, наприклад, **Перевищення**. Відмітку у вигляді рядка **Вивод на екран:** для полів **Дата** та **Об'єм** можна зняти, а для обчислювального поля - поставити. Варто доповнити рядок **Умовие отбора:** виразу **Перевищення** умовою **>0** для виведення тільки тих даних, що відповідають випадку перевищення ліміту. Структура описаного запиту розміщена на рис. 4.63.

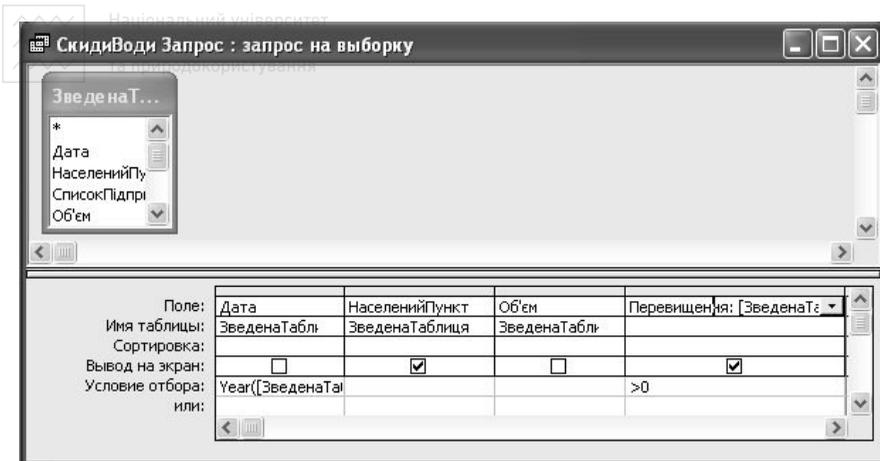


Рис. 4.63. Запит на вибірку з використанням обчислювального поля в режимі Конструктор

Розв'язок поставленої задачі є результатом виконання запиту, який показаний на рис.4.64.

Для проведення розрахунків та аналізу даних використовуються також запит на оновлення та операція групування. Кнопки



Тип запроса та **Групповые операции**, за допомогою яких можна перейти у відповідний режим, містяться на панелі інструментів **Конструктор запросов**, яка з'являється у вікні програ-

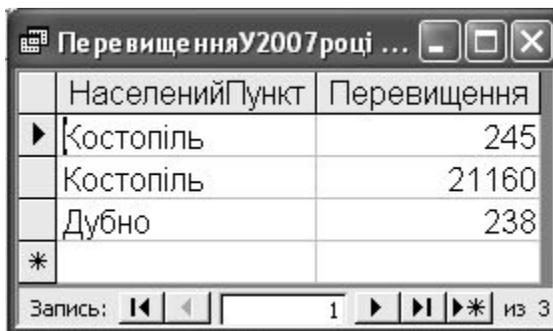


Рис. 4.64. Запит з обчислювальним полем в режимі таблиці ми в режимі конструктора запитів.

Запит на оновлення дозволяє швидко змінити дані у групі записів,

відібраних за певним критерієм. Створення такого запиту починають з підготовки звичайного запиту на вибірку, який після цього перетворюють у запит на оновлення за допомогою команди **Запрос / Обновление** програми Access або командою меню **Обновление** кнопки **Тип запроса**. Після цього з бланку запиту зникнуть рядки **Сортировка** и **Вывод на экран**, але з'явиться рядок **Обновление:**, який потрібно заповнити новим значенням або виразом. Бланк **Конструктора** потрібно заповнювати лише тими полями таблиці, з якими проводитимуться обчислення. Запити на оновлення мають позначку .

Групповые операции застосовуються для обчислення одного спільного значення для кількох записів. Функція групування встановлюється у тих полях, за якими здійснюється об'єднання даних. Застосування функцій групування вже згадувалося при описі перехресного запиту.

Розглянемо застосування ще однієї із стандартних функцій – функції **управління** **IIf()**. Вона записується у вигляді:

IIf (умова; значення1; значення2),

де **умова** – логічний вираз; **значення1** – значення, якого набуває функція, якщо логічний вираз істинний (**True**); **значення2** – значення, якого набуває функція, якщо логічний вираз хибний (**False**).

Нехай, наприклад, потрібно перевірити наявність перевищення ліміту скиду забрудненої води підприємством. Якщо ліміт становить 1000 м^3 , то рядок **Поле: вікна Конструктор запитів** буде містити наступне: **Повідомлення: IIf(СкидиВоди![Об'єм] >1000; 'Перевищення ліміту'; 'Норма')**. Результат перегляду такого запиту зображений на рис. 4.65.



Дата	Об'єм	Повідомлення
01.01.2001	313	Норма
01.01.2001	361	Норма
01.01.2001	1286	Перевищення ліміту
01.01.2001	1424	Перевищення ліміту

Запись: 1 из 27

Рис. 4.65. Запит з повідомленням в режимі таблиці

Формування звітів

Звіт (англ. report) – це кінцевий результат розв’язування задач з використанням СКБД. Дані у звіті можна подати у будь-якому вигляді (форматі) в залежності від поставленого завдання та потреб користувача. Звіти відображають зведену інформацію, яка отримана в результаті певних узагальнень. Причому інформація може подаватися як у текстовому варіанті, так і за допомогою діаграм.

Звіти мають широкі можливості для візуалізації та друку інформації. Усі звіти поділяються на три категорії: **звіти у рядок**, **звіти у стовець**, **поштові наклейки**. Цікавою особливістю Access є можливість створення **складених** звітів. За допомогою механізму підзвіту користувач може з’єднати декілька звітів в одному документі. Створені в Access звіти можна зберігати у форматі, який підтримується іншою програмою.

В Access звіти можна створювати за допомогою функції **Автозвіт**, за допомогою програми **Мастер** або в режимі **Конструктор**.

Найшвидший спосіб створення звіту – використання функції **Автозвіт**, коли програма запропонує вибрати один із існуючих шаблонів розміщення даних (у стовпчик чи в рядок). Для реалізації цього можна перейти на вкладку **Отчеты**, клацнути кнопку **Создать**, вибрати варіант **Автоотчет: ленточный**, вказати джерело даних, наприклад, запит **ПерехреснийДляЗведенаТаблиця**. Звіт після цього можна трохи доопрацювати. Наприклад, перейменувати його у **АвтозвітНаБазіПерехресногоЗапиту**, змінити в режимі **Конструктор** заголовок із сформованого автоматично на **Звіт за весь**

період спостережень в населених пунктах області, у верхньому колонтитулі замість **Итоговое значение Об'єм** вказати **Всього** тощо.

Створений автозвіт, поданий у режимі **Перегляд**, показаний на рис. 4.66.

Назва	Всього: ВАР "Волний"	ВАР "Сльга"	ВАР сирзавод
Замчислю	35792		35792
Любомарка	6033	6033	
Стыр	5110		
Устя	26789	3181	

Рис 4.66. Автозвіт після доопрацювання у режимі перегляду

Майстер допомагає користувачеві створити структуру звіту, а саме – вибрати поля з інформацією для аналізу, згрупувати та відсортувати дані, створити макет звіту та вказати параметри його друку.

На екрані звіт можна відобразити лише у **двох режимах: Конструктор та Перегляд**. За замовчуванням після вибору значка звіту на вкладці **Отчеты** звіт відображається у режимі перегляду. Якщо потрібно відобразити структуру звіту у режимі конструктора, то можна скористатися кнопкою **Конструктор** панелі інструментів вікна БД. Змінити відображення звіту на екрані можна також за допомогою команди **Вид** рядка меню програми або кнопки **Вид** панелі інструментів вікна програми.

Обчислювальні поля у звіті можна використовувати для проведення обчислень як з даними одного запису, так і для деякої групи записів чи усіх записів одного звіту. Якщо потрібно провести обчислення для групи записів, то обчислювальне поле розміщують в **область заголовка або примітки групи**. Якщо ж обчислення стосуються усіх записів звіту, поле повинно знаходитися у **примітці або заголовку звіту**. При створенні таких обчислювальних полів

використовують операцію групування. Змінити рівні групування можна у діалоговому вікні **Сортировка и группировка** командою **Вид/ Сортировка и группировка**.

У програмі Access передбачено можливість з'єднувати в одному звіті декілька підзвітів. При цьому користувач може проводити підсумкові обчислення за записами об'єднаних звітів. Для вставки підпорядкованого звіту у головний потрібно використати спеціальний елемент керування **Подчиненная форма / Отчет**. Для зв'язування звітів потрібно відкрити головний звіт в режимі **Конструктор** та відобразити на екрані вікно БД. З вікна БД необхідно перетягнути значок звіту, який повинен бути підзвітом головного, розмістивши його у потрібній області проекту звіту.

Для того, щоб створити фотографію звіту, яку можна переслати електронною поштою або переглянути за допомогою Web-браузера, виділяють потрібний звіт та виконують команду **Файл / Экспорт**. У діалоговому вікні **Экспорт объектов** у полі **Тип файла** необхідно вказати формат файлу **Снимок** та вибрати папку для реєстрації файлу.

Обмін даними з іншими програмами

Інформацію, яка міститься у базі даних, створеній за допомогою іншої СКБД, можна **імпортувати** у програму Access. В свою чергу Access-таблиці також можна **експортувати**, зберігаючи їх у форматах, які зрозумілі іншим форматам СКБД.

Для імпорту даних використовують **Майстер імпорту**, який можна запустити на виконання командою **Файл / Внешние данные / Импорт**. При цьому відкриється вікно діалогу **Импорт**, у якому потрібно знайти папку з файлом для імпорту та вибрати у полі **Тип файла** відповідний формат. В результаті база даних доповниться новою таблицею.

Access розуміє різні **формати** файлів, зокрема, текстові файли (.txt), а також файли, створені в Microsoft Excel, Paradox, dBASE V, Microsoft FoxPro, Lotus 1-2-3, ODBS DataBase.

Крім можливості імпортування даних з файлів іншої програми, в Access існує можливість **аналізувати дані, які подано у форматі іншої прикладної програми**, а також опрацьовувати їх без виконання операції імпортування. Такі дані програма сприймає як зовнішню таблицю. Вони залишаються у початковому вигляді і з ними можна працювати як в оригінальній програмі, так і в Access. Для виконання такої операції потрібно виконати команду **Файл / Внешние данные /**



Связь с таблицами. У вікні діалогу **Связь** необхідно вказати місцезнаходження і тип об'єкта та натиснути кнопку **Связь**. В результаті у базу даних буде вбудована нова таблиця, яка буде показана значком із стрілкою на вкладці **Таблицы**.

Для експорту даних потрібно виділити об'єкт, де знаходяться дані, та виконати команду **Файл / Экспорт**. У вікні діалогу **Экспорт объекта** необхідно вибрати папку та потрібний формат файлу, після чого клацнути кнопку **Сохранить**.

Крім імпорту та експорту, обмін даними можна здійснювати за допомогою **буфера обміну Microsoft Office** командами **Копировать**, **Вырезать**, **Вставить** або відповідними кнопками на панелях інструментів чи клавішними комбінаціями **Ctrl+C**, **Ctrl+X** та **Ctrl+V**.

Найпростішим способом переміщення чи копіювання об'єктів є **метод транспортування** за допомогою миші. Відмітимо, що копіюються не лише дані, але і параметри їх форматування.

При використанні буфера обміну або методу транспортування відсутній зв'язок між даними з бази даних та їх джерелами, тобто при зміні інформації у джерелі дані в таблицях Access автоматично не поновлюються. Враховуючи це, більш широко використовують спосіб **вбудовування та зв'язування об'єктів** (англ. **Object Linking and Embedding - OLE**). При вбудовуванні об'єкт, який поміщається у базу, стає її складовою частиною. Клацання двічі ЛКМ на вбудованому об'єкті забезпечує швидкий запуск потрібної програми та автоматичне завантаження об'єкта. В такий спосіб створюється можливість зручного редагування об'єкта на місці та фіксації внесених змін у базі даних. Об'єкт вбудовують лише у випадку необхідності його збереження після вилучення файлу-джерела. Для реалізації вбудовування об'єктів потрібно виконати команду **Вставка / Объект** і у вікні діалогу вибрати опцію **Создать новый** чи **Создать из файла**.

Вбудовування об'єктів можна також здійснювати через буфер обміну. Для цього копію об'єкта необхідно занести у буфер обміну, після чого виконати **Правка / Специальная вставка** та вказати можливий спосіб вставки, який залежить від типу об'єкта.

За необхідності динамічного обміну інформацією між джерелом та базою даних використовують **зв'язування об'єктів**. При цьому у базі даних зберігається код поля файлу-джерела. Зв'язок встановлюється аналогічно вбудовуванню об'єктів, але у вікні **Вставка объекта**

вмикається додатково опція **Св'язь**. Зв'язок між джерелом та приймачем можна **редагувати** або навіть **знищувати**. Для цього слід виконати команду **Правка / Св'язи OLE / DDE**.

Контрольні запитання і завдання

1. Дайте визначення бази даних.
2. Що являє собою система баз даних?
3. Які операції можна виконувати над даними?
4. Поясніть термін СКБД.
5. Назвіть відомі моделі даних. Опишіть реляційну модель даних.
6. Чому дані зберігають у багатьох таблицях?
7. З яких об'єктів складається база даних Access?
8. Які основні способи створення бази даних у середовищі MS Access?
9. Сформулюйте визначення таблиці бази даних. Як створити таблицю бази даних?
10. У яких випадках слід створювати нову таблицю бази даних за допомогою конструктора? Коли таблиці створюють з використанням майстра?
11. Сформулюйте вимоги до імен об'єктів у Access.
12. Дайте визначення поля, запису.
13. Коли для поля встановлюється властивість підстановки значень з іншої таблиці?
14. Перелічіть властивості полів.
15. Обгрунтуйте використання форм.
16. Як і для чого створюють фільтри?
17. Для чого створюють запити?
18. Що є характерним для звітів і як здійснити їх підготовку?
19. Як здійснити обмін даними з іншими програмами?



РОЗДІЛ 5. ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

5.1. Основи алгоритмізації

Розв'язування інженерної задачі з використанням комп'ютера передбачає наступні етапи:

- **постановка задачі.** На цьому етапі формулюється зміст, мета задачі, описуються вхідні дані та форми подання результатів розв'язку, вказуються обмеження задачі;
- **побудова математичної моделі** передбачає перехід від словесного опису задачі до математичного. Створення математичної моделі вимагає глибоких знань предметної області застосування задачі та математичного апарату;
- **розробка алгоритму розв'язування задачі;**
- **програмування** - опис алгоритму розв'язку задачі конкретною алгоритмічною мовою;
- на етапі **відлагодження і тестування програми** виявляють помилки, перевіряють відповідність одержаних результатів поставленій задачі на тестових прикладах;
- **аналіз отриманих результатів.**

Найбільш трудомісткою є розробка алгоритму розв'язування задачі.

Алгоритмом називається повністю визначена скінченна послідовність дій (правил), виконання яких забезпечує розв'язання задачі.

Основні властивості алгоритмів:

- **скінченність** – виконання алгоритму повинно закінчитися через скінченне число кроків;
- **результативність** – виконання алгоритму повинно приводити до певного результату. Воно не може закінчуватися невизначеністю або не закінчуватися взагалі;
- **визначеність** – алгоритм повинен бути написаний так, щоб при його виконанні у виконавця не виникало двозначних тлумачень. Іншими словами, різні виконавці згідно з алгоритмом повинні діяти однаково та прийти до одного і того ж результату;
- **масовість** – розроблений алгоритм повинен надавати можливість розв'язувати деякий клас задач.



Опис алгоритму розв'язування задачі можна здійснювати одним із способів:

- **словесним:** спосіб наближений до звичайної розмовної мови;
- **графічним:** спосіб передбачає подання алгоритму у вигляді схеми блоків;
- **запис алгоритму однією із мов програмування.**

Найбільш наочним і поширеним є опис алгоритму у вигляді блок-схем. Блоки, які найчастіше використовуються при розробці алгоритмів, показані на рис. 5.1.

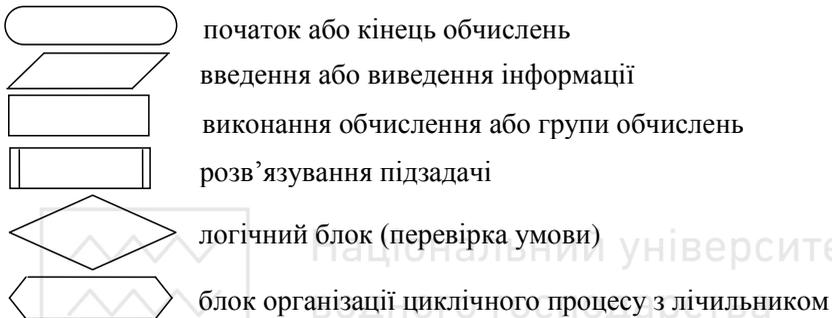


Рис.5.1. Стандартні блоки для опису алгоритмів

Блок-схема - послідовність з'єднаних між собою стандартних блоків, кожен з яких містить вказівку на виконання конкретної дії, що записана всередині блоку.

Декілька однотипних дій дозволяється записувати в одному блоці, якщо це не впливає на структуру та розуміння алгоритму. Блоки з'єднуються лініями із стрілками, які вказують на наступний блок. При необхідності, для кращого розуміння алгоритму, над стрілками роблять надписи та здійснюють нумерацію блоків. Часто в окремий блок виділяють деяку підзадачу або типовий алгоритм, які в програмі оформляють у вигляді підпрограм (процедур чи функцій). Приклади типових алгоритмів: обчислення суми, добутку, знаходження мінімуму тощо.

В загальному випадку алгоритм складається із стандартних елементів, які з'єднуються у лінійну, розгалужену або циклічну структуру. Розглянемо схеми алгоритмів різної структури.

Алгоритм лінійної структури передбачає виконання дій послідовно одна за одною в порядку їх запису. Фрагмент блок-схеми з

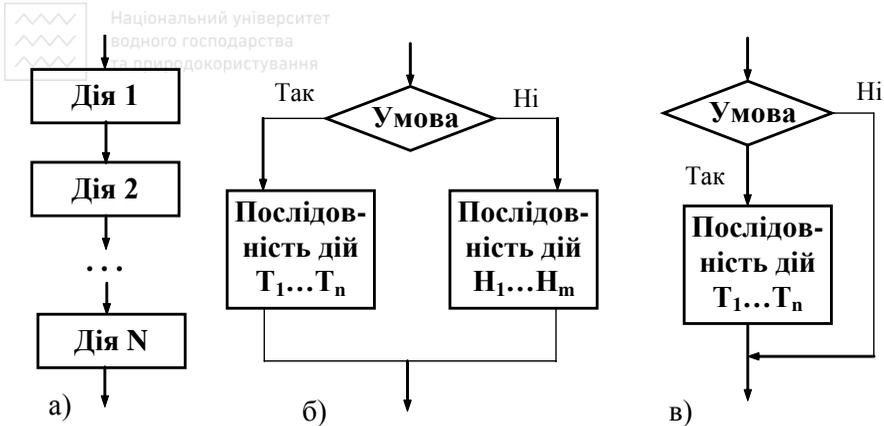


Рис. 5.2. Фрагменти блок-схем алгоритмів: а) лінійної структури; б) з повним розгалуженням; в) з неповним розгалуженням

таким порядком обчислень показаний на рис. 5.2 а).

В алгоритмах розгалуженої структури в ході обчислень передбачається вибір тієї чи іншої послідовності дій у залежності від виконання певної умови. Розрізняють повне (рис. 5.2 б)) та неповне розгалуження (рис. 5.2 в)).

Алгоритми циклічної структури передбачають багаторазове повторення деякої послідовності дій.

Циклом називають послідовність дій, яка неодноразово повторюється.

Дії, що виконуються в циклі, називають **тілом циклу**. Керування кількістю повторень виконання тіла циклу може здійснюється за допомогою змінної, яку називають **параметром циклу**. Для організації циклу повинні бути відомі: початкове значення параметра циклу, закон зміни цього параметра перед кожним новим повторенням тіла циклу, а також умова, при виконанні якої здійснюється вихід з циклу та виконується оператор, що розміщений поза межами циклу. Якщо умова закінчення циклу не виконується, відбувається повторне виконання послідовності дій з початку до кінця тіла циклу.

Блок-схеми алгоритмів циклічного обчислювального процесу наведені на рисунку 5.3: а) – цикл з передумовою; б) – цикл з післяумовою (постумовою); в) – цикл з лічильником (параметром).

Найпоширенішим прикладом циклічного обчислювального процесу є **цикл з параметром** (рис. 5.3. в)), у якому параметр циклу



змінюється за законом арифметичної прогресії (крок зміни є сталим) та кількість повторень заздалегідь задана або її можна визначити.

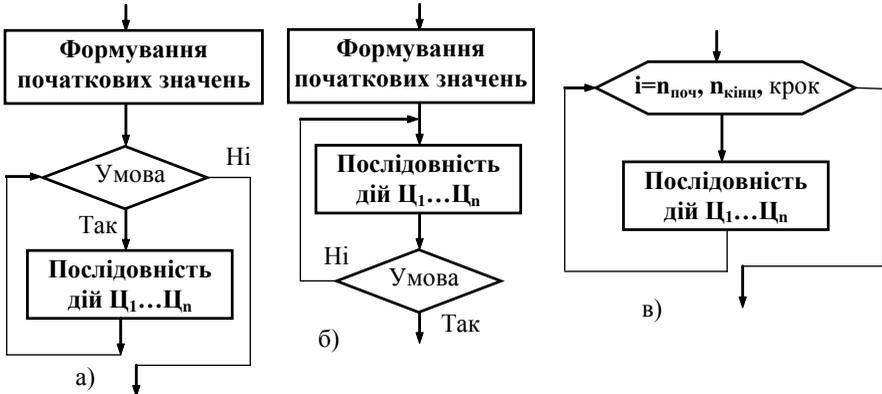


Рис. 5.3. Фрагмент блок-схеми алгоритму з циклом:

- а) із передумовою; б) із післяумовою; в) із параметром

В ітераційних алгоритмах (англ. iteration – повторення, результати обчислень однієї або декількох величини є входними даними для наступних обчислень цих же величин) кількість повторень наперед невідома. Вона визначається в процесі виконання тіла циклу (умову завершення повторень записують, як правило, на початку циклу). Для формування таких алгоритмів використовують цикли з перед або післяумовою.

Якщо в тілі циклу розміщений інший цикл (інші цикли), таку конструкцію називають **вкладеними циклами**. Кількість вкладених циклів не обмежується. Для кожного значення параметра зовнішнього циклу параметр внутрішнього циклу послідовно набуває всіх можливих значень. Вкладені цикли використовують, зокрема, при опрацюванні елементів масивів (векторів, матриць) та ін..

Алгоритми розв’язування інженерних задач в більшості випадків є складними та поєднують у собі різні типи обчислювальних процесів.

Приклад 1. Обчислити кількість автосамоскидів N_C , необхідну для безперервної роботи екскаватора за формулою:

$$N_C = \frac{t + t_3 + t_p + t_m}{t_3},$$



де $t = \frac{2L}{V_{cp}}$; L – середня відстань перевезення ґрунту, км; V_{cp} – середня швидкість руху самоскида, км/год; t_3 – час завантажування одного самоскида, год; t_p – розрахункова тривалість розвантажування, год; t_M – тривалість маневрування транспортних засобів, год.

Блок-схема алгоритму розв’язування задачі (лінійний обчислювальний процес) показана на рис. 5.4.

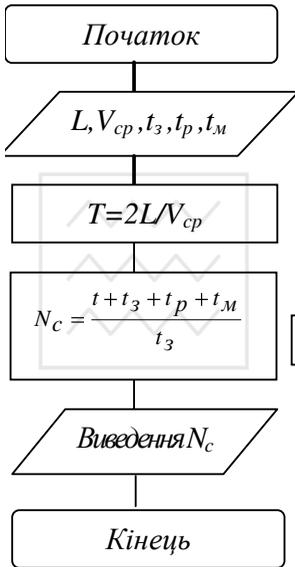


Рис. 5.4

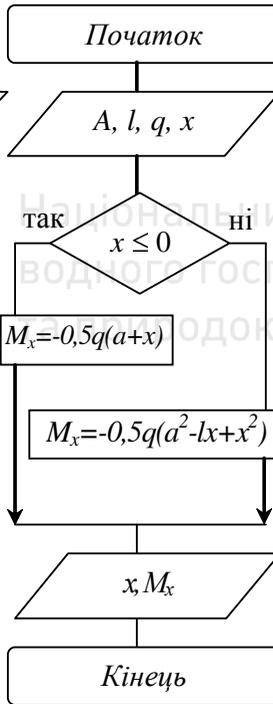


Рис. 5.5

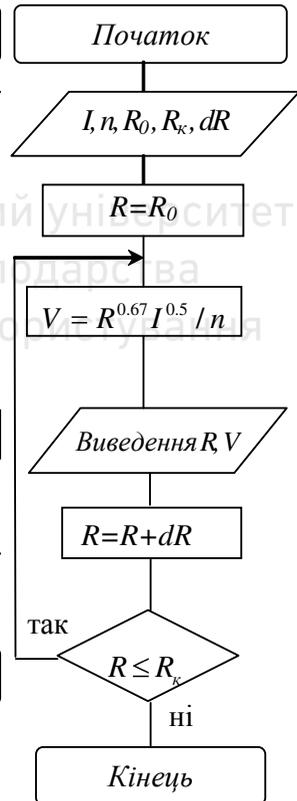


Рис. 5.6

Національний університет
з водного господарства та природокористування

Приклад 2. Скласти блок-схему алгоритму обчислення моменту згинання M_x в будь-якому перерізі однопрогінної балки з консолями,

якщо навантаження q розподілене рівномірно, за співвідношенням:

$$M_x = \begin{cases} -0,5q(a+x)^2, & \text{якщо } x \leq 0; \\ -0,5q(a^2 - lx + x^2), & \text{якщо } x > 0; \end{cases}$$

a - довжина консолі; l - відстань між опорами; x - абсциса перерізу.

Блок-схема алгоритму розв'язування задачі (обчислювальний процес з розгалуженням) зображена на рис. 5.5.

Приклад 3. Скласти блок-схему алгоритму обчислення швидкості руху рідини в трубі за формулою $V = \frac{R^{0,67} \cdot I^{0,5}}{n}$, де R – гідравлічний радіус, м; I – гідравлічний ухил потоку; n – коефіцієнт шорсткості труби. Початкові дані для проведення обчислень наступні:

$0,70 \leq R \leq 0,86$; $\Delta R = 0,02$; $I = 0,003$; $n = 0,014$.

Позначимо $R_0 = 0,70$; $R_k = 0,86$; $dR = \Delta R$. Блок-схема алгоритму розв'язування задачі містить циклічний обчислювальний процес з післяумовою (рис. 5.6).

5.2. Основи програмування мовою Visual Basic 6.0

Складання програми будь-якою мовою програмування полягає в послідовному описі всіх пунктів алгоритму розв'язування задачі засобами цієї мови.

Мова програмування BASIC була розроблена у 1963 році співробітниками Дартмутського коледжу для вивчення основ програмування та реалізації алгоритмів розв'язування інженерних задач. Починаючи з 1982 року вона стала основним інструментальним засобом, який включений в операційні системи фірми Microsoft. З розвитком техніки та комп'ютерних технологій вдосконалювалася і мова програмування. З'явилися версії Visual Basic, Visual Basic.Net, а в пакет MS Office вбудована мова програмування Visual Basic for Applications (VBA).

За своєю суттю Visual Basic 6.0 є потужним інтегрованим середовищем розробки нових додатків Windows з використанням об'єктно-орієнтованих методів та підходів. Знання основ програмування мовою Visual Basic є запорукою успішного розв'язання практичних задач з використанням сучасних комп'ютерних



5.2.1. Елементи мови програмування

Кожна мова визначається алфавітом, синтаксисом та семантикою. В синтаксисі мови будемо використовувати кутові (<, >) та квадратні ([,]) дужки. Текст, що записаний рядком символів та розміщений між кутовими дужками, позначає один неподільний елемент мови. Текст, розміщений у квадратних дужках, позначає необов'язковий елемент мови. Ключові слова будемо виділяти напівжирним шрифтом.

Алфавіт - сукупність символів, які можна використовувати в мові.

Алфавіт мови Visual Basic складають:

- великі і малі літери латинського алфавіту;
- арабські цифри;
- спеціальні символи: + - * / \ = < > . , () : ^ @ & ! % ' \$ # _
- пробіл.

В русифікованих версіях MS Windows та Visual Basic дозволяється використовувати символи кирилиці для формування імен ідентифікаторів.

Лексемами називають мінімально значущі одиниці тексту програми, а саме: спеціальні символи, ключові слова, ідентифікатори, константи.

Програма, записана мовою програмування Visual Basic, складається з лексем та роздільників. Роздільником служить пробіл. Дві сусідні лексеми, якщо вони позначають зарезервоване слово, ідентифікатор або константу, повинні бути відділені одним або декількома роздільниками.

Роздільник не може бути частиною лексеми, за винятком рядків символів. Для запису одного оператора (логічного рядка) тексту програми в декілька рядків, введено роздільник рядків, що складається з пробілу та символу підкреслення (_). Цей засіб дає змогу записувати текст програми так, щоб він повністю вміщався на сторінці екрана. Логічний рядок програми Visual BASIC може містити до 1023 символів і не більш як 10 роздільників рядків.

Ключове слово – зарезервований термін, значення якого визначене розробником мови і не може бути перевизначене користувачем.



До ключових слів відносять, наприклад, **End, Dim, As, New, Next, If, Then, Else, For, Sub**. За замовчуванням ключові слова у вікні редагування коду програми виділяються шрифтом синього кольору.

***Ідентифікатором** називається послідовність літер, цифр і знаку підкреслення, яка починається з літери і використовується для позначення імен констант, типів, змінних, процедур, функцій, модулів, полів у записах.*

Ідентифікатор може містити до 255-ти символів. Іноді використовується **уточнений ідентифікатор**, який складається з двох частин (ідентифікаторів), між якими стоїть крапка.

***Константою** називається елемент даних програми, значення якого не може змінюватися під час виконання програми.*

Константами є числа, рядки символів, дати. Значення константи визначається на етапі компіляції програми. Кожна константа може мати ім'я. Константи поділяють на **користувацькі** та **вбудовані** (визначені розробниками мови Visual Basic). Всі вбудовані константи починаються із символів vb. Вони призначені для зберігання найрізноманітнішої інформації: кодів клавіш, значення кольорів, властивостей об'єктів тощо. Крім того, є дві логічні константи, які позначаються ключовими словами **True** (істина) і **False** (хибність). Інформацію про існуючі константи, їхні значення та застосування можна отримати, скориставшись каталогом об'єктів (кнопка **Object Browser**, відповідний пункт меню або клавіша **F2**).

Цілі або дійсні **числа** можуть бути записані із знаком плюс або мінус або без знака. Дійсні числа записуються у **звичайній формі** (з фіксованою крапкою) або **експонентній** (з плаваючою крапкою). В останньому випадку дійсне число записується у вигляді mEr , де m – мантиса (дійсне число з фіксованою крапкою), r – порядок (ціле число).

Константи дати і часу записуються в різних форматах, але починаються і закінчуються символом #, наприклад: #10:10:00# (час доби), #01/31/2007# (дата).

Рядком символів (текстовим рядком) називається послідовність символів (в тому числі і тих, які не входять в алфавіт мови), що записані в одному рядку програми і обмежені з обох боків лапками. Рядок, який не містить символів, називається **порожнім рядком**.

Коментарем називається послідовність будь-яких символів, яка починається із апострофа або ключового слова **Rem**. Коментар, що починається із апострофа, записується після оператора до кінця рядка тексту програми. Ключове слово **Rem** обов'язково записується з початку рядка тексту програми, всі символи в цьому ж рядку після ключового слова **Rem** вважаються коментарем.

Змінна – ділянка оперативної пам'яті, що має ім'я та тип, в яку, під час виконання програми, записується певне значення.

Кожна змінна позначається ідентифікатором. Для доступу до вмісту ділянки оперативної пам'яті достатньо знати ім'я змінної. Це ім'я є унікальним і не змінюється в процесі виконання програми. Вибирати імена змінних бажано так, щоб вони містили інформацію про призначення змінної і не були занадто довгими.

5.2.2. Опис змінних та констант

У мовах програмування усі дані належать до певних типів. Будь-який тип визначає множину значень, які можуть набувати дані, і набір операцій, які можна застосувати до даних. Використовують наступні основні прості типи даних: числові (**Integer**, **Byte**, **Long**, **Currency**, **Single**, **Double**), рядковий (**String**), логічний (**Boolean**), дата (**Date**); а також довільний (**Variant**) і об'єктний (**Object**) типи. Детальніші відомості про типи даних вміщені в додатку 5.1. В іменах змінних рекомендується вказувати спеціальні префікси, що пояснюють тип даних.

Змінні можна описувати у програмах явно або неявно. Якщо змінна у програмі завчасно не описана, то її створення та визначення типу відбувається автоматично при першому присвоєнні значення.

Щоб не допустити виникнення помилок, які пов'язані з неявним описом імен змінних, модуль додатку повинен містити наступний оператор:

Option Explicit

Він вказує на те, що кожна змінна в даному модулі повинна бути описана до її використання.

Явний опис змінної здійснюється оператором

Dim <Ім'я змінної> [**As** <тип змінної>]

або з використанням суфіксу

Dim <Ім'я змінної>[<суфікс>]

Приклади опису змінних:

Національний університет
«Львівська політехніка»
та природничих наук

```
Dim curMoneu1 As Currency
```

```
Dim curMoneu2@
```

```
Dim Гроші As Currency
```

Тут змінні curMoneu1, curMoneu2, Гроші мають один і той же тип. Тип змінної curMoneu2 визначає суфікс @ (див. додаток 5.1).

У прикладі

```
Dim Moneu3
```

при описі змінної не вказано тип, тому вона автоматично матиме тип **Variant**, що відповідає опису **Dim Moneu3 As Variant**. Тільки змінні цього типу можуть багаторазово змінювати свій тип в залежності від значення, яке присвоюється цій змінній.

Приклади опису змінних рядкового типу:

```
Dim Адреса As String
```

```
Dim strName As String*20
```

Символ * та число 20 після ключового слова **String** вказує на те, що змінна strName може містити до 20 символів.

Приклад неявного опису змінної в тексті програми:

```
Manu@=45.76
```

В результаті, змінна Manu буде мати тип **Currency**, її тип надалі не можна змінювати (на відміну від типу **Variant**).

Змінні типу **Boolean** можуть мати тільки два значення: **True** (відповідає будь-яке ціле число, крім 0) та **False** (відповідає ціле число 0).

Присвоюючи значення змінним типу **Date**, необхідно враховувати наступне: якщо значення дати обмежене символами #, то використовується американський формат дати та часу; якщо обмежувачем служать лапки – застосовується встановлений в операційній системі формат дати і часу. *Наприклад:*

```
dtmDat = #02-22-2007# ' американський формат дати;
```

```
dtmDat = "22 лютого 2007 " ' український формат дати.
```

Опис змінних, окрім типу, визначає ще й область її застосування та період існування.

Областю видимості змінної називають ту частину програмного коду, в якій вона визначена та доступна.

Використовується три області видимості: **локальна, контейнерна, глобальна**. **Локальні** змінні визначаються в тілі процедури або функції та доступні тільки всередині цієї програмної одиниці. В



процедурах і функціях опис змінних можна також здійснювати з допомогою ключового слова **Static**, наприклад:

Static sngZ **As Single**

Значення локальних змінних, що описані з допомогою ключового слова **Static** (ці змінні ще називають статичними), зберігаються протягом усього часу виконання додатку до наступної зміни, а значення змінних, описаних з допомогою ключового слова **Dim**, існують тільки під час виконання процедури чи функції (їх значення до наступного виклику процедури не зберігаються). Якщо потрібно оголосити статичними всі локальні змінні процедури, ключове слово **Static** записують у заголовок процедури.

Контейнерні змінні доступні всім підпрограмам всередині того модуля (контейнера), в якому вони описані. Їх опис здійснюється з допомогою ключових слів **Dim** або **Private** та розміщується до опису підпрограм модуля в секції **General Declarations** (рис. 5.7).

Глобальні змінні доступні у всіх модулях додатку, що розробляється. Вони описуються з допомогою ключового слова **Public** та розміщуються в секції **General Declarations** стандартного модуля. Якщо, наприклад, змінна `intPi`, описана в модулі форми `Form1`, то використати її значення в інших формах можна за допомогою звертання `Form1.intPi`.

Іменовані константи можна описувати аналогічно опису змінних з допомогою оператора наступної структури:

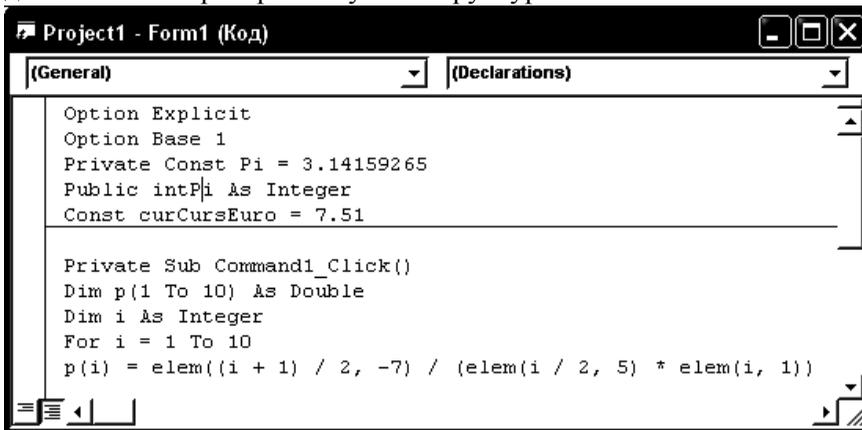


Рис. 5.7. Приклади описів змінних, констант і масивів

[Public | Private] Const <Ім'я константи> **[As <Тип константи>]**



Const strName = "Visual Basic"
Private Const Pi = 3.14159265
Const curCursEuro **As Currency** = 7.4862

5.2.3. Масиви

Масивом називається послідовність елементів одного типу, впорядкована за номером послідовності.

Доступ до елемента масиву здійснюється за іменем масиву та номером елемента в масиві (індексом). Кількість індексів визначає кількість вимірів. Масиви є одновимірні (вектори, лінійні списки), двовимірні (матриці) та інші (наприклад, набори матриць).

Для елементів масиву виділяється неперервна область пам'яті, в якій елементи зберігаються послідовно, зокрема, для матриці: спочатку розміщуються елементи першого рядка, потім другого і т.д.

Індексами є послідовні цілі числа з деякого діапазону. Для кожного індексу вказують нижню та верхню межі значень, що визначають кількість елементів. За замовчуванням значення нижньої межі дорівнює 0 (тобто, перший елемент масиву має нульовий індекс). Оператор наступної структури

Option Base 0 | 1

дозволяє встановити значення нижньої границі індексів масиву, що використовується за замовчуванням, 0 або 1. Цей оператор записується один раз у секції **General Declarations** модуля (рис. 5.7).

У мові програмування Visual Basic існують два **види масивів**: фіксованої розмірності (фіксовані) та динамічні. У **фіксованих** масивах верхня та нижня межа індексів не може бути змінена в процесі виконання програми. Опис таких масивів подібний до опису змінних:

Dim arrA1(11 **to** 20) **As Integer**

Dim arrA2(9) **As Integer**

Кожен із описаних тут масивів містить 10 елементів цілого типу. В першому масиві нумерація елементів починається з 11, у другому – з 0. Наступні два рядки ілюструють опис двовимірного масиву:

Dim arr_intB(1 **to** 10, 1 **to** 5) **As Integer**

Dim arr_intC(5, 4) **As Integer**



Масив `arr_intB` складається з 10 рядків, у кожному з яких є 5 елементів; нумерація рядків та стовпців починається з 1. У масиві `arr_intC` є 6 рядків та 5 стовпців (нагадаємо, що нумерація рядків та стовпців, за замовчуванням, починається з 0).

Динамічні масиви (масиви, що мають динамічні розмірності) застосовують у тих випадках, коли кількість елементів масиву наперед невідома. При виконанні програми, в міру необхідності, можна змінювати як кількість елементів масиву, так і його розмірність (кількість індексів). Опис динамічного масиву виконується у два етапи: спочатку здійснюється опис типу елементів масиву без вказування кількості, потім оператором **Redim** визначається розмірність масиву. *Наприклад:*

```
Dim arr_Матриця() As Integer ' опис типу  
Redim arr_Матриця(7) ' вказання розмірності  
Redim arr_Матриця(2 to 7, 4)
```

У наведених прикладах при зміні розмірності масиву всі значення елементів масиву втрачаються (значення числових елементів замінюються на 0, рядки символів – на порожні рядки, дати – на значення #0:00:00#). Якщо необхідно зберігати значення елементів масиву при зміні розмірності, то можна змінювати тільки верхню межу останньої розмірності, і в операторі зміни розмірності записується ключове слово **Preserve**, наприклад:

```
Redim Preserve arr_Матриця(2 to 7, 6)
```

У даному випадку кількість рядків матриці не зміниться, але добавиться два стовпці. Якщо спробувати змінити значення інших границь – виникне помилка виконання програми.

Для визначення нижньої границі розмірності масиву використовують функцію `LBound(<ідентифікатор масиву>[, <номер розмірності>])`, а для визначення верхньої – функцію `UBound(<ідентифікатор масиву>[, <номер розмірності>])`.

Щоб вивільнити пам'ять, яку займає динамічний масив, використовують оператор:

```
Erase <список >
```

де <список > – одне або декілька розділених комами імен масивів. Якщо в операторі **Erase** вказане ім'я фіксованого масиву, то його елементи заново ініціалізуються: елементи числового типу отримують значення 0; рядкового типу змінної довжини – "" (порожній рядок); рядкового типу фіксованої довжини – 0; типу



5.2.4. Вирази. Основні операції

Виразом називається сукупність операндів, з'єднаних між собою знаками операцій. **Операндами** є константи, змінні, елементи масивів, функції.

Правила запису арифметичних виразів:

- вираз записується в рядок зліва направо і виконується з урахуванням **пріоритету** операцій. Для зміни порядку виконання операцій у виразах застосовують круглі дужки;
- між двома операндами обов'язково записується знак операції;
- типи операндів повинні або збігатися, або бути сумісними.

Після обчислення значення виразу отримується результат певного типу, який залежить від типу операндів і виду операцій. Значенням змінної типу **Variant** також може бути **Empty** (відповідає неініціалізованій змінній цього типу), **Null** (як правило, при роботі з базами даних) та **Error** (для обробки помилок в процедурах з використанням функції CVerr).

Всі операції в таблиці 5.1 перелічені в порядку спадання їх пріоритетів (окрім операцій порівняння, які мають однаковий пріоритет). Результати виконання логічних операцій з операндами наведені в таблиці істинності (додаток 5.2).

Для визначення порядку виконання операцій у виразах існують **наступні правила:**

- операції з однаковим пріоритетом виконуються зліва направо;
- вираз, що міститься в дужках, обчислюється як окремий операнд.
- операнд, який знаходиться між двома операціями з різним пріоритетом, зв'язується з операцією, що має вищий пріоритет; операнд, який знаходиться між двома операціями з однаковим пріоритетом, зв'язується з операцією, що знаходиться зліва від нього.

Таблиця 5.1.

Основні операції

Унарні арифметичні	
+, -	збереження, протилежний знак

Бінарні арифметичні

^	піднесення до степеня		
*, /	множення, ділення		
\	цілочисельне ділення		
mod	залишок від ділення націло		
+, -	додавання, віднімання		
Рядкові операції			
+, &	конкатенація (об'єднання рядків тексту)		
Логічні операції		Операції порівняння	
Imp	імплікація	Is	входження
Eqv	еквівалентність	Like	подібність
Xor	виключаючи або	>=, <=	більше рівне, менше рівне
Or	або (диз'юнкція, логічне додавання)	>, <	більше, менше

продовження таблиці 5.1.

And	і (кон'юнкція, логічне множення)	<>	не дорівнює
Not	заперечення	=	дорівнює

Приклади запису виразів:

- а) $0.5 * q * (a ^ 3 - x / (q ^ 2 + 1)) + 2 ;$
- б) $(x >= a) Or (x <= a - 10) ;$
- в) `strFam + " " + strIm + " " + strBat;`
- г) `"Кора" Like "К?[а-я]*"` .

На рис. 5.8 у вікні **Немедленно** показані результати обчислення виразів у операторах VB6. Оператор **Print** дозволяє вивести значення виразу, який записаний після цього слова. Виклик вікна **Немедленно (Immediate)** здійснюється командою **Прямое Окно (Immediate Window)** пункту меню **Вид (View)** або комбінацією клавіш **Ctrl+G**. В цьому вікні дозволяється вводити команди та оператори VB6 і отримувати результат відразу після натискання клавіші **Enter**. В будь-який момент часу можна змінити раніше введену команду та отримати новий результат.



```
Немедленно
x=2
print x^4-1
15
print x^4-1<16
Правильно
print x^10
1024
print cos(0)
1
```

Рис. 5.8. Вікно термінового виконання команд

5.2.5. Основні вбудовані функції

Вбудованими (стандартними) функціями називають такі функції, обчислення яких є стандартними компонентами системи VB6. Вони не потребують додаткових визначень та описів в програмі.

Наявність стандартних функцій значно спрощує процес програмування при розв'язуванні задач з використанням математичних, рядкових та інших функцій. У таблиці 5.2 наведені основні математичні функції. Рядкові функції та функції роботи з датами розміщені в додатку 5.3. При розв'язуванні задач потрібні функції можна використовувати в операторах та виразах. Аргументом функції може бути будь-який вираз допустимого типу, значення якого не виходить за межі області визначення функції. Аргументи функцій записуються в дужках. Аргументи тригонометричних функцій вказуються в радіанах.

Таблиця 5.2

Основні вбудовані математичні функції

Функція	Опис дії функції
Abs(x)	Повертає модуль (абсолютне значення) x
Atn(x)	Повертає арктангенс x
Cos(x)	Повертає косинус кута x
Exp(x)	Повертає значення константи e в степені x

Fix(x) Int(x)	Повертають цілу частину дробового числа. Для від'ємних чисел Fix повертає найближче більше від'ємне ціле число, Int – найближче менше від'ємне ціле число
Log(x)	Повертає натуральний логарифм x
Rnd[(x)]	Генерує випадкове число між 0 та 1. Якщо присутній параметр, то він вказує на спосіб генерації. Перед викликом функції Rnd потрібно виконати оператор Randomize
Round(x [,intI])	Повертає число x , заокруглене до вказаної кількості десяткових знаків $intI$. Якщо кількість десяткових знаків не вказана, повертається ціле число
Sgn(x)	Повертає -1, якщо $x < 0$; 0, якщо $x = 0$; 1, якщо $x > 0$
Sin(x)	Повертає синус кута x
Sqr(x)	Повертає квадратний корінь з x
Tan(x)	Повертає тангенс кута x

Наведемо приклади запису виразів з використанням вбудованих функцій для обчислення деяких математичних величин.

1. Для знаходження числа π можна використати вираз: $\text{atn}(1)*4$
2. Щоб знайти значення математичних функцій $\arcsin x$; $\arccos x$; $\text{arctg } x$ використовують вирази: $\text{atn}(x/\text{sqr}(1-x^2))$; $\text{atn}(\text{sqr}(1-x^2)/x)$; $\text{atn}(1)*2-\text{atn}(x)$
3. Для обчислення значення $\log_a b$ використовується вираз: $\log(b)/\log(a)$
4. Для переведення числа x з градусної міри в радіанну можна використати вираз: $x*\text{atn}(1)/45$

5.2.6. Основні оператори

Додаток, записаний мовою Visual Basic, складається з модулів, які в свою чергу діляться на процедури. Кожна процедура містить оператори, що реалізують алгоритм розв'язку задачі. До основних операторів належать:

- оператор присвоєння;
- оператор виклику процедури;
- умовні оператори;
- оператор вибору;



Оператор присвоєння

Оператор присвоєння має вигляд:

<результат> = <вираз>

або

Let <результат> = <вираз>

де <результат> – змінна, елемент масиву, ім'я функції. Дія оператора присвоєння полягає в обчисленні значення виразу та присвоєнні цього значення величині, що записана зліва від знаку присвоєння. Тип виразу і змінної повинен бути однаковим або сумісним. *Приклади:*

$zx = (\text{sqr}(a) - 3 * a * b) / 10^{(1/3)}$

$L = M \text{ and } N$

Потрібно пам'ятати, що змінній числового типу можна присвоювати значення будь-якого числового виразу, при цьому тип результату перетворюється до типу змінної. Якщо результат обчислення виразу має більшу точність, ніж величина в лівій частині оператора присвоєння, то проводиться заокруглення обчисленого результату до потрібної точності.

Оператор виклику процедури

Для виклику процедури використовують оператор наступної структури:

Call <назва процедури> (<параметри процедури>)

або

<Назва процедури> <параметри процедури>

Робота з процедурами буде розглянута детальніше в пункті 5.5. Наведемо приклади виклику процедури `intMax`:

Call `intMax(x, y)`

`intMax x, y`

Умовні оператори

Умовні оператори реалізують алгоритмічні структури розгалуження та мають декілька різновидностей. Синтаксис умовного оператора (в один рядок):

If <логічний вираз> **Then** <оператор1> [**Else** <оператор2>]

Даний умовний оператор використовується в тому випадку, коли після ключових слів **Then** або **Else** потрібно записати лише один оператор. Якщо результатом обчислення логічного виразу є значення

True, то виконується оператор, що записаний після ключового слова **Then**. Якщо логічний вираз набуває значення **False**, виконується оператор, розміщений після ключового слова **Else** або оператор, записаний після цього умовного оператора (при відсутності ключового слова **Else** та оператора 2).

Синтаксис умовного оператора (в декілька рядків без додаткових перевірок умов):

```
If <логічний вираз> Then
    <послідовність операторів 1>
[ Else
    <послідовність операторів 2> ]
End If
```

В даному випадку в першому рядку (після ключового слова **Then**) не можна записувати жодного оператора. В рядку, де записане ключове слово **Else**, можна записати інший оператор, але перед ним потрібно поставити двокрапку.

Синтаксис умовного оператора (в декілька рядків з додатковими перевітками умов):

```
If <логічний вираз1> Then
    <послідовність операторів 1>
[ ElseIf <логічний вираз2> Then
    <послідовність операторів 2>
    ... ]
[ Else
    <послідовність операторів N> ]
End If
```

В цій структурі три крапки позначають можливість багаторазового повторення інструкції **Elseif ... Then ...**

Дія умовних операторів, що записані в декілька рядків, аналогічна дії цього ж оператора, записаного одним рядком. Після ключових слів **Then** і **Else** в наступних рядках дозволяється записувати не менше одного оператора. Після ключового слова **Elseif** записується додатковий логічний вираз (додаткова перевірка умови). Після виконання операторів, які записані нижче ключових слів **Then** чи **Else**, управління передається оператору, який розміщений в наступному рядку після рядка із ключовими словами **End if**.

Як альтернативу умовному оператору можна використовувати функцію **Iif** наступної структури:



Iif (<логічний вираз >, <значення 1>, <значення 2>)

Вона повертає в точку виклику перше значення, якщо результатом обчислення логічного виразу є значення **True**. В іншому випадку повертається друге значення.

Приклади запису логічних операторів та функції **Iif**, які дають однаковий результат обчислень:

```
If ss > 0 Then
    Text3.Text = " > 0"
Else
    Text3.Text = " <= 0"
End If
If ss > 0 Then Text3.Text = " > 0" Else _
    Text3.Text = " <= 0"
Text3.Text = Iif (ss > 0, " > 0", " <= 0")
```

В усіх трьох випадках властивості **Text** об'єкта **Text3** присвоюються рядок символів " > 0" або " <= 0" в залежності від значення числової змінної **ss**.

Оператор вибору

Синтаксис оператора вибору:

```
Select Case <вираз>
Case <діапазон значень_1>
    <послідовність операторів_1>
[Case <діапазон значень_2>
    <послідовність операторів_2>
... ]
[Case Else
    <послідовність операторів N>]
End select
```

Діапазоном значень може бути: одне значення виразу; перелік значень, розділених комою; діапазон значень. Приклади запису діапазону значень:

```
-5
-1, 1, 11
100 to 200 'будь-яке число в межах від 100 до 200
Is > 200 'будь-яке число, більше від 200
-1, 10, 20 to 30, is >50
```



Якщо в операторі вибору обчислене значення виразу належить <діапазону значень_1>, то виконується <послідовність операторів_1>. Якщо значення належить <діапазону_2>, виконується <послідовність операторів_2>. Якщо значення не належить жодному з перелічених діапазонів і присутня інструкція **Case Else**, то виконується <послідовність операторів N>.

Наприклад:

```
Select Case 5 * ss - 5
Case 0
    Text3.Text = " = 0"
Case Is > 0
    Text3.Text = " > 0"
Case Is < 0
    Text3.Text = " < 0"
End Select
```

Оператори циклу

Оператори циклу реалізують алгоритмічні структури циклів з параметром, передумовою, післяумовою. Розглянемо ці оператори.

Структура оператора циклу з параметром, блок-схема якого зображена на рис. 5.3. в), має вигляд:

```
For <ім'я змінної> = <знач.1> To <знач.2>
    [Step <знач.3>]
    [<послідовність операторів>]
Next [<ім'я змінної>]
```

де <ім'я змінної> – параметр циклу, <знач.1>, <знач.2> – початкове та кінцеве значення параметра циклу відповідно, <знач.3> – крок зміни значень параметра циклу. Якщо відсутній крок зміни параметра циклу – приймають значення кроку рівним одиниці.

Приклад запису циклу з параметром:

```
For i = 0 To 90 Step 2
    arrA(i \ 2) = i * atn(1) / 45
Next i
```

В цьому прикладі спочатку змінній **i** присвоюється значення 0 та виконується перевірка, чи не перевищує це значення 90. Оскільки результат перевірки **True**, то виконується тіло циклу до ключового слова **Next**. На наступних кроках проходження циклу значення параметра збільшується на крок 2 і його знову порівнюють з числом 90. На останньому кроці (при $i=90$) параметр збільшиться на 2 і в

результаті перевірки отримаємо значення **False**. Тому тіло циклу вже виконуватися не буде, а управління буде передано оператору, який записаний після рядка з текстом **Next i**.

Оператор циклу з передумовою, алгоритм якого показаний на рис. 5.3. а), може бути записаний за одним із варіантів:

Do While <вираз> <послідовність операторів> Loop	While <вираз> <послідовність операторів> Wend	Do Until <вираз> <послідовність операторів> Loop
--	---	--

Оператор циклу з післяумовою, алгоритм якого показаний на рис. 5.3. б), може мати одну із наступних структур:

Do <послідовність операторів> Loop While <вираз>	Do <послідовність операторів> Loop Until <вираз>
--	--

В цих операторах циклу використовується вираз логічного типу.

Ключове слово **While** вказує на завершення циклу в тому випадку, коли вираз, що записаний після нього, набуде значення **False**. В іншому випадку (значення виразу **True**) послідовність операторів між словами **Do** та **Loop (While, Wend)** буде виконана знову. Ключове слово **Until** вказує на завершення циклу тоді, коли вираз, що записаний після нього, набуде значення **True**.

В операторі циклу з передумовою оператори тіла циклу можуть не виконатися жодного разу, якщо при першому обчисленні значення виразу отримаємо **False**. В операторі циклу з післяумовою оператори тіла циклу виконуються хоча б один раз.

Фрагмент програмного коду реалізації задачі табулювання значень функції з використанням оператора циклу з передумовою може мати вигляд:

```
x = xp
Do While x <= xk
  If x > =1 Then
    y = (Sqr(a * x) - b) / Tan(x) ^ 9
  Else
    y = Exp(-x ^ 2) * (a + b * x) / Log(x) ^ 2
  End If
  Print x, y
  x = x + dx
```



Розглянемо деякі інші оператори, які полегшують розробку програм мовою Visual Basic.

Оператор Exit

Оператор **Exit** дозволяє здійснити вихід з тієї програмної структури (оператора, процедури, функції), де він розміщений. Синтаксис оператора:

Exit Do | For | Function | Sub

Приклад запису оператора виходу:

Exit Do

Ключове слово **Do** в операторі **Exit** означає вихід з оператора циклу **Do ... Loop** та передачу управління наступному оператору, який розміщений після цього циклу. Вихід із програмної структури не означає її завершення та очищення або закриття всіх величин.

Оператор End

Синтаксис оператора:

End [Function | Sub | If | Select | Type | With]

Застосування оператора **End** з одним із ключових слів є обов'язковим і означає завершення певної програмної структури чи відповідного оператора. Оператор **End**, записаний без ключових слів, дозволяє завершити той додаток, де він знаходиться (здійснює завершення виконуваного проекту, очищення та закриття всіх величин та об'єктів, закриття файлів), Зайнята програмою оперативна пам'ять, при такому завершенні роботи, вивільняється. *Приклад* завершення умовного оператора:

End If

Оператор переходу GoTo

Оператор переходу здійснює безумовну передачу управління на рядок всередині процедури, що має мітку. Синтаксис оператора:

Goto <мітка>

Мітка є ідентифікатором або числом з двокрапкою, наприклад:

Goto M1

. . .

M1: **End Sub**



Оператор опису користувачьких типів **Type**

Для створення користувачем власних типів даних можна використовувати оператор **Type** наступної структури:

```
Type <ідентифікатор типу>  
    <ідентифікатор 1-го елемента> As <тип>  
    [<ідентифікатор 2-го елемента > As <тип>  
    . . . ]  
End Type
```

Наприклад:

```
Type Stud  
    Прізвище As String*20  
    ДатНародж As Date  
    Ідн As Long  
End Type
```

Тут описаний новий тип даних Stud, в якому є три елементи даних різного типу. Змінні такого типу прийнято називати **записами**, а елементи запису – **полями**. Опис змінних типу Stud може бути наступним:

```
Dim Студент As Stud 'Студент – запис типу Stud
```

```
Private Студенти(20) As Stud 'масив з 20 записів
```

Значення присвоюються полям запису. Доступ до полів здійснюється наступним чином:

```
<ідентифікатор запису>.< ідентифікатор поля>
```

Тобто, між ідентифікаторами запису та поля ставиться крапка (так званий уточнений ідентифікатор).

Наприклад:

```
Студент.ДатНародж = #12 березня 1990#
```

5.3. Діалогові вікна

Діалогові вікна використовуються для відображення інформації, введення даних та керування роботою програми. Вони можуть бути наступних видів: **нестандартні** – створюються користувачем, **вбудовані** – створені розробником системи програмування Visual Basic, **стандартні** – реалізовані розробником системи програмування та викликаються методами елемента керування Common Dialog. Діалогове вікно може бути **модальним**, тобто таким, що не дозволяє перевести фокус на інше вікно та спонукає користувача ввести



інформацію або виконати деяку дію. **Немодальні** діалогові вікна дозволяють перевести фокус на інше вікно без закриття активного вікна.

Звернемо увагу на вбудовані вікна, які реалізовані двома стандартними функціями MsgBox та InputBox.

Функція MsgBox забезпечує формування вікна повідомлень (рис. 5.9) та має наступний синтаксис:

MsgBox (<повідомлення> [,<кнопки>] [,<заголовок>]
[,<довідка>, <розділ>])

де: <повідомлення> – обов'язковий параметр, який містить рядок символів, що відображається всередині діалогового вікна; <кнопки> – параметр, який визначає кількість та тип кнопок (див. додаток 5.4), за замовчуванням значення параметра дорівнює 0; <заголовок> – текст, який виводиться у заголовок вікна, якщо його немає – відображається назва додатку; <довідка> – текстовий рядок, що містить назву файлу контекстної довідкової системи; <розділ> – числове значення, яке вказує на розділ файлу довідки, що стосується даного діалогового вікна.

У діалогових вікнах функції MsgBox можна виводити значки, значення констант яких наведено в додатку 5.5. При використанні значків параметр, який визначає кнопки функції, збільшується на значення константи, що відповідає тому чи іншому значку.

Приклади реалізації діалогових вікон (рис. 5.9):

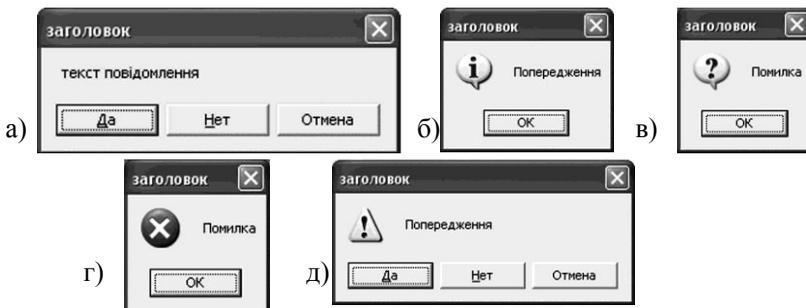


Рис. 5.9. Вікна повідомлень

- а) `kk= MsgBox ("Текст повідомлення", 3, "заголовок")`
- б) `kk = MsgBox ("Попередження", 64, "заголовок")`
- в) `MsgBox "помилка", 32, "заголовок"`

г) MsgBox("помилка", 16, "заголовок")

д) kk = MsgBox("Попередження", 51, "заголовок")

(у наведених прикладах змінній kk присвоюється значення коду натиснутої кнопки)

Функція InputBox забезпечує формування вікна для введення інформації. Вона має наступний синтаксис:

InputBox(<повідомлення> [,<заголовок>] [,<значення>]

[,<Xпозиція>] [,<Yпозиція>] [,<довідка>, <розділ>])

де <повідомлення> – обов'язковий параметр, що містить рядок символів, який відображається в діалоговому вікні; <заголовок> – текст, який виводиться у заголовку вікна, якщо його немає – відображається назва додатку; <значення> – значення, яке виводиться у текстове поле та присвоюється за замовчуванням. Параметри <довідка>, <розділ> мають те ж саме призначення, що й у функції MsgBox, а параметри <Xпозиція>, <Yпозиція> вказують координати верхнього лівого кута вікна.

Функція InputBox повертає рядок символів, введений користувачем. Вона використовується для введення даних з клавіатури, які присвоюються змінним. Оскільки змінні не завжди мають тип рядка символів, потрібно робити перетворення типів та, можливо, контроль за введеною інформацією. Якщо у вікні функції InputBox користувач вибирає кнопку **Отмена**, то повертається порожній рядок.

Приклад реалізації діалогового вікна (рис. 5.10):

```
kk1 = InputBox("Введіть відстань" +  
Chr(10) + Chr(13) + "між дренами (м)", _  
"Введення даних", 40)
```



Рис. 5.10. Вікно введення значень відстані між дренами

В повідомленні даного вікна використовуються символи переведення каретки Chr(13) та переходу на новий рядок Chr(10).



5.4. Робота з файлами

Використання файлів є важливим етапом роботи користувача за комп'ютером. У файлах зберігаються результати роботи, з них зчитуються дані для виконання розрахунків. Найчастіше використовуються операції зчитування інформації із файлу в змінні програми та запис значень змінних у файли. З файлами також можна виконувати всі стандартні операції (див. додаток 5.6), які дозволяє операційна система: копіювання, знищення, перейменування тощо.

Існує декілька різновидностей файлів, які визначаються форматом даних: **двійкові (бінарні)**; **текстові**; **типізовані** (файли записів). За методом доступу файли можуть бути **прямого** (до них належать двійкові та типізовані) чи **послідовного** доступу (наприклад, текстові файли).

Перед початком роботи з файлом його потрібно відкрити. Файл будь-якого типу відкривається оператором **Open**, який має різний синтаксис для різних типів файлів. Одночасно можуть бути відкриті декілька файлів. Для цього кожному файлу у програмі ставиться у відповідність спеціальна файлова змінна, яку ще називають **файловим номером**. Будь-які дії з файлом виконуються з використанням звертання до відповідної файлової змінної. Визначення першого вільного номера для файлу, що відкривається, здійснюється функцією **FreeFile**. Повернене цією функцією значення присвоюється змінній цілого типу, яка і буде файловим номером.

Файли будь-якого типу закривають оператором

Close <файловий номер>

З файлами тісно зв'язане таке поняття, як **вказівник** поточного елемента файлу (поточної позиції). Він показує, який елемент файлу є **активним** в даний момент і для якого можна виконувати операції читання або запису. Для двійкових та текстових файлів номером елемента є номер байту, для типізованого – номер запису. Нумерація всіх елементів починається з 1. Оскільки змінні складаються з певної кількості байт, для коректної обробки даних потрібно точно визначати позицію елемента файлу. Для цього визначають розмір змінної, що записується або зчитується, у байтах з допомогою функції **LenB**(<змінна>). Перехід на потрібний елемент файлу здійснюється оператором наступної структури:

Seek [#]<файловий номер>, <позиція>



Визначення номера елемента файлу, який в даний момент часу є активним, здійснюється функцією **Seek**(<файловий номер>).

Для перевірки умови досягнення кінця файлу в операції зчитування даних використовують функцію **Eof**([#]<файловий номер>). Якщо досягнуто кінця файлу, то ця функція повертає значення **True**, в іншому випадку – **False**. Розмір файлу в байтах можна визначити функцією **Lof**([#]<файловий номер>).

Двійкові файли (довільний доступ). Вміст таких файлів розглядається як послідовність байт з довільними даними, які розглядаються у бінарному вигляді. Вказівник активного елемента вказує на певний байт у файлі. Всі операції над даними здійснюються поелементно, при цьому вказівник зміщується на певну кількість байт, що відповідає розміру змінної в байтах, яка бере участь в операції. Для коректної обробки файлу потрібно використовувати змінні відповідного типу та встановлювати вказівник активного запису на перший байт інформації для змінної.

Оператор відкриття двійкового файлу випадку має наступну структуру:

Open <назва файлу> **For Binary As** <файловий номер>

де <назва файлу> – рядок або змінна типу **String**, що містить назву файлу. Якщо шлях не вказано, то робота з файлом відбувається в робочому (активному) каталозі; **For** – ключове слово для визначення типу файлу; **Binary** – ключове слово, яке використовується при відкритті або створенні двійкового файлу; **As** – ключове слово для визначення змінної, що містить файловий номер; <файловий номер> – ім'я змінної, яка містить номер файлу. Відкриття двійкового файлу відбувається відразу для читання та запису.

Запис даних у двійковий файл здійснюється оператором

Put [#]<файловий номер>, [<позиція>], <змінна>

де <позиція> – порядковий номер байту, починаючи з якого буде заноситися інформація у файл. Якщо цей параметр відсутній, запис проводиться в наступний елемент; <змінна> – ідентифікатор змінної, значення якої записується у файл.

Для зчитування даних з двійкового файлу у змінну використовується оператор

Get [#]<файлова_змінна>, [<позиція>], <змінна>

Типізовані файли (довільний доступ). Інформація в таких файлах зберігається у вигляді набору однакових записів певної структури. Кожен запис у файлі – деякий набір даних певних типів, які мають строго визначений номер та розмір. Розміром запису є значенням суми розмірів всіх даних (полів) у байтах. Доступ до даних здійснюється за номером запису, що є номером елемента у файлі. Відкривання файлу здійснюється оператором наступної структури:

```
Open <назва файлу> [For Random [Access [<доступ>]]] _  
As <файловий номер> Len = <розмір>
```

Режим **Random** використовується за замовчуванням для типізованих файлів. Ключове слово **Access** визначає режим доступу: **Read** – для читання; **Write** – для запису; **Read Write** – для читання та запису. Якщо доступ не вказаний, використовується режим **Read Write**. Після слова **Len** та знаку = вказується розмір одного запису файлу у байтах. Якщо розмір запису у байтах, що записується у файл, має більший розмір, ніж вказаний при описі, то виникає помилка.

Запис і читання даних з файлу відбувається операторами тієї ж структури, що і для двійкових файлів, тільки під позицією тут розуміють порядковий номер запису в типізованому файлі.

Приклад операцій з типізованими файлами

```
Dim Студент As Stud  
Nrec = LenB(Студент)  
FP = "a.dat"  
FN = FreeFile  
Open FP For Random As FN Len = Nrec  
.  
.  
.  
Put FN, 5, Студент  
.  
.  
.  
Get FN, 3, Студент  
Close (FN)
```

Текстові файли (послідовний доступ). Вмістом таких файлів є послідовність рядків довільної довжини, які були створені будь-яким текстовим редактором або іншим чином. При цьому в кінці кожного рядка міститься сполучення символів, що означають його закінчення (символи з кодами 13 та 10). Будь-який файл завершується спеціальним символом з кодом 26 (позначається EOF).

Операції з файлами, які відкриті як текстові, повинні виконуватися послідовно символ за символом або цілими рядками. При цьому після



опрацювання чергового символу або рядка вказівник автоматично переміщується на наступний символ (байт) або початок наступного рядка. При зчитуванні інформації посимвольно потрібно передбачати обробку кінця рядка.

Для відкриття файлу використовується оператор

```
Open <назва файлу> For <призначення> _  
      As <файловий номер>
```

де <призначення> може набувати наступних значень: **Input** – зчитування інформації; **Output** – виведення даних у файл; **Append** – доповнення файлу новою текстовою інформацією.

Файл, який відкривається для зчитування інформації (призначення **Input**), повинен існувати, інакше виникає помилка. Ключові слова **Output** та **Append** дозволяють відкривати існуючі чи створювати нові текстові файли. Відкриття існуючого файлу для виведення інформації (**Output**) приводить до знищення інформації, яка була у файлі.

Формування текстових файлів здійснюється операторами наступної структури:

```
Print #<файловий номер>, <список виводу>  
Write #<файловий номер>, <список виводу>
```

Якщо елементи списку виведення розділені комами, то у файлі текстова інформація відокремлюється символами табуляції. Якщо ж елементи списку записані через крапку з комою, то дані у файл записуються без роздільників. Символьні дані, записані оператором **Write #**, беруться в лапки, а числа виводяться без лапок. Між даними одного рядка ставиться кома.

Ці оператори здійснюють виведення у файл, починаючи:

- з першої позиції при першому виведенні, якщо файл відкритий як **Output**;
- з позиції, розміщеної відразу після останнього елемента файлу, якщо файл відкритий як **Append**;
- з активної позиції, якщо вона була встановлена оператором **Seek**, або після останнього виведеного символу.

Для прикладу, на рис. 5.11 наведений результат виконання наступного тексту програми:

```
Dim Fn1 As Integer, Fn2 As Integer  
Dim stro As String, stro1 As String  
Dim nFn1 As String, nFn2 As String
```



```
Fn1 = FreeFile  
nFn1 = "file1.txt"
```



Рис. 5.11. Результати виведення у файл операторами

Print # та Write #

```
Open nFn1 For Output As Fn1  
Fn2 = FreeFile  
nFn2 = "file2.txt"  
Open nFn2 For Output As Fn2  
For i = 1 To 5 Step 2  
stro = InputBox("Введення даних", , _  
"Рядок " + Str(i))  
stro1 = InputBox("Введення даних", , _  
"Рядок " + Str(i + 1))  
Print #Fn1, stro, i + 100, stro1; i + 150  
Write #Fn2, stro, i + 100, stro1; i + 150  
Next  
Close Fn1  
Close Fn2
```

Для зчитування рядків тексту з файлу у змінну використовується оператор

Line Input #<файловий номер>, <змінна>

Дані, записані з допомогою **Write #**, можуть бути прочитані оператором **Input #**. Для зчитування у змінну заданої кількості символів використовується функція **Input**(<кількість символів>, [#]<файлова_змінна>)



Наведемо приклад фрагмента програми зчитування всієї інформації з текстового файлу та виведення її у форму:

```
Do Until EOF(Fn1)
Line Input #Fn1, stro
Print stro
Loop
```

Використання оператора циклу з передумовою, в якій виконується перевірка умови досягнення кінця файлу, дозволяє здійснити послідовне введення відповідними операторами всієї інформації з файлу будь-якого типу, не здійснюючи ніяких операцій переходу всередині файлу.

5.5. Користувацькі процедури та функції

Із окремих операторів будуються крупніші блоки додатку – процедури, які, як правило, реалізують певну, логічно завершену, частину алгоритму розв'язування задачі.

Процедурою називають іменовану послідовність певним чином оформлених операторів, яку можна багаторазово виконувати.

Виклик процедури, записаної в додатку, здійснюється з будь-якого місця програмного коду. Процедури бувають двох видів: процедури обробки подій та загальні процедури.

Процедури обробки подій зв'язані з об'єктами VB6 і виконуються у випадку появи події. При цьому враховується як вид події, так і ім'я елемента, з яким зв'язана подія.

Ім'я процедури обробки подій завжди складається з двох частин, що з'єднані символом `_`: назви об'єкта та назви події. Наприклад, **Form1_Load()** – назва процедури, яка автоматично виконується при відкриванні форми з назвою **Form1**; **Command1_Click()** – процедура, яка автоматично виконується при клацанні на кнопці з назвою **Command1**.

Загальні процедури не зв'язані з об'єктами VB6 і подіями додатку.

Загальні процедури виконуються тільки тоді, коли вони викликаються з інших процедур. Ці процедури, як правило, реалізують спільні дії або алгоритми розв'язування певних задач.

Загальні процедури поділяють на підпрограми, вбудовані функції, користувацькі функції.



Синтаксис опису процедури-підпрограми:

```
[Public | Private | Static] Sub NameSub[(<параметри>)]  
    [<оператори>]
```

End Sub

Синтаксис опису користувацької процедури-функції (надалі будемо її називати функцією):

```
[Public | Private | Static] Function NameFun[(<параметри>)]  
    [As <тип>]
```

```
    [<оператори>]  
    <NameFun> = <вираз>  
    [<оператори>]
```

End Function

В заголовку процедур слова NameSub та NameFun є будь-якими ідентифікаторами, що позначають ім'я підпрограми або функції відповідно. В якості параметрів процедур необхідно вказувати імена тих змінних, значення яких потрібні в даний момент.

Приклад. Знайти змочений периметр каналу, що має форму трапеції, за наступною формулою: $x = b + 2h\sqrt{1 + m^2}$, м, де b – ширина дна каналу, м ($b=0,8$ м); m – коефіцієнт закладання відкосів ($m=1.25$); h – глибини заповнення каналу, м.

Текст програми, в якому обчислення змоченого периметра реалізовано функцією **perum(a As Single)**, може бути наступним:

```
Const m = 1.25  
Const b = 0.8  
  
Private Sub Form_Load()  
Dim x As Single  
Dim h As Single  
h = Val(TextBox( _  
    "Глибина заповнення каналу = ", _  
    "Введіть h, м", "1.0"))  
x = perum(h)  
MsgBox "x= "+Str(x), vbOKOnly, _  
    "Змочений периметр"  
End  
  
End Sub  
  
Private Function perum(a As Single) As Single
```


$$\text{perum} = b + 2 * a * \text{Sqr}(1 + m \wedge 2)$$
End Function

В результаті виконання програми (процедури обробки події завантаження форми **Form_Load()**) користувачу буде запропоновано ввести у вікні значення h (у рядку введення вже записано за замовчуванням значення $h=1.0$). Після введення значення та

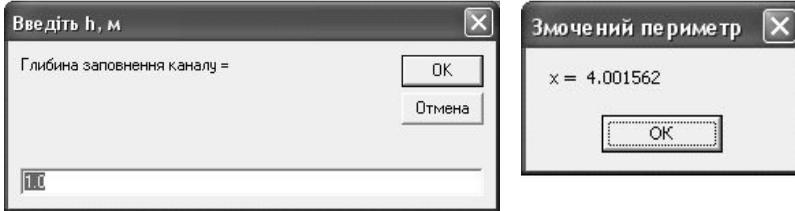


Рис. 5.12. Результати роботи програми для обчислення змоченого периметра каналу

натискання кнопки **OK** на екрані з'явиться вікно з результатами обчислень (рис. 5.12).

З наведених прикладів та структури опису процедур бачимо, що процедури-підпрограми або функції можуть мати тип **Private**, **Public** (за замовчуванням) або **Static**. Процедура, описана з ключовим словом **Private**, доступна всім іншим процедурам даного модуля. Наявність ключового слова **Public** означає, що процедура доступна іншим процедурам у всіх модулях. Слово **Static** означає, що всі локальні змінні процедури зберігають свої значення між викликами цієї процедури (тобто є статичними).

Основні особливості підпрограм та функцій:

1. Функція містить оператор присвоєння назві функції значення виразу.
2. Виклик функції може бути розміщений в операторі.
3. Функція повертає в місце виклику значення, яке присвоєне назві функції.
4. Підпрограма не містить значень, які зв'язані з її назвою. Обчислені в підпрограмі величини передаються в місце виклику через параметри або глобальні змінні.

В операторі виклику процедури-підпрограми **Call**, якщо вона не має параметрів, після назви процедури записують відкриваючу та зариваючу круглі дужки. Виклик функції може здійснюватися аналогічно до виклику підпрограми. В цьому випадку функція не



повертає ніякого значення. Процедури можуть бути рекурсивними, тобто викликати самі себе.

Параметри процедур передаються одним із двох способів:

1) передавання за посиланням; параметри процедур називаються **параметрами-змінними** (це відбувається за замовчуванням). Змінені значення параметрів передаються в місце виклику. При такому способі передавання параметрів в описі процедури може вказуватися ключове слово **ByRef**;

2) передавання за значенням використовує **параметри-значення**. В цьому випадку значення параметрів не змінюються, змінюються лише копії цих параметрів і змінені значення не доступні в місці виклику. При такому передаванні перед відповідним параметром в заголовку опису процедури обов'язкова наявність ключового слова **ByVal**.

Приклад тексту програми, яка реалізує передачу параметрів різними способами:

```
Private Sub Command1_Click()  
    Dim a As Integer, b As Integer  
    a = Val(Text1.Text)  
    b = Val(Text2.Text)  
    Call param1(a, b)  
    Text3.Text = Str(a)  
    Text4.Text = Str(b)  
    a = Val(Text1.Text)  
    b = Val(Text2.Text)  
    Call param2(a, b)  
    Text5.Text = Str(a)  
    Text6.Text = Str(b)  
End Sub  
Private Sub param1(ByRef a, ByRef b)  
    a = a + b  
    b = a + 5  
End Sub  
Private Sub param2(a, ByVal b)  
    a = a + b  
    b = a + 5  
End Sub  
Private Sub Command2_Click()
```



End
End Sub

Результат роботи програми показаний на рис. 5.13.

Перед іменем параметра можна вказувати ключове слово **Optional**, яке позначає, що даний параметр необов'язковий. При цьому всі наступні параметри повинні бути вказані з цим ключовим словом. Наведемо приклад опису функції з необов'язковими параметрами:

```
Private Function perum(Optional a _  
    As Single = 5) As Single  
    perum = b + 2 * a * Sqr(1 + m ^ 2)  
End Function
```

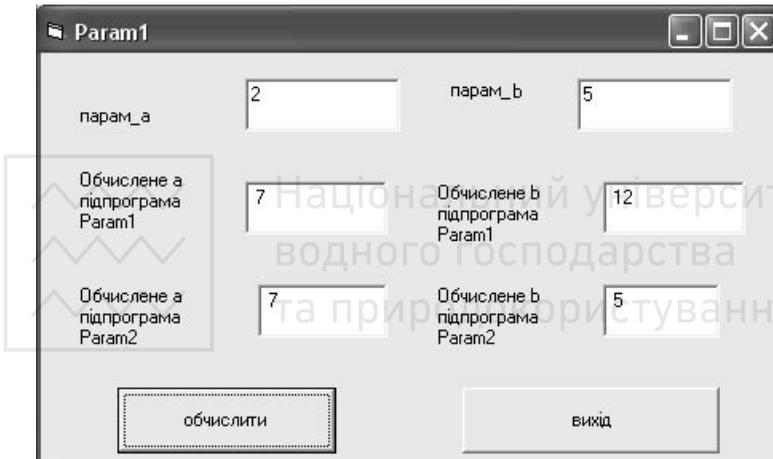


Рис. 5.13. Результат роботи програми ілюстрації способів передачі значень параметрів

Для перевірки наявності параметра типу **Variant** використовується функція **isMissing(<ім'я параметра>)**. Якщо параметр вказаний, повертається результат **False**, інакше – **True**.

Якщо кількість параметрів процедури наперед не відома, останнім параметром процедури може бути **Optional** - масив елементів типу **Variant**, описаний з ключовим словом **ParamArray**. Це ключове слово не може застосовуватися із **ByVal**, **ByRef** або **Optional**.



5.6. Виведення інформації

Розв'язування інженерних та наукових задач передбачає виведення результатів виконання обчислень у потрібній формі, зокрема, текстовій чи графічній. Ми вже розглянули виведення інформації у файли, а також використання для цього діалогових вікон. Крім того, виведення інформації можна здійснювати у форму чи малюнок. Результати обчислень, які перетворені у текстовий формат даних, присвоюються деяким елементам керування (TextBox, Label тощо).

Виведення інформації безпосередньо у форму або малюнок здійснюється методом **Print**. Його синтаксис:

[<Ім'я об'єкта>.]**Print** <список виведення>

Інформація виводиться вниз та вправо, починаючи з тієї позиції, яку задають значення властивості **CurrentX** і **CurrentY** – відповідно горизонтальна і вертикальна координати точки на об'єкті. Пікселі виведених символів замінюють пікселі об'єкта без можливості відновлення. Елементи списку виведення відокремлюються комою або крапкою з комою. Якщо елементи відокремлені комою, кожен наступний елемент виводиться з початку наступної зони, ширина якої дорівнює ширині 14 символів. При наявності символу ";" між елементами, вони виводяться без пропусків. Кожне наступне виконання методу **Print** приводить до виведення інформації з початку нового рядка. Якщо метод **Print** завершується символами ";" або ":", то виведення продовжується в поточному рядку. Коли метод **Print** застосовується без параметрів – здійснюється перехід на новий рядок.

Після використання для очищення вмісту об'єкта методу **Cls**, значення властивостей **CurrentX** і **CurrentY** дорівнюють 0 і метод **Print** виводить інформацію, починаючи з лівого верхнього кута.

При виведенні інформації на формах і малюнках різні частини тексту можуть відображатися різними шрифтами. В інших об'єктах зміна шрифту впливає на весь текст, що знаходиться в ньому. Вибір шрифту та його властивостей здійснюється присвоєнням параметрам властивості **Font** оператором наступної структури:

[<Ім'я об'єкта>.]**Font**.<параметр>=<значення>

Наприклад:

```
Form1.Font.Name="Arial"  
Form1.Font.Size =20  
Form1.Font.Bold=True
```

У цьому прикладі для наступного виведення у форму з іменем Form1 встановлений напівжирний шрифт **Arial** з розміром 20.

В залежності від встановлених параметрів один символ може займати різну кількість пікселів на екрані. Зокрема, один символ шрифту **MS Sans Serif** розміром 8 пунктів на об'єкті займає в середньому 5 пікселів в ширину (ширина різних символів – різна), висота рядка при цьому дорівнює 13 пікселів. У шрифтах типу **Courier** ширина всіх символів однакова. У методі **Print** для виведення із заданої позиції поточного рядка використовують функцію **Tab**(<номер позиції>). Під однією позицією тут розуміють ширину символу. Виведення вказаної кількості пробілів здійснюється функцією **Spс**(<кількість>).

Для виведення інформації у потрібному форматі використовується функція **Format**(<вираз>, <формат>). Другим параметром є іменованний формат або рядок символів, значення кожного з яких наведене в таблиці 5.3. *Приклади* іменованих форматів: Currency – грошовий формат, виводиться два символи справа від десяткової крапки числа, зліва враховується розділювач для тисяч; Long Date – повний формат виведення дати.

Приклади використання методу **Print** :

```
Print Format(s, "percent")
Print Tab(10); Format(sum, "00.0000E+")
Form1.Print Format(sum, "$#,###.##")
Picture1.Print Format(Now, "General Date")
```

Таблиця 5.3.

Призначення символів у рядку формату

Символ	Призначення
0	виведення символу, якщо він є, інакше нічого; якщо виводяться числа, вони зліва доповнюються нулями
#	виведення цифри або символу, якщо вони є, інакше нічого
.	вивід десяткової крапки
,	виводить пробіл, який є розділювачем тисяч
%	перетворює число у проценти (домножає на 100 та

водно	дописує знак %)
E- , e- , E+ , e+	виведення у експонентній формі (знак + показує, що знак перед показником степеня завжди присутній)
< , >	переведення символів у верхній або нижній регістр

Якщо у форматі не передбачає виведення нулів, то всі незначущі нулі на початку числа і в кінці дробової частини числа ігноруються. Всі символи, які не позначають той чи інший формат, виводяться на об'єкт у такому вигляді, в якому записані у символному рядку формату.

Приклад виведення інформації на малюнок (шостий рядок) :

```
Picture1.Cls
Picture1.ScaleMode = vbPixels
Line (100, 100)-(200, 200)
For i = 1 To 20
Picture1.CurrentY = (i - 1) * 13
Picture1.Print "вантуз № "; i
Next
```

Приклад виведення (присвоєння) в елемент TextBox (четвертий рядок):

```
Dim NewLine As String
NewLine = Chr(13) + Chr(10)
For i = 1 To 20
Text1.Text = Text1.Text + NewLine + _
    "вантуз № " + Str(i)
Next
```

Приклад виведення (присвоєння) в елемент ListBox (другий рядок):

```
For i = 1 To 20
List1.AddItem "вантуз № " + Str(i)
List1.ListIndex = List1.ListCount - 1
Next
```

Результати роботи додатку з фрагментами коду виведення в елементи TextBox та ListBox показані на рис. 5.14.



Рис. 5.14. Результати виведення в елементи TextBox та ListBox

Наведемо особливості роботи з елементами TextBox та ListBox.

В елементі керування TextBox можна відображати смуги скролінгу, встановивши відповідне значення властивості ScrollBar, для полегшення перегляду тієї частини тексту, яка не вміщається у вікні цього елемента. Якщо смуг скролінгу немає, то перегляд невидимої частини тексту здійснюється тільки клавішами переміщення курсора. Значення змінної NewLine в прикладі дозволяє організувати виведення послідовності символів в декілька рядків, але для цього властивість MultiLine повинна мати значення **True**. Оскільки елемент TextBox за замовчуванням дозволяє редагувати свій вміст, для заборони зміни відображеного тексту значення властивості Locked міняють на **True**. В елементі ListBox відображається значення List1.ListCount елементів типу String масиву List1.List, нумерація яких починається з 1. Додавання нового елемента у масив List здійснюється методом AddItem. Вертикальна смуга скролінгу з'являється автоматично в тому випадку, коли кількість елементів масиву перевищує кількість рядків вікна об'єкта ListBox. В цьому об'єкті дозволяється вибирати один елемент із списку. Вибране значення (номер елемента масиву List) зберігається у властивості ListIndex.

5.7. Робота з графікою

Відображення вмісту файлів з малюнками може відбуватися в таких об'єктах, як форма (Form), малюнок (**Image**), зображення (**PictureBox**), кнопка (Command). При цьому імені файлу з малюнком

Национальний університет
та природознавства

присвоюється властивість **Picture** цих об'єктів. Малювання у VB6 (проведення ліній Line, фігур Shape тощо) виконується тільки у формі або на об'єкті **PictureBox**, оскільки об'єкт **Image** не може бути контейнером для інших графічних елементів. Об'єкт **PictureBox**, як і форма, являє собою вікно, в якому можна розміщувати не тільки малюнки, а й інші елементи.

При виконанні робіт з графічними об'єктами потрібно вказувати місце, де вони розміщені. Для цього використовують екранні координати точки у вигляді пари чисел (X, Y). За замовчуванням координати верхнього лівого кута об'єкта (0, 0). Координата X збільшується вправо до максимального розміру об'єкта в заданих одиницях виміру, а координата Y збільшується вниз. У таблиці 5.4 наведені одиниці виміру, які встановлюються в залежності від значення властивості ScaleMode.

Таблиця 5.4.

Значення властивості ScaleMode

Значення	Константа	Одиниці виміру
0	vbUser	Визначаються користувачем
1	vbTwips	Твіпі (1/20 пункту або ~0.002 см)
2	vbPoints	Пункти (1/72 дюйма)
3	vbPixels	Пікселі
4	vbCharacters	Символи (120x240 твіпів)
5	vbInches	Дюйми (2,54 см)
6	vbMillimeters	Міліметри
7	vbCentimeters	Сантиметри

За замовчуванням використовується одиниця виміру твіп, яка є найбільш точною одиницею для екранних розмірів, оскільки не залежить від роздільної здатності екрана. Масштаб форми або зображення можна змінювати програмно, встановлюючи значення потрібних властивостей, чи методом **Scale**, що має наступний синтаксис:

[<об'єкт>.] Scale [(<X1>,<Y1>) – (<X2>,<Y2>)]

Тут (<X1>,<Y1>), (<X2>,<Y2>) координати лівого верхнього та правого нижнього кутів форми (зображення) відповідно. Виклик методу **Scale** без параметрів відновлює початковий масштаб вікна і повертає початок координат у лівий верхній кут. Для встановлення системи координат з початком відліку (0, 0) у центральній точці об'єкта з однаковою шириною і довжиною (**ScaleWidth** = 2a, **ScaleHeight** = 2a) записують метод **Scale** у наступному вигляді:

```
[<об'єкт>.]Scale (-a,a) – (a,-a)
```

Перехід з однієї системи координат в іншу здійснюється методами **ScaleX**, **ScaleY**, синтаксис яких:

```
[<об'єкт>.]ScaleX (<розмір X>, <система1>, <система2>)
```

```
[<об'єкт>.]ScaleY (<розмір Y>, <система1>, <система2>)
```

Першим параметром методів є розмір малюнка по горизонталі або вертикалі відповідно. Цей розмір буде перетворений із системи координат 1 у систему координат 2 (два останні параметри), які є значеннями з таблиці 5.4. В якості позначення першої системи координат використовують константу **HiMetric**, якщо розміри малюнка перевищують розміри об'єкта, в якому він розміщений.

Якщо потрібно замінити малюнок в об'єкті, застосовують функцію **LoadPicture**(<Файл>), яка дозволяє завантажити в елемент малюнок із вказаного файлу, наприклад:

```
Picture1.Picture= LoadPicture("Вантуз.bmp")
```

Якщо ця функція викликана без параметрів, відбудеться очистка елемента аналогічно виклику методу **Cls**.

Для збереження малюнка у файл на диск використовують наступний оператор:

```
SavePicture <малюнок>, <файл>
```

де малюнок – властивість **Picture** відповідного елемента. *Наприклад:*

```
SavePicture Picture1.Picture, pict.bmp
```

Для повернення числового значення кольору вказаного пікселя, використовують метод **Point**

```
[<об'єкт>.]Point (X, Y)
```

Щоб встановити вказаній точці потрібний колір, використовують метод:

```
[<об'єкт>.]Pset [Step] (X, Y), <колір>
```

Ключове слово **Step** задає режим роботи методів у відносних координатах (від початкової точки).

Малювання лінії здійснюється методом **Line**:

Национальний університет
водного господарства
та державного водного транспорту
Дніпропетровська області

[<об'єкт>.] **Line** [[**Step**] (X1,Y1)] - [**Step**] (X2,Y2) , [<колір>][, **B**[**F**]]
де (X1,Y1) - початкова точка лінії. Якщо вона не зазначена, лінія починається з поточної позиції. Параметр <колір> визначає колір лінії. Якщо він не заданий, для малювання використовується значення властивості форми **ForeColor**. Явний вибір значення кольору виконується з використанням функції **QBColor(i)**, де i- константа, що визначає значення кольору. Якщо вказана буква **B** (від Box - прямокутник), то замість лінії малюється прямокутник. Пари координат визначають розміщення його протилежних кутів. Прямокутник заповнюється кольором, зазначеним у **FillColor**, а тип заповнення визначається властивістю **FillStyle** форми. Якщо вказана буква **F**, прямокутник зафарбовується тим же кольором, що і границі. Параметр **F** використовується тільки разом з параметром **B**. Тип лінії встановлюється у властивості **DrawStyle**, а ширина лінії – **DrawWidth**.

Для малювання еліпсів та дуг використовують метод **Circle** наступної структури:

[<об'єкт>.] **Circle** [**Step**] (X, Y), <радіус>, [<колір>],
[<початок>], [<кінець>], [<відношення>]

Точка (X, Y) визначає центр фігури. Параметром <відношення> визначається відношення довжини вертикальної півосі до горизонтальної (при 1 – отримаємо коло, в інших випадках – еліпс). Параметри <початок> і <кінець> визначають кут початку і закінчення дуги. Кути задаються в радіанах і відраховуються проти годинникової стрілки. Якщо кути вказані із знаком мінус, буде зображено сектор.

5.8. Робота з мишею та клавіатурою

Події, які виникають при клацанні кнопкою миші, переміщенні миші, натисканні клавіші клавіатури, впливають на хід обчислювального процесу.

Програма, керована подіями (англ. *event-driven*), – програма, яка циклічно виконує певний код, очікуючи на деякі події та опрацьовуючи їх.

Коли відбуваються події, програма виконує одну або декілька процедур, які називають опрацьовувачами подій (англ. *event handlers*)



або процедурами опрацювання подій (англ. event procedures). Всі процедури опрацювання подій повинні мати наступну структуру опису:

Private Sub <ім'я об'єкта>_<ім'я події>([<параметри>])

Події відбуваються також при завантаженні форми (Load), активізації вікна форми (Activate), зміні розмірів (Resize) ініціалізації об'єкта (Initialise), зміні значень (Change), зміні розміщення повзунка смуги скролінгу (Scroll) тощо.

Розглянемо коротко деякі події миші:

- Click – виникає при клацанні на елементі;
- DblClick – виникає у випадку подвійного клацання;
- MouseDown – натиснута будь-яка кнопка;
- MouseUp – відпущена будь-яка натиснута кнопка;
- MouseMove – вказівник миші перемістився.

Перелічені події можуть розпізнаватися будь-яким елементом керування, що розміщений на формі, тільки в тому випадку, коли вказівник миші перебуває на ньому, а не на якомусь іншому елементі. В останні три процедури обробки подій миші передаються наступні параметри: **Button As Integer** – визначає стан кнопок миші; **Shift As Integer** – містить інформацію про натискання клавіш Alt, Ctrl, Shift; **X As Single, Y As Single** – координати вказівника миші.

При одночасному натисканні двох або більше клавіш значення параметрів додаються.

Джерелом (англ. source) називається той елемент, який переміщується. *Адресатом* (англ. target) називається об'єкт, в якому після переміщення буде розміщене джерело.

Таблиця 5.5.

Основні значення параметрів Button та Shift

Значення	Константа	Натиснута клавіша
параметр Button (миша)		
0		Немає
1	vbLeftButton	Ліва



2	vbRightButton	Права
4	vbMiddleButton	Середня
параметр Shift (клавіатура)		
0		Немає
1	vbShiftMask	Shift
2	vbCtrlMask	Ctrl
4	vbAltMask	Alt

Для переміщення об'єктів з допомогою миші застосовується технологія “перемістити і відпустити” (англ. Drag-and-Drop). Переміщення об'єктів під час роботи додатку автоматично не відбувається. Для цього в програмі необхідна наявність певних процедур опрацювання подій. Стандартні події, що виникають при переміщенні вказівника миші, та метод їх опрацювань наведені в таблиці 5.6.

Таблиця 5.6.
Події та метод переміщення

Назва	Опис
DragDrop	Подія, що виникає при завершенні операції переміщення вказівника миші
DragOver	Подія, яка виконується в тому випадку, коли елемент знаходиться над об'єктом-адресатом, причому операція переміщення ще не завершена
Drag	Метод, який використовується на початку та при закінченні операції переміщення (перед його використанням необхідно встановити DragMode = 0)

Подія **DragDrop** має наступні параметри: **Source As Control** – посилання на елемент-джерело; **X As Single, Y As Single** – координати точки об'єкта-адресата, в якій відпускається кнопка миші після завершення операції.



Метод **Drag** має один параметр – **Action**, значення якого: 0 - відміна операції; 1 - початок переміщення; 2 - завершення переміщення.

Натискання клавіш клавіатури викликає три події: **KeyDown** – виникає при натисканні клавіші, **KeyPress** – натиснута клавіша, **KeyUp** – клавіша відпускається. Вони мають першим параметром код натиснутої клавіші, процедури **KeyDown**, **KeyUp**, крім коду клавіші, мають другим параметром змінну **Shift**.

5.9. Середовище розробки програм Visual Basic 6.0



Рис. 5.15. Ярлик VB6

Запуск програми Microsoft Visual Basic 6.0 здійснюється командою пункту меню **Програми** головного меню Windows або з допомогою ярлика (рис. 5.15).

В результаті на екрані монітора відкриться діалогове вікно (рис. 5.16).



Рис. 5.16. Вікно нового проекту VB6

Воно містить три закладки:

- **Новый** – об'єднує шаблони для створення нового проекту;
- **Существующий** – дозволяє відкрити раніше створені проекти;



• **Файлы** — наведений перелік файлів, з якими відбувалася робота за даним комп'ютером.

Для створення нового стандартного додатку потрібно вибрати шаблон **Standard EXE**. Надалі будемо розглядати тільки русифіковану версію програми.

***Проект** – набір файлів, з яких складається додаток, що розробляється. Під проектом також розуміють сам додаток в процесі його створення.*

Проект VB6 складається з декількох компонент, які відображаються у вікні проекту. Вміст вікна проекту зберігається у файлі з розширенням VBR та містить список елементів, що завантажуються в інтегроване середовище розробки додатку. Основні файли та їх розширення наведені в таблиці 5.7.

Таблиця 5.7.

Склад проекту

Файл проекту	Розширення файлу
Файл проекту VB6	VBP
Група проекту	VBG
Форма	FRM
Модуль VB6	BAS
Довідковий файл Windows	HLP

Після вибору стандартного додатку відкриється інтегроване вікно середовища розробки додатку VB6 (рис. 5.17). В даному вікні можна проектувати та описувати складові частини додатку, редагувати програмний код, виконувати відлагодження додатків, компілювати розроблені додатки у виконуваний код. Інтегроване вікно містить наступні елементи:

- головне меню;
- панель інструментів **Standard**;
- панель елементів керування **ToolBox** – основний інструмент для візуальної розробки додатків;
- вікно конструктора форм (**Project1 – Form1 (Form)**) для підготовки форм з використанням елементів панелі ToolBox;

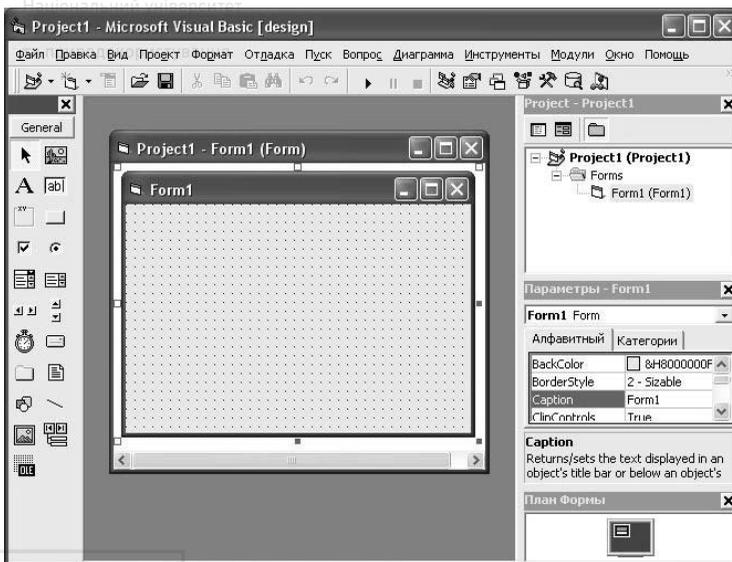


Рис. 5.17. Вікно середовища розробки нового проекту VB6

- вікно провідника проектів (**Project - Project1**);
- вікно властивостей активного об'єкта (**Параметри – Form1**);
- вікно розміщення форм (**План форми**) для планування розміщення вікон форм додатку на екрані монітора.

Для створення або редагування тексту програми додатку можна відкрити вікно редагування коду (**Код**) (рис. 5.18).

Стандартна панель інструментів містить наступні команди (в порядку розміщення зліва направо):

- відкриття нового стандартного проекту (Add Standard EXE Project);
- створення нової форми (Add Form);
- виклику редактора меню (Menu Editor);
- відкриття проекту (Open Project);
- збереження проекту (Save Project);
- виконання проекту (Start);
- завершення додатку (End);
- зупинки виконання додатку (Break);
- виведення провідника проекту (Project Explorer);
- відкриття вікна властивостей (Properties Window);
- відкриття вікна макету форм (Form Layout Window);



```
Form
Load
Else
Text3.Text = " <= 0"
End If
Select Case 5 * ss - 5
Case 0
Text3.Text = " = 0"
Case Is > 0
Text3.Text = " > 0"
Case Is < 0
Text3.Text = " < 0"
End Select
End Sub

Private Sub Form_Load()
End Sub
```

Рис. 5.18. Вікно розробки тексту програми

- відкриття вікна браузерa об'єктів (Object Browser);
- відкриття вікна панелі елементів керування (Toolbox);
- відкриття вікна властивостей активного об'єкта (Параметри – Form1);
- відкриття вікна перегляду даних (Data View Window);
- виклику менеджера візуальних компонентів (Visual Component Manager).

Панель елементів керування містить наступні кнопки: Вказівник (Pointer), Малюнок (PictureBox), Мітка (Label), Текст (TextBox), Рамка (Frame), Кнопка (CommandButton), Прапорець (CheckBox), Перемикач (OptionButton), Комбінований список (ComboBox), Список елементів (ListBox), Горизонтальна смуга скролінгу (HScrollBar), Вертикальна смуга скролінгу (VScrollBar), Таймер (Timer), Список дисків (DriveListBox), Список каталогів (DirListBox), Список файлів (FileListBox), Фігура (Shape), Лінія (Line), Зображення (Image), Дані (Data), OLE.

Вікно конструктора форм – основна робоча область, в якій проектується всі елементи віконного інтерфейсу.

Вікно властивостей містить значення властивостей елементів інтерфейсу користувача. В заголовку вікна вказується назва форми, в якій знаходиться даний об'єкт. Нижче – список об'єктів форми, з якого



вибирається необхідний об'єкт. Ще нижче – два стовпці з рядками властивостей того чи іншого об'єкта (набір властивостей залежить від типу об'єкта), які розміщені в алфавітному порядку (Alphabetic) або за категоріями (Categorized). Перший стовпець містить назви властивостей, а другий – значення властивостей. Значення властивостей вводяться вручну або вибираються із списку. *Наприклад*, властивість **Caption** визначає заголовок або надпис на елементі; **Picture** – назва файлу, в якому міститься малюнок для виведення в елемент.

Для кожного елемента керування існують методи, що виконують певні дії. *Наприклад*, методи форм: **Cls** - очистка форми; **Print** - виведення тексту на форму.

В операційній системі Windows розробка додатків складається з наступних етапів: розробка екранної форми з елементами керування, написання процедур обробки подій, які виникають в процесі роботи програми, написання підпрограм реалізації алгоритму розв'язку задачі.

Пояснює процес розробки додатку на прикладі 1 (див. пункт 5.1). Спочатку завантажимо VB6, у вікні нового проекту (рис. 5.16) вибираємо стандартний проект та натискаємо кнопку **Открыть**. Після цього з'явиться вікно **Project1 – Form1 (Form)** з порожнім макетом форми, в якому розміщуються потрібні елементи керування. Для розв'язання поставленої задачі використовуються елементи **Label**, **TextBox**, **Command**, які розмістимо на макеті форми (рис. 5.19).

Кожен елемент, після розміщення його на формі, автоматично дістає назву, яка складається з імені відповідного елемента керування та його порядкового номера, наприклад, Label1. Ця назва автоматично присвоюється властивості (Name). При необхідності її можна змінити. Назва елемента використовується в програмі при звертанні до нього, наприклад: Text1.Text – властивість Text елемента Text1.

Для кожного поставленого на форму об'єкта заповнюємо деякі поля обов'язкових властивостей, які наведені в таблиці 5.8.

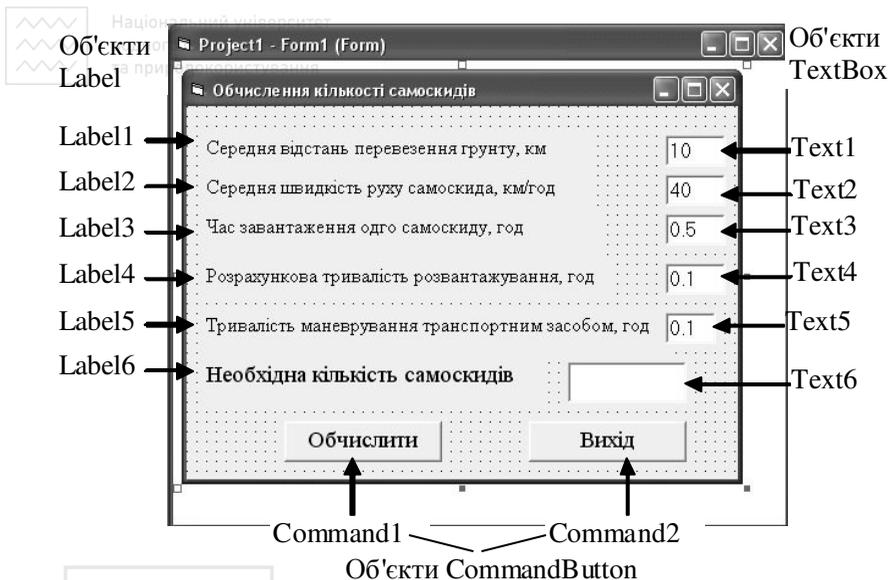


Рис. 5.19. Макет форми

Таблиця 5.8.

Властивості елементів додатку

Назва елемента	Властивість	Значення властивості
Form1	Caption	Обчислення кількості самоскидів
	Font	Times New Roman, 12
Label1	Caption	Середня відстань перевезення ґрунту, км
	Font	Times New Roman, 12
Label2	Caption	Середня швидкість руху самоскида, км/год
	Font	Times New Roman, 12
Label3	Caption	Час завантаження одного самоскида, год
	Font	Times New Roman, 12
Label4	Caption	Розрахункова тривалість розвантажування, год
	Font	Times New Roman, 12



Label5	Caption	Тривалість маневрування транспортним засобом, год
	Font	Times New Roman, 12
Label6	Caption	Необхідна кількість самоскидів
	Font	Times New Roman, 14
Text1	Text	10
	Font	Arial, 12
Text2	Text	40
	Font	Arial, 12
Text3	Text	0.5
	Font	Arial, 12
Text4	Text	0.1
	Font	Arial, 12
Text5	Text	0.1
	Font	Arial, 12
Text6	Text	
	Locked	True
	Font	Arial, 14
Command1	Caption	Обчислити
	Font	Times New Roman, 14
Command2	Caption	Вихід
	Font	Times New Roman, 14

Об'єкти Text1 - Text5 використовуються для введення значень параметрів формули обчислення, Text6 використовується для відображення результату обчислень. Кнопка Command1 (Обчислення) призначена для виклику процедури обчислення та виведення результату, Command2 (Вихід) призначена для виходу з програми та завершення роботи додатку. Для цих кнопок необхідно написати програмний код. Найкраще це зробити, двічі клацнувши на відповідній кнопці лівою клавішею миші (вікно для написання програмного коду також ще можна відкрити командою **Код** з пункту меню **Вид** або командою **Показати код** контекстного меню елемента). Двічі клацаємо мишкою на об'єкті Command1. У вікні, що розгорнеться, запишемо програмний код процедури обробки події Command1_Click() наступним чином:

```

Private Sub Command1_Click()
Dim L As Single, Vsr As Single
Dim Tz As Single, Tp As Single
Dim Tm As Single, T As Single
Dim Nc As Single
L = Val(Text1.Text)
Vsr = Val(Text2.Text)
Tz = Val(Text3.Text)
Tp = Val(Text4.Text)
Tm = Val(Text5.Text)
T = 2 * L / Vsr
Nc = (T + Tz + Tp + Tm) / Tz
Text6.Text = Format(Nc, "#####.0")
End Sub

```

В даному тексті процедури кожен оператор розміщений в окремому рядку. Якщо необхідно записати два оператори в одному рядку програми, то вони розділяються двокрапкою. Для створення програмного коду для кнопки Command2 двічі клацаємо на ній лівою клавiшею миші та після заголовку процедури Command2_Click() вставляємо рядок з оператором **End**. Маємо:

```

Private Sub Command2_Click()
End
End Sub

```

Перед виконанням розробленого проекту збережемо його на носії інформації. Виконання розробленого додатку здійснюється з допомогою інструменту **Старт** або клавiші F5 (пункт меню **Пуск**, команда **Старт**). В результаті виконання отримаємо екранну форму додатку, зображену на рис. 5.20.

Середовище розробки додатків VB6 дозволяє автоматизувати набір програми шляхом вибору із списку потрібних елементів. Список з'являється при наборі деяких операторів та властивостей елементів (рис. 5.21).

Пошук потрібного елемента здійснюється клавiшами керування курсором або шляхом послідовного набору символів назви властивості. Вибір із списку здійснюється клавiшею **Tab** або **Ctrl+Enter**.



Середня відстань перевезення ґрунту, км	10
Середня швидкість руху самоскида, км/год	40
Час завантаження одго самоскиду, год	0.5
Розрахункова тривалість розвантажування, год	0.1
Тривалість маневрування транспортним засобом, год	0.1
Необхідна кількість самоскидів	2,4

Обчислити Вихід

Рис. 5.20. Вікно роботи додатку визначення кількості самоскидів

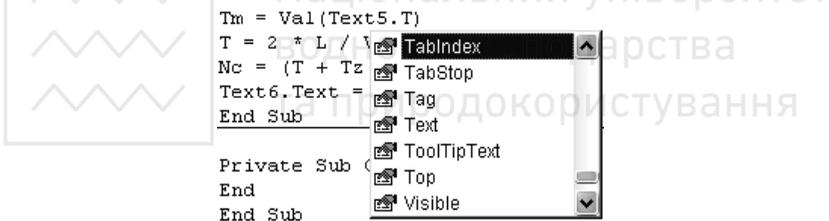


Рис. 5.21. Список для елемента TextBox

5.10. Програмування деяких алгоритмів

5.10.1. Табулювання значень функції

Задача табулювання значень функції передбачає обчислення в конкретних точках на вказаному відрізку значень функції та виведенні таблиці значень аргументу і функції. Розглянемо реалізацію алгоритму розв'язування задачі (див. приклад 3 пункту 5.1) засобами Visual Basic.

Спроекуємо екранну форму (рис. 5.22). Змінені властивості об'єктів форми наведено в таблиці 5.9.



Рис. 5.22. Проект форми задачі табулювання значень функції

Таблиця 5.9.

Властивості елементів форми

Назва елемента	Властивість	Значення властивості
Form1	Caption	Швидкість руху рідини в трубі
Label1	Caption	Значення I
Label2	Caption	Значення n
Label3	Caption	Значення R0
Label4	Caption	Значення Rk
Label5	Caption	Значення dR
Text1	Text	0.003
Text2	Text	0.014
Text3	Text	0.7
Text4	Text	0.86



Text5	Text	0.02
Command1	Caption	Обчислити
Command2	Caption	Вихід

Значення властивості Font для всіх перелічених в таблиці елементів встановлено: Arial, Ж, 14.

Алгоритм табулювання значень функції реалізовано у процедурі Command1_Click(), що відповідає кнопці **Обчислити**.

```
Private Sub  
Dim R As Single, R0 As Single  
Dim Rk As Single, dR As Single  
Dim I As Single, n As Single  
Dim v As Single  
Cls  
I = Val(Text1.Text)  
n = Val(Text2.Text)  
R0 = Val(Text3.Text)  
Rk = Val(Text4.Text)  
dR = Val(Text5.Text)  
R = R0  
Print " R", " v"  
Do  
v = R ^ 0.67 * I ^ 0.5 / n  
Print R, Format(v, "0.0000")  
R = R + dR  
Loop While R < Rk  
End Sub
```

Текст процедури для кнопки **Вихід** повністю відповідає попередньому прикладу. Виведення результатів обчислень реалізовано у форму (рис. 5.23).



R	v	Значення l	Значення n	Значення R0	Значення Rk	Значення dR
0.7	3.0807	0.003				
0.72	3.1394		0.014			
0.7399999	3.1976					
0.7599999	3.2552					
0.7799999	3.3123					
0.7999999	3.3690			0.7		
0.8199999	3.4252				0.86	
0.8399999	3.4810					0.02
0.8599999	3.5363					

Обчислити

Вихід

Рис. 5.23. Результати табулювання функції $v(R)$

5.10.2. Пошук мінімального (максимального) елемента масиву

Пошук максимального або мінімального елемента є досить поширеним завданням в багатьох задачах. Алгоритм пошуку максимуму відрізняється від алгоритму пошуку мінімуму, як правило, лише заміною знаку $>$ на $<$ або навпаки. Розглянемо один із способів відшукування мінімуму на прикладі.

Приклад. Знайти найменшу денну витрату води річкою в березні і номер дня, в якому була ця витрата.

Для розв'язання задачі денні витрати води розмістимо у масиві дійсних чисел W_{ut} , який складається з 31 елемента. Номер елемента масиву відповідає номеру дня. Значення мінімальної витрати розмістимо у змінній $MinW$, а день цієї витрати позначимо змінною dnW .

Запишемо словесний алгоритм розв'язування задачі.

1. Для i від 1 до 31 виконати:
ввести значення витрати для i -го дня.
2. $MinW := W_{ut}(1)$.
3. $dnW := 1$.
4. Для i від 2 до 31 виконати
 - 4.1. Якщо $W_{ut}(i) < MinW$ то
 - 4.1.1. $MinW := W_{ut}(i)$
 - 4.1.2. $dnW := i$
5. Вивести значення $MinW$, dnW .



Для розв'язування задачі спроектуюмо екранну форму (рис. 5.24 а)).
Змінені властивості об'єктів форми наведено в таблиці 5.10.

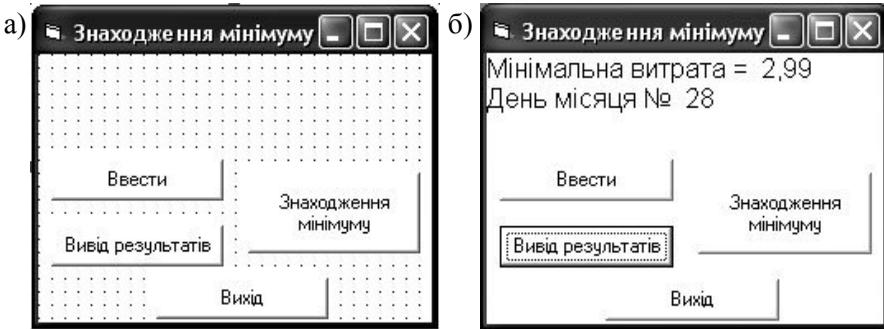


Рис. 5.24. Проект форми а) та результат роботи додатку б)

Таблиця 5.10.

Властивості елементів форми

Назва елемента	Властивість	Значення властивості
Form1	Caption	Знаходження мінімуму
Command1	Caption	Ввести
Command2	Caption	Вихід
Command3	Caption	Знаходження мінімуму
Command4	Caption	Вивід результатів

Алгоритм відшукування мінімального елемента масиву та його номера реалізовано у процедурі Command3_Click(), що відповідає кнопці **Знаходження мінімуму**. Програмний код розв'язування задачі наступний:

```
Dim Wut(1 To 31) As Single
Dim MinW As Single
Dim dnW As Byte

Private Sub Command1_Click()
Dim i As Byte
For i = 1 To 31
```



```
Wut(i) = Val(TextBox(Str(i) + _  
"-й день", "Введення витрати", 0))
```

```
Next i  
End Sub
```

```
Private Sub Command2_Click()  
End  
End Sub
```

```
Private Sub Command3_Click()  
Dim i As Byte  
MinW = Wut(1)  
dnW = 1  
For i = 2 To 31  
    If Wut(i) < MinW Then  
        MinW = Wut(i)  
        dnW = i
```

```
    End If  
Next i  
End Sub  
Private Sub Command4_Click()  
Cls  
Print "Мінімальна витрата = "; MinW  
Print "День місяця № "; dnW  
End Sub
```

Результати виконання програмного коду показані на рис. 5.24 б).

5.10.3. Обчислення суми (добутку) елементів масиву

Алгоритм обчислення суми елементів масиву наступний:

1. Змінній, що позначає суму, присвоїти значення нуль.
2. В циклі від початкового до кінцевого елемента суми виконати: значення суми збільшити на значення елемента.

Алгоритм обчислення добутку відрізняється від алгоритму обчислення суми тим, що змінній спочатку присвоюється значення 1, а в другому пункті замість знаку плюс записують знак множення.

Приклад. Знайти загальну суму опадів за 5 місяців, декадні значення яких розміщені в масиві `Opad`, що містить 5 рядків та 3 стовпці.

Фрагмент програмного коду додатку, що реалізує обчислення суми, запишемо у вигляді окремої функції.

```

Dim Opad(1 To 5, 1 To 3) As Single
Private Function Suma() As Single
Dim sum As Single
Dim i As Byte, j As Byte
sum = 0
For i = 1 To 5
    For j = 1 To 3
        sum = sum + Opad(i, j)
    Next j
Next i
Suma = sum
End Function

```

5.11. Основні поняття об'єктно-орієнтованого програмування

В результаті розвитку технологій програмування складних задач виник новий напрямок у розробці програм, в основу якого покладено поняття об'єкта. Суть об'єктно-орієнтованого підходу до програмування задач полягає в тому, що всі елементи додатку вважаються об'єктами, які взаємодіють з користувачем та між собою. Окрім того, в програму переносяться вже визначені у предметній області функціональні моделі. Структура моделей дозволяє багаторазове використання їх в інших задачах чи в інших предметних областях.

Об'єкт (англ. *object*) – деяка окрема сутність, яка має свої властивості, поведінку, взаємодію. Іншими словами, об'єкт – програмна одиниця, яка складається із властивостей та методів.

Об'єкти мають дві характеристики – стан та поведінку. Програмні об'єкти зберігають свій стан у **властивостях**. Загалом, об'єкти характеризуються наступними основними поняттями.

Стан – кожен об'єкт у будь-який момент часу перебуває в деякому стані, що характеризується набором властивостей.

Властивість (англ. *property*) – окрема якість (параметр) об'єкта (назва, розміри, місцезнаходження тощо). Сукупність властивостей об'єкта визначає його стан.

Метод (англ. *method*) – функція або процедура, яка реалізує певну поведінку об'єкта.

Подія (англ. *event*) – деяка дія, яка є засобом взаємодії між об'єктами. Об'єкти генерують задані події і виконують визначені дії як реакції на події, що відбуваються. Подія – аналог повідомлень, які відправляють чи отримують об'єкти.

Подіями, як уже відмічалось, можуть бути клацання мишею, завантаження форми, настання певного часу системного годинника, тощо. Події ініціюються: діями користувача; повідомленнями, що надходять від інших додатків; активним додатком.

В операційній системі Windows розробка програм полягає як в написанні процедур обробки подій, які виникають в процесі роботи програми, так і в написанні підпрограм реалізації алгоритму розв'язування задачі.

Об'єкти, що мають загальні властивості та поведінку, об'єднуються в класи.

Клас (англ. *class*) – окремий тип даних, що має єдину структуру та об'єднує об'єкти із спільними властивостями та методами.

Екземпляр класу (англ. *instance*) – конкретний об'єкт, що має структуру класу.

Таким чином, кожен об'єкт класу має всі властивості, методи та події класу. Основна характеристика класу – можливість створення на його основі нових класів. Тому всі класи утворюють **ієрархічну** систему і можуть бути дочірніми стосовно декількох класів, що розміщені вище.

Клас, який не має свого попередника, називається базовим (англ. *base class, superclass*).

Новий клас, створений на основі базового, **наслідую** всі його характеристики.

Всі події для об'єктів класу визначаються на етапі його проектування. Для всіх класів існує дві події: **подія створення об'єкта класу** (англ. Initialize) та **подія знищення об'єкта класу** (англ. Terminate).

Колекція (англ. *collection*) – набір об'єктів (не обов'язково одного класу), що має спільне ім'я.



Для створення користувацьких колекцій використовують тип даних **Collection**. До кожної користувацької колекції автоматично включаються наступні методи та властивості:

- метод **Add** – додає об'єкт до колекції;
- метод **Remove** – вилучає об'єкт з колекції;
- метод **Item** – повертає об'єкт з колекції;
- властивість **Count** – повертає загальну кількість об'єктів у колекції.

Технології сучасних систем програмування об'єднані в RAD-системах. Ці системи містять безліч готових класів у вигляді візуальних компонент, що вставляються у програму клацанням кнопок миші. Розробник програми повинен тільки спроектувати загальний вигляд вікон свого додатку, визначити обробку основних подій (які оператори будуть опрацьовуватися при натисканні на кнопки, виборі пунктів меню та клацанні кнопок миші). Програміст у цьому випадку не витрачає час на розробку допоміжного програмного коду, а зосереджується повністю на реалізації алгоритму.

Контрольні запитання і завдання

28. Назвіть етапи розв'язування задачі.
29. Що називається алгоритмом та які його властивості? Які існують стандартні блоки для опису алгоритмів? Які є структури алгоритмів?
30. Що називається циклом? Які є типи циклів?
31. Які є основні елементи мови програмування VB6?
32. Яким чином описуються змінні у VB6? Які існують області видимості змінних VB6?
33. Як описуються масиви у VB6? Яким чином змінюються розмірності масивів? Яка структура та призначення оператора опису користувацьких типів?
34. Що називається виразом та які правила запису виразів? Які основні операції використовуються у мові програмування VB6?
35. Які існують основні оператори VB6? Як отримати результати виконання оператора відразу після його запису?
36. Які є основні вбудовані функції VB6?
37. Яка структура та призначення оператора присвоєння?
38. Яка структура та призначення умовних операторів?
39. Яка структура та призначення операторів циклу?

40. Для яких цілей використовуються оператори End та Exit?
41. Яке призначення та структура опису вікна повідомлень? Яке призначення та структура опису вікна введення інформації?
42. Як застосовуються файли довільного доступу?
43. Як застосовуються текстові файли?
44. Що називається процедурою? Які існують процедури?
45. Яка структура та призначення оператора виклику процедури? Який вигляд має опис підпрограм та функцій?
46. Яким чином передаються значення змінних у процедури?
47. Яким чином здійснюється виведення інформації на форму?
48. Як здійснюється форматне виведення інформації?
49. Які є одиниці виміру екранних координат? Якими методами здійснюється малювання основних геометричних фігур?
50. Які виникають події при роботі з мишкою?
51. Яким чином здійснити переміщення об'єктів на формі?
52. Який склад проекту VB6?
53. Перелічіть основні елементи керування VB6 та їх властивості.
54. Запишіть процедуру табулювання значень функції $y = \sin(x)$.
55. Запишіть алгоритм та програму реалізації пошуку максимального елемента масиву.
56. Знайдіть добуток десяти перших елементів масиву.
57. Яка суть об'єктно-орієнтованого програмування? Дайте визначення наступним поняттям: об'єкт, стан, властивість, метод, подія, клас, колекція. Які є методи колекцій?



РОЗДІЛ 6. КОМП'ЮТЕРНІ МЕРЕЖІ

6.1. Історія створення та склад мережі

Експерименти з передачі даних між комп'ютерами розпочалися у 50-х роках XX століття. Першими прикладами комп'ютерних мереж можна вважати багатокористувацькі системи на базі суперЕОМ, в яких до центрального комп'ютера (консолі) підключалися термінали (як правило, дисплей з клавіатурою). У 1960 р. був розроблений перший модем, а в 1962 році описана можливість створення мережі, яка б забезпечувала доступ до віддалених комп'ютерів. Це сприяло тому, що у 1965 році була створена перша мережа з комутацією каналів з використанням звичайних телефонних ліній. Ефективність такого зв'язку була низькою, тому в 1969 році була створена нова мережа ARPANET, в якій використовувалася комутація пакетів. Операційна система Unix стала першою системою, яка підтримувала роботу мережі на базі набору утиліт UUCP для зв'язку між віддаленими вузлами. Потім ці утиліти лягли в основу організації служб новин мережі Internet. Першою окремою програмою для мережі ARPANET стала програма роботи з електронною поштою, розроблена у 1972 р.

Окрім згаданої мережі, розвивалися й інші типи мереж, наприклад: CSNET, BITNET, JANET, NSFNET, а також розроблялися нові протоколи стандартизації передавання інформації. В 1981 році підписано договір, який дозволяв використовувати мережу ARPANET без обмежень. У 1983 році з'явився перший стандарт для протоколів TCP та IP відповідно транспортного та мережевого рівнів, що надалі забезпечило можливість створення сучасних шлюзів та маршрутизаторів. Поступово ці протоколи витіснили в мережах майже всі інші протоколи. Згодом мережі різних типів почали об'єднуватися. Об'єднання декількох локальних мереж англійською мовою стали називати internet (дослівно – мережа мереж).

Все це, а також розвиток апаратного і програмного забезпечення, розробка та прийняття нових стандартів сприяло виникненню у 1989 р. мережі Internet, яка об'єднала більшість мереж в усьому світі. Кількість комп'ютерів у мережі обчислюється мільярдами. Можна сказати, що нині без мережевих технологій не може існувати економіка жодної країни.



Наведемо основні визначення, що стосуються комп'ютерних мереж.

Комп'ютерна мережа – група комп'ютерів, між якими встановлений зв'язок за допомогою спеціальних апаратних та програмних засобів.

З'єднання двох комп'ютерів для обміну даними без використання додаткових пристроїв називають **прямим з'єднанням**.

Топологія мережі – структурна схема зв'язків між основними функціональними елементами мережі.

Вузол (англ. station, node, site) – окремий пристрій в мережі, який має мережеву адресу (хост, комп'ютер, файл-сервер, принтер тощо).

Сервер (англ. server) – комп'ютер у мережі, ресурси якого використовуються іншими комп'ютерами.

Шлюз (gateway) – комп'ютер або порт колективного доступу, який об'єднує декілька мереж або який використовують для під'єднання мережі до головного комп'ютера.

Маршрутизатор або **роутер** (англ. router) – апаратно-програмний пристрій, який на підставі інформації про топологію мережі та певних правил приймає рішення про пересилання пакетів між різними частинами мережі.

Хост (англ. host - господар) – комп'ютер, що працює у локальній мережі з використанням протокольного стеку TCP/IP.

Робоча станція (англ. workstation) – комп'ютер в мережі, який використовує ресурси сервера (інша назва – клієнт).

Середовище передавання даних – сукупність ліній передавання даних та іншого обладнання, яке забезпечує обмін даними між комп'ютерами відправника та отримувача інформації.

Каналом передавання даних називають сукупність програмно-технічних засобів та фізичного середовища передавання сигналу даних.

Кодування – певне перетворення даних для збільшення завадостійкості, швидкості та зручності передавання.



Бурхливий розвиток інформатизації та необхідність обміну й обробки великих обсягів даних зумовило об'єднання в мережу багатьох різноманітних пристроїв. Тепер вона складається з терміналів, вузлів та іншого мережевого обладнання (рис. 6.1). В якості терміналів можуть використовуватися персональні комп'ютери, пристрої PDA (Personal Digital Assistant), пейджери, мобільні телефони, побутові прилади та ін. У мережах прикладні процеси зв'язуються між собою через середовище передавання даних каналами зв'язку.

Для фізичної організації мереж використовують мережеві адаптери (карти), маршрутизатори, комутатори, концентратори та інші пристрої. З допомогою цих засобів можна створити наступні основні типи топологій мереж: **загальна шина, кільце, зірка**. Топологія із загальною шиною передбачає з'єднання комп'ютерів між собою через загальну шину (магістраль). На кінці шини встановлюють спеціальні термінатори. При кільцевій топології шина утворює кільцеву магістраль (тобто її кінці з'єднують).

Зіркоподібна топологія забезпечує обмін інформацією між вузлами через концентратори або комутатори.

Концентратор (хаб, англ. *hub*) – пристрій для з'єднання між собою багатьох комп'ютерів через порти (від 4 до 24). Один порт передбачений для з'єднання з іншим хабом або комутатором. В

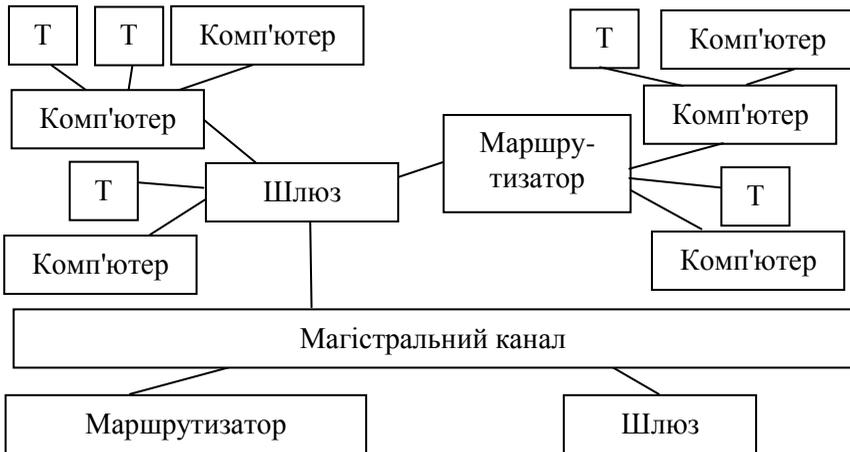


Рис. 6.1. Схема сучасних комп'ютерних мереж (Т – термінал)

концентраторі сигнал від одного комп'ютера підсилюється та передається на всі порти.

Комутатор (свіч, англ. *switch*) – пристрій, конструктивно виконаний у вигляді мережевого концентратора, що працює як високошвидкісний багатопортовий міст. Вбудований в нього механізм комутації дає змогу виконувати сегментування локальної мережі та виділяти смугу пропускання кінцевим станціям мережі. Комутатор містить процесор та внутрішню пам'ять для зберігання таблиць адрес.

Комутатори спрямовують сигнал від вузла тільки в той порт, до якого приєднаний потрібний комп'ютер, за допомогою **MAC-адресації**. В кожному комутаторі створюється таблиця відповідності адрес пристроїв та портів, до яких вони приєднані.

MAC-адреса (*MAC – Media Access Control*) – апаратна адреса мережевого пристрою (мережева карта, свіч, хаб тощо) у вигляді шестибайтного числа (48 біт) яка записана в постійній пам'яті пристрою. Ця адреса призначається виробником та однозначно ідентифікує будь-який пристрій в мережі.

Обмін даними між вузлами може відбуватися кабелями або без них. Передавання даних без використання кабелів називається передаванням в ефірних середовищах. Залежно від частоти, ефірні середовища поділяють на радіоканали, ультракороткохвильові, інфрачервоні, мікрохвильові, лазерні.

Кабельне з'єднання комп'ютерів може здійснюватися за допомогою коаксіального або волоконно-оптичного кабелю, скрученої пари. Найпоширеніше та найдешевше на даний момент з'єднання вузлів із комутатором чи концентратором з допомогою скрученої пари. Скручена пара – вісім ізольованих мідних дротів, скручених попарно в спільній ізоляції та з можливим екрануванням. На кінцях скрученої пари встановлюються конектори RJ-45, які вставляються в мережеві пристрої. Найпоширеніша скручена пара 5 категорії дозволяє організувати 100 мегабітний канал передачі даних. Сигнали у скрученій парі можуть передаватися без підсилення на сотні метрів.

У волоконно-оптичних кабелях використовують прозоре скловолокно в оболонці, що відбиває світло. Сучасні волоконно-оптичні системи дозволяють передавати сигнал із швидкостями порядку терабіт за секунду на відстань декількох тисяч кілометрів.

При підключенні окремих комп'ютерів до глобальних мереж можуть використовуватися модеми з комутованим доступом, які

забезпечують зв'язок телефонними лініями. Цей тип підключення найчастіше використовується для з'єднання з постачальником послуг Internet при відсутності інших можливостей.

Якщо немає можливості створити кабельну мережу, використовують безкабельні типи зв'язку. Один із таких типів використовує радіочастоти та радіомодеми. Кожен радіомодем має антену та передавач для направленої передавання сигналу.

Для супутникового зв'язку використовують геостационарні (технологія VSAT) або низькоорбітальні супутники (наприклад, система Iridium). Для цього у клієнта встановлюється еліптична антена.

Наступний сучасний тип зв'язку базується на можливостях інфраструктури мобільної телефонії та системи Bluetooth. Мережа з Bluetooth використовує неліцензований частотний діапазон малого радіусу дії (до 100 метрів) та низької завадостійкості. Для забезпечення доступу до Internet через мобільні телефони у 1998 р. розроблена технологія WAP. Модель WAP сумісна з моделлю Web, для неї існує свій мікробраузер.

В даний час у мережі Internet використовуються практично усі відомі лінії зв'язку від низькошвидкісних телефонних ліній до високошвидкісних цифрових супутникових каналів. Операційні системи, які використовуються в мережі Internet, також відрізняються розмаїтістю: від Unix до мобільних ОС.

6.2. Призначення та принципи побудови комп'ютерних мереж

В загальному випадку для створення комп'ютерної мережі необхідне спеціальне апаратне забезпечення (мережеве обладнання) та спеціальне програмне забезпечення (мережеві програмні засоби).

Комп'ютерні мережі призначені для:

- обміну інформацією;
- розподілення обчислень;
- розподіленого зберігання інформації (файлів);
- розподіленого використання периферійних пристроїв;
- централізованого керування;
- забезпечення доступу до інших мереж.

Застосування мереж в установах дозволяє зменшити матеріальні затрати на апаратне та програмне забезпечення без зниження його продуктивності. Сучасні програмні продукти, за своєю суттю, є

багатокористувацькими. Вони дозволяють працювати за різними комп'ютерами з одним інформаційним об'єктом, наприклад, базою даних, одночасно контролюючи та захищаючи дані.

У 80-х роках ХХ століття набуло поширення встановлення зв'язків між комп'ютерами в межах одного підприємства, що сприяло виникненню локальних мереж. Бурхливому розвитку локальних мереж сприяли поява персональних комп'ютерів, здешевлення мікропроцесорної техніки та розвиток програмного забезпечення. Зараз понад 80 % інформаційних потоків зосереджено в локальних мережах. Найчастіше підключення комп'ютерів до локальної мережі здійснюється з використанням технології Ethernet, яка потребує наявності в комп'ютері мережевої карти відповідного стандарту.

Комп'ютерні мережі класифікують за наступними ознаками:

- географічним розташуванням – локальні, регіональні, глобальні;
- сферою застосування – побутові, офісні, промислові;
- комплексом архітектурних вирішень – Ethernet, Token Ring, Arcnet;
- топологією мережі – шинна, кільцева, зіркоподібна, деревоподібна, повнозв'язна;
- фізичним середовищем передавання – мережа з волоконно-оптичним, коаксіальним кабелем, інфрачервоним каналом, скрученою парою;
- методом доступу до фізичного середовища – мережі з опитуванням, суперництвом, маркерним доступом;
- набором протоколів – мережі TCP/IP, SPX/IPX.

Основою класифікації комп'ютерних мереж є комплекс їх архітектурних особливостей. Наведемо основні класи мережевих вирішень.

Шина комп'ютера, інтерфейсні комп'ютерні засоби (USB, Firewire, COM-порти) призначені для приєднання до комп'ютера периферійних пристроїв. Передача даних здійснюється на короткі відстані (до десятків метрів) із швидкістю до сотень мегабіт за секунду.

Малі локальні та побутові мережі (SOHO – Small Office Home networks) створюються в межах одного офісу, будинку. Вони забезпечують передачу даних на сотні метрів із швидкістю до 1 гігабіта за секунду.

Локальні мережі (LAN – Local Area Network) забезпечують передачу інформації в межах одного або декількох близько

розташованих будинків із швидкістю кілька сотень мегабіт за секунду. Локальні мережі забезпечуються технологіями Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring. Вартість пристроїв комунікації в цих мережах значно нижча, ніж вартість іншого обладнання. На базі локальних мереж можна створювати **розподілену інформаційну систему**, в якій паралельно опрацьовується інформація та спільно використовуються ресурси.

Іноді в окремий клас виділяють **кампусні** (англ. campus – студентське містечко) **мережі**, що об'єднують недалеко розташовані будинки. Ці мережі можуть мати специфічні технологічні вирішення.

Корпоративні мережі – об'єднання локальних мереж за допомогою каналів зв'язку в межах однієї великої організації (компанії).

Глобальні мережі (WAN – Wide Area Network) не обмежуються окремою територією. Швидкість передачі інформації в магістральних каналах тут досягає терабіт за секунду. Власниками каналів зв'язку глобальних мереж є провайдери, які надають платні послуги.

Регіональні мережі можуть ґрунтуватися на технологіях локальних або глобальних мереж і розгортаються в межах населеного пункту, району, області.

В локальній мережі комп'ютери можуть мати різне функціональне призначення. В одноранговій LAN відсутній виділений сервер, а кожна робоча станція може виконувати функції як клієнта, так і сервера. У серверних мережах (мережі з виділеним сервером) існує чіткий розподіл функцій: одні комп'ютери постійно є серверами, інші клієнтами. Розрізняють наступні різновиди серверів: файловий сервер, сервер друку, поштовий сервер, Web-сервер, DNS-сервер.

Оскільки комп'ютерна мережа – складна система, то для її нормальної роботи всі складові частини повинні працювати за однаковими правилами (стандартами). Стандарти можуть бути міжнародними, регіональними (в межах однієї країни) тощо. Існують стандарти Міжнародного телекомунікаційного союзу (ITU), Міжнародної організації зі стандартизації (ISO). Для опрацювання інформації на EOM при ISO створено технічний комітет 97, стандарти якого мають формат NNNN ISO.

Відкрита система – це система, яка побудована та діє відповідно до вимог міжнародних стандартів.

Якщо користувач підключений до мережі Internet, він використовує відкриту систему на своєму комп'ютері. У системах, що функціонують відповідно до міжнародного стандарту 7498 ISO, середовище передавання даних розподілено на сім рівнів: 7 –

Application (Прикладний); 6 – Presentation (Відображення); 5 – Session (Сеансовий); 4 – Transport (Транспортний); 3 – Network (Мережевий); 2 – Data Link (Канальний); 1 – Physical Link (Фізичний). На базі цих рівнів будується модель взаємодії відкритих систем ISO/OSI (Model of Open System Interconnections – OSI). Зв'язок між рівнями відбувається за допомогою протоколів.

Протокол – набір синтаксичних та семантичних правил, який визначає поведінку об'єктів під час їх взаємодії.

Рівні 7, 6, 5 орієнтовані на обслуговування прикладних процесів (програм користувачів), а рівні з четвертого по перший – на мережу (на надійне передавання даних). Кожен рівень комп'ютера-відправника взаємодіє з відповідним рівнем комп'ютера-отримувача через віртуальні з'єднання. Фізичне з'єднання існує тільки на першому рівні. Віртуальні з'єднання є основою роботи в Інтернеті та встановлюються між однаковими рівнями різних комп'ютерів. Кадр, що передається від відправника до адресата, може проходити через сотні транзитних комп'ютерів, при цьому піднімаючись з 1 до 3 рівня, де визначаються адреси, а потім знову переходити до нижчого рівня для передавання на інший комп'ютер.

На кожному рівні для передавання даних можуть використовуватися декілька протоколів, кожен з яких виконує свої функції. Зокрема, на прикладному рівні використовують наступні протоколи: керування терміналами, керування діалогом, керування файлами, керування задачами, керування системою, забезпечення цілісності даних. А на рівні відображення використовують протоколи вибору потрібної форми відображення даних, перетворення даних (кодування), шифрування інформації тощо.

Суміжні рівні однієї системи взаємодіють через спільну межу, яку називають *інтерфейсом*. В комп'ютері відправника з вищого рівня до нижчого послідовно передається блок даних, який називається інтерфейсним. Кожен рівень додає до інтерфейсного блоку службову інформацію керування і таким чином формує протокольний блок даних даного рівня. Протокольні блоки даних на транспортному рівні діляться на однакові частини та нумеруються. Ці частини називають *пакетами*. На транспортному рівні прокладають маршрути доставки пакетів від відправника до отримувача, тобто здійснюється маршрутизація залежно від адреси призначення, завантаженості каналів зв'язку, їх надійності тощо. Адресою на мережевому рівні є IP-адреса (див. розділ 6.4), а на канальному рівні – MAC-адреса. Для прокладання маршрутів використовують маршрутизатори.

Пакет на каналному рівні доповнюється даними MAC-адрес відправника та отримувача, причому пакети можуть об'єднуватися або розрізатися на частини. Блоки даних каналного рівня називають **кадрами**. Через фізичний рівень кадри передаються отримувачу. З усіх кадрів, які функціонують у каналі зв'язку, на каналному рівні відбувається відбір даних, адресованих даному комп'ютерові. Відібрані кадри на комп'ютері отримувача переміщуються з нижнього рівня на верхній, з них відокремлюється службова інформація, виконуються інші необхідні дії. Таким чином здійснюється відтворення переданої інформації та її отримання відповідною прикладною програмою.

6.3. Безпека роботи в комп'ютерних мережах

Використання мереж та серверів даних дозволяє використовувати санкціонований доступ до даних. На рівні файлової системи дані захищаються шляхом встановлення відповідних атрибутів або прав доступу. Мережеві версії програм надають різним користувачам різні права стосовно маніпуляції даними. Для входу в систему або програму застосовують імена та паролі.

Робота в мережі передбачає посилений захист від вірусів та атак програм хакерів на порти. Програмні порти – спеціальні канали доступу, відкриті для обміну інформацією в комп'ютерних мережах та Інтернеті. Захист портів передбачає використання спеціалізованих програм – **файрволів** (firewall) або **брандмауерів**. В операційній системі Windows існує декілька рівнів безпечної роботи в мережі Internet, які поділені на зони: Інтернет, місцева інтрамережа, надійні вузли, обмежені вузли. Окрім того, встановлення в ОС Windows рівня конфіденційності дозволяє обмежити приймання та застосування "cookie" на комп'ютері.

Cookie – технологія, яка підтримується протоколом http, що дозволяє створювати текстовий файл розміром 4 Кб на комп'ютері користувача з інформацією про нього і відомостями про характер перегляду сайту, персональні налаштування.

Власник комп'ютера може дозволити приймання всіх «cookie» або забезпечити повне їх блокування. Середній рівень конфіденційності забезпечує блокування «cookie», які не відповідають політиці безпеки, та «cookie», які використовують особисту інформацію без згоди власника комп'ютера.



Використання ресурсів Інтернет та електронної пошти передбачає отримання електронних повідомлень – листів. Серед листів з потрібною інформацією, є достатньо багато **спаму** – непрошених повідомлень в електронній пошті або телеконференціях. Спам, окрім надоїдливості та перевантаження великою кількістю повідомлень, може містити віруси та троянські програми. Для боротьби із спамом використовують фільтрацію повідомлень у поштових програмах або спеціалізовані програми-фільтри.

Робота в локальних мережах, особливо при їх підключенні до глобальних мереж, передбачає інтенсивне використання мережевої політики безпеки. Адміністратором мережі обмежується як доступ в локальну мережу ззовні, так і вихід за межі локальної мережі користувачів, які не мають на це права. Для забезпечення політики безпеки між локальною та глобальною мережами встановлюють **брандмауери**. Ними можуть бути програми або спеціалізовані комп'ютери. Застосування в локальних мережах комутаторів дозволяє виключити майже повністю можливість перехоплення та розшифрування переданої інформації. Програми перехоплення та розшифрування інформації називають **сніферами**.

Для роботи за комп'ютером в локальній мережі кожен користувач повинен мати свій обліковий запис, в якому вказано його ім'я, права використання ресурсів комп'ютера та мережі. Користувачів з однаковими правами об'єднують в групи. Приклади груп: системні адміністратори, звичайні користувачі, користувачі Internet, гості та інше. Операційна система Windows дає змогу об'єднувати декілька комп'ютерів локальної мережі в робочу групу. В робочій групі можна використовувати власну політику безпеки та бази даних бюджетів користувачів. Кожен власник комп'ютера в робочій групі самостійно вирішує, які ресурси виділити у спільне користування, а на які встановити обмеження прав доступу до своїх ресурсів. У робочу групу об'єднують комп'ютери тих користувачів, які мають один напрямок діяльності або працюють в одному приміщенні.

6.4. Мережа Internet

6.4.1. Протокольний стек TCP/IP

В кожній мережі повинно забезпечуватися надійне передавання даних за потрібною адресою, що забезпечується протоколами транспортного та мережевого рівнів на базі інших протоколів. Першим протоколом транспортного рівня став протокол АННР в

мережі ARPANET. Далі протоколи TCP та IP, розроблені для ОС Unix, стали стандартом для використання в багатьох локальних та глобальних мережах.

IP (англ. Internet Protocol – міжмережевий протокол) – протокол мережевого рівня для передавання та маршрутизації повідомлень.

TCP (англ. Transmission Control Protocol – протокол керування передачею) – протокол транспортного рівня з попереднім налагодженням сполучення, гарантує надійне передавання пакетів та правильну послідовність їх проходження.

Застосування протоколів TCP та IP назвали протокольним стеком і записують TCP/IP. Архітектура цього протокового стеку відповідає моделі ISO/OSI та включає в себе наступні об'єкти: процеси, хости, мережі. Модель протокового стеку TCP/IP містить чотири рівні: прикладний, транспортний, міжмережевий, мережевого інтерфейсу. Цю модель підтримують всі сучасні операційні системи, зокрема Windows XP, Vista.

*Об'єднання декількох мереж на базі протокового стеку TCP/IP називають **internet**. Корпоративні мережі, які побудовані на принципах **internet**, називають **intranet**. Мережу **intranet**, яка тісно зв'язана з іншими мережами, називають **extranet**.*

***Інтернет (Internet)** – сукупність інформаційних ресурсів, які містяться в глобальній комп'ютерній мережі **internet**.*

В протоковий стек TCP/IP входять протоколи каналного рівня; протоколи прикладного рівня та рівня відображення: Telnet, FTP, SMTP, SNMP, HTTP; протоколи ICMP, ARP, RARP, TCP, UDP (див. табл. 6.1). Всі протоколи стеку TCP/IP використовують сервіси протоколу IP. Але протокол IP не підтримує послідовного передавання пакетів та не гарантує надійності їх передавання.

Таблиця 6.1.

Додаткові протоколи стеку TCP/IP

Назва	Призначення
ICMP	Діагностичний протокол для інформації, що передається. Застосовується також для адміністрування
ARP	Протокол трансформування IP-адреси в MAC-адресу

RARP	Протокол визначення за MAC-адресами IP-адрес
UDP	Данаграмний протокол транспортного рівня
Telnet	Протокол емуляції термінала
FTP	Протокол передавання файлів
SMTP	Протокол електронної пошти
SNMP	Протокол керування мережею
HTTP	Протокол передавання гіпертекстової інформації

6.4.2. Адресація в TCP/IP

Кожен рівень моделі комп'ютерної мережі містить свій механізм адресації, яка визначається стандартами протоколів. Адреса IP-протоколу мережевого рівня є IP-адресою вузла. Для протоколу стандарту IPv4 адреса є унікальним 32-бітним числом, яке для зручності записується 4 десятковими числами від 0 до 255, що відділяються крапками. Наведемо приклади адрес:

192.112.36.5; **128.174.5.6;** **10.6.79.32.**

На транспортному рівні адресою виступає номер порту, що відповідає службі, якій адресовано IP-пакет. Номери портів знаходяться в межах від 0 до 65535. Для системних засобів використовують номери до 1024. Наприклад: 23 – порт служби FTP для передавання інформації за протокольним стеком TCP; 80 – порт, який застосовується для передавання інформації за протоколом HTTP.

Як правило, звертання до порту відбувається через **сокети** (socket), які є основними засобом взаємодії процесів в мережі. Прикладні програми підключаються до сокетів і видають запит на прив'язку до потрібної адреси. Потім дані передаються від одного сокета до іншого за вказаною адресою.

Сокет (англ. socket - гніздо) – інтерфейс між прикладним та транспортними рівнями що є засобом організації взаємодії між прикладними процесами в мережі.

Таким чином, для того щоб розрізнити потоки даних, які надходять конкретному користувачеві або програмі на даній машині, в протоколі IP використовують сокети, тобто адресу машини та адресу порту. Наприклад: 10.0.0.1:80

Оскільки internet - це об'єднання багатьох мереж, то в кожній IP-адресі міститься інформація для маршрутизаторів про те, до якої



мережі належить комп'ютер. На номер мережі вказують ті числа адреси, які записані спочатку. Права частина адреси вказує на номер комп'ютера в мережі. Таким чином, адреси комп'ютерів однієї мережі містять однаковий префікс. Номери будь-яких двох вузлів в одній мережі збігатися не повинні. Мережі поділені на класи, кожному класу відповідає свій діапазон адрес (таблиця 6.2). Формат повної адреси вузла має вигляд: a.b.c.d.

Таблиця 6.2.

Діапазони адрес мереж різних класів

Клас	Значення 1-го байта (a)		Адреса мережі	Адреса вузла	Кількість мереж	Кількість вузлів
	min	max				
A	1	126	a	b.c.d	126	16777214
B	128	191	a.b	c.d	16384	65534
C	192	223	a.b.c	d	2097151	254

Адреса мережі зі значенням першого байта 127 зарезервована для тестової перевірки наявності зв'язку. Адреси мереж, починаючи з номера 224, можуть використовуватися тільки спеціальним чином і для адрес комп'ютерів не застосовуються. Для мереж, які не мають прямого виходу в Internet або мають вихід через шлюз, використовують наступні діапазони адрес комп'ютерів: 10.0.0.1-10.255.255.254, 172.16.0.1-172.31.255.254, 192.168.0.1-192.168.255.254.

Адреси в мережі Internet призначаються єдиним відомством – Мережевим інформаційним центром (Network Information Center – NIC). Цим забезпечується унікальність адрес. Організації, яка подала запит на отримання адреси, видається тільки мережева частина адреси, яка залежить від кількості комп'ютерів у мережі. Подальший розподіл адрес всередині мережі проводить системний адміністратор мережі.

На практиці при адресації вузлів в мережах використовують 32-х бітну маску вузла. В ній послідовність одиничних бітів виділяє номер мережі, а нульові біти відповідають номерам вузлів у мережі. Наприклад, якщо для адреси 10.2.42.15 застосувати маску 255.255.255.0, то отримаємо номер мережі 10.2.42, а наведена маска позначає всі комп'ютери цієї мережі класу C.

З виникненням дефіциту адресного простору 4-х байтних IP адреспочався поступовий перехід до нової версії протоколу IPv6, яка



містить 128-бітну адресу.

Один комп'ютер може бути підключеним до декількох мереж. В цьому випадку він повинен мати стільки різних мережевих пристроїв під'єднання з MAC-адресами, до скількох мереж він підключений. Йому також призначають відповідну кількість логічних адрес. Найчастіше декілька адрес мають шлюзи.

6.4.3. Доменні імена

Кожен вузол мережі, як уже відмічалось, має свою унікальну IP-адресу, яка може бути статичною і видаватися провайдером або бути динамічною (визначається сервером кожен раз при новому підключенні). Однак користувачам застосовувати IP-адреси незручно. Набагато краще мати справу із символічними іменами які легко запам'ятовуються. Для однозначного перетворення IP-адреси у символічне ім'я (і навпаки) існує служба доменних імен (англ. Domain Name System, або скорочено DNS), яка започаткована у 1983 р. Служба DNS використовує структуру об'єктів, яку називають деревом каталогів (рис. 6.2).

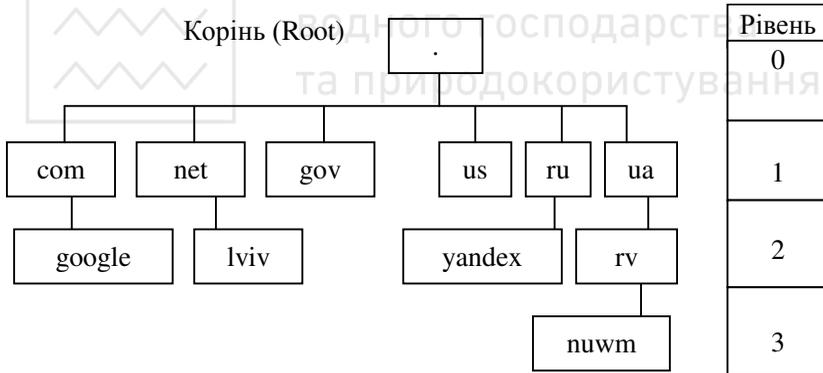


Рис. 6.2. Організація структури доменних імен

Доменами називають елементи рівнів (вузли дерева), які мають ім'я та можуть містити вкладені домени нижчого рівня (піддомени, субдомени).

*Верхній запис в ієрархії імен DNS позначається крапкою і називається **кореневим доменом**. Нижче від нього знаходяться домени першого рівня, що визначаються за: **географічним розміщенням (країна)** чи **типом організації** (див табл. 6.3).*



Перелік доменів першого рівня

Домен	Пояснення
arpa	мережа ARPAnet
arts	ресурси гуманітарної природи
com	комерційні організації
edu	навчальні заклади (університети, школи тощо)
firm	ділові ресурси мережі
gov	урядові установи (крім військових)
info	надання інформаційних послуг
mil	військові організації (армія, флот і т.д.)
net	мережні служби
nom	індивідуальні ресурси
org	некомерційні організації
rec	ігри і розваги
store	торгівля
web	організації, що регулюють діяльність www
xx	двобуквенні коди країн (ua –Україна, ru - Росія), які аналогічні позначенням на автомобільному транспорті

Система DNS складається з розподіленої бази даних, бібліотечних процедур, протоколу обміну інформацією з базою даних, сервісних засобів. Комп'ютер, де зберігається база даних DNS, називається **DNS-сервером**. Всі DNS-сервери з'єднані між собою, структура зв'язків відповідає ієрархії доменів. Хости у базі даних DNS утворюють чітку ієрархію, адміністрування управління якими покладається на адміністраторів відповідного рівня.

Управління коренем та першим рівнем дерева доменів здійснюється відомством NIC. Всі решта доменів адмініструються окремими організаціями відповідного рівня компетентності. Власник



домену відповідає за адміністрування всіх імен піддоменів свого домену. Максимальна довжина повного доменного імені не повинна перевищувати 255 символів, а довжина імені комп'ютера – 63 символи. Усі імена нечутливі до регістра.

На нижньому рівні ієрархії доменів знаходяться імена хостів. Доменне ім'я хосту складається з послідовного переліку знизу до верху всіх імен відповідної гілки дерева, між якими ставиться крапка. *Наприклад*, ім'я **nuwm.rv.ua** позначає комп'ютер, який розташований на території України, перебуває в піддоміні rv – Рівне та має ім'я nuwm.

6.4.4. Мережеві сервіси

Кожен користувач в мережі використовує сервісні послуги відповідних протоколів. Починаючи з 1990 року в мережі Internet, практично всі протоколи були витіснені протокольним стеком TCP/IP. У 1989 році була опублікована ідея Тіма Бернерса-Лі щодо управління інформацією, яка дозволяла включати в документи посилання на інші документи. На базі цього у 1990 році розроблений перший Web-документ, а у 1991 році для клієнтів UNIX розроблений перший браузер. У 1993 році було дозволено використання технології WWW без обмежень, що сприяло появі перших браузерів для ОС Windows та Mac. Компанією Netscape, яка створена у 1994 р., був розроблений популярний браузер з назвою Netscape спочатку для Unix-систем, а потім і для Windows.

Всі сервіси мережі можна умовно поділити на дві категорії: **обмін інформацією** між абонентами і **використання баз даних** мережі. Розглянемо найбільш популярні види сервісів, які використовуються для зв'язку між абонентами:

1. **TelNet** - віддалений доступ. Він дає можливість абонентові працювати на будь-якій ЕОМ мережі Internet (запускати програми, змінювати режим роботи тощо), як на своїй власній. Технологію TelNet застосовують для дистанційного керування технічними об'єктами, наприклад, промисловими маніпуляторами, відеокамерами, телескопами та ін.
2. **FTP** (англ. File Transfer Protocol - протокол передачі файлів) дозволяє абонентові обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі. **NFS** (англ. Network File



System (операційна розподілена файлова система) надає можливість користуватися файловою системою віддаленого комп'ютера.

3. **E-mail** (електронна пошта) - обмін поштовими повідомленнями з будь-яким абонентом мережі Internet. Тут можна пересилати як текстові, так і двійкові файли. Для обміну повідомленнями використовують одну із програм The Bat!, MS Outlook, MS Mail та ін. Поштова адреса має вигляд: **petrenko@nuwm.rv.ua** При відправленні повідомлення зазвичай вказують адресу отримувача, адреси користувачів, які повинні отримати копії листа, тему повідомлення. До листа можна приєднувати певну кількість файлів-вкладень.
4. **ICQ** – пошук IP-адреси абонента, підключеного до мережі, і спілкування користувачів у режимі реального часу. Спілкування відбувається шляхом обміну текстовими повідомленнями.
5. **Net-meeting** (Internet-телефонія) забезпечує спілкування з іншою людиною через мережу в режимі реального часу. Існує ряд вимог для забезпечення послуги: наявність у ПК колонок, мікрофона. Крім того, потрібно забезпечити паралельний вихід з абонентом на один із розмовних серверів.
6. **News** (новини) – створює можливість одержання новин із електронних дощок оголошень і переміщення власної інформації на дошки оголошень. Електронні дошки оголошень мережі Internet формуються за тематикою. Користувач може вибрати будь-які групи новин.
7. **IRC** (чат) – пряме спілкування кількох людей у режимі реального часу (по черзі через псевдонім). Програма може бути застосована для ділового спілкування, проведення нарад, консультацій та ін.

Крім перелічених вище послуг, мережа Internet надає також наступні **специфічні послуги**:

Webster - мережева версія тлумачного словника англійської мови.

Факс-сервіс - дає можливість користувачеві відправляти повідомлення факсимільним зв'язком, користуючись факс-сервером мережі.

Електронний перекладач – виконує переклад висланого тексту з однієї мови на іншу.



До систем автоматизованого пошуку інформації в мережі Internet належать:

Gopher - дозволяє знаходити інформацію за ключовими словами і фразами. Робота із системою Gopher нагадує перегляд змісту. Для вибору потрібної теми користувачеві пропонується ряд вкладених меню. Крім текстів Gopher-сервери містять графіку та аудіо, які можна відтворювати чи пересилати. У випадку виникнення труднощів, можна скористатися службою Veronica, яка здійснює пошук більш ніж у 500 системах Gopher, звільняючи користувача від рутинної роботи.

WAIS - ще більш потужний засіб одержання інформації, ніж Gopher, оскільки вона здійснює пошук ключових слів у всіх текстах документів. Запити посилаються у Wais спрощеною англійською мовою. Це робить Wais більш привабливою для користувачів-непрофесіоналів. У мережі Internet існує понад 300 Wais-бібліотек.

WWW (англ. *World Wide Web* – *всесвітнє навутиння*) – глобальна гіпертекстова система, яка використовує Internet як транспортний засіб. Потенційно вона є найбільш могутнім засобом пошуку. Гіпертекст з'єднує різні документи на основі заздалегідь визначених слів - **гіперпосилань**. Для сервісу www стандартом є протокол http, наприклад <http://www.1plus1.tv>

Окремі документи називаються **web-сторінками**, групи тематично об'єднаних сторінок – **web-сайтами**.

На одному web-сервері може бути розташовано декілька web-сайтів.

Для перегляду web-сторінок призначені спеціальні програми, які називають **браузерами**. Браузер виконує відображення документа на екрані керуючись **тегами** – командами, які автор вставив у його текст. Правила запису тегів визначаються мовою розмітки гіпертекстів – **HTML** (англ. *Hyper Text Mark Language*).

Серфінг – довільне переміщення між документами. **Навігація** – цілеспрямований пошук документів.

Приклади найбільш поширених браузерів: Opera, Mozilla Firefox, Netscape Navigator, Mosaic та Internet Explorer, який влаштований у операційну систему Windows. Робота в браузерах аналогічна роботі у програмі Провідник (особливо це стосується Internet Explorer, див. рис. 6.3).

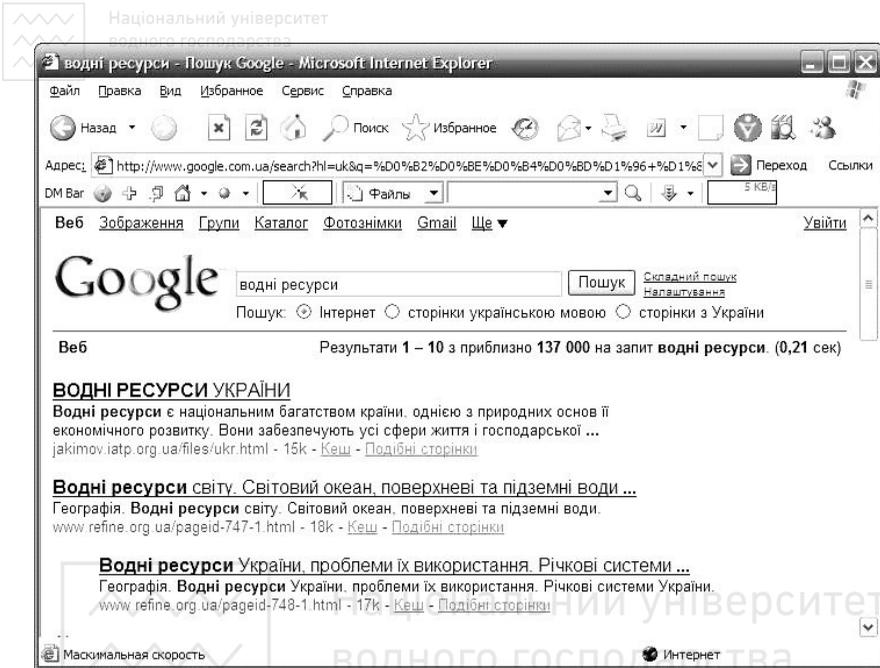


Рис. 6.3. Вікно програми Internet Explorer

Основними завданнями браузерів є передавання запитів на отримання ресурсів та отримання відповіді від сервера ресурсів чи отримання самих ресурсів, якщо вони є. Окрім того, браузери дозволяють: переглядати мультимедійну інформацію, копіювати інформацію на свій комп'ютер із сторінки тощо. Користувач може в браузері відразу перейти на сторінку з потрібною інформацією шляхом введення адреси Web-сторінки в адресному рядку. Якщо невідомо, де розміщена потрібна інформація, то можна скористатися послугами пошукових серверів, таких як google.com.ua, yandex.ru, yahoo.com та ін. Пошук інформації починається із введення ключових слів.

Для підключення до мережі Internet необхідно вибрати провайдера та укласти з ним угоду про надання послуг зв'язку.

Провайдер – фірма або організація, яка надає клієнтам послуги з використання ресурсів Internet.

Напрямок розвитку Internet в основному визначає комітет Спільнота Internet (англ. Internet Society, ISOC). ISOC – це організація на

громадських засадах, метою якої є сприяння глобальному інформаційному обміну через Internet. Вона призначає раду старійшин, що відповідає за технічне керівництво і розвиток Internet. Рада старійшин затверджує стандарти і розподіляє ресурси (*наприклад*, адреси).

Діапазон прикладних програм, які використовуються для роботи в мережі, досить широкий: від патентованих, що поставляються потужними фірмами-розробниками, до написаних користувачами самостійно. Найбільш стандартизовані прикладні програми використовують для організації віддаленого доступу, обміну файлами і електронною поштою.

В даний час у мережі Internet використовуються практично усі відомі лінії зв'язку від низькошвидкісних телефонних ліній до високошвидкісних цифрових супутникових каналів. Операційні системи, які використовуються в мережі Internet, також відрізняються розмаїтістю.

Практично всі сервіси мережі побудовані на принципі **клієнт-сервер**. Взаємодія клієнт-сервер будується зазвичай у такий спосіб: після надходження запитів від клієнтів сервер запускає програми надання мережевих сервісів. В міру виконання запущених програм сервер відповідає на запити клієнтів.

Усе програмне забезпечення мережі також можна поділити на клієнтське і серверне. При цьому програмне забезпечення сервера займається наданням мережевих сервісів, а клієнтське програмне забезпечення здійснює передачу запитів серверові та одержання відповідей від нього.

Контрольні запитання і завдання

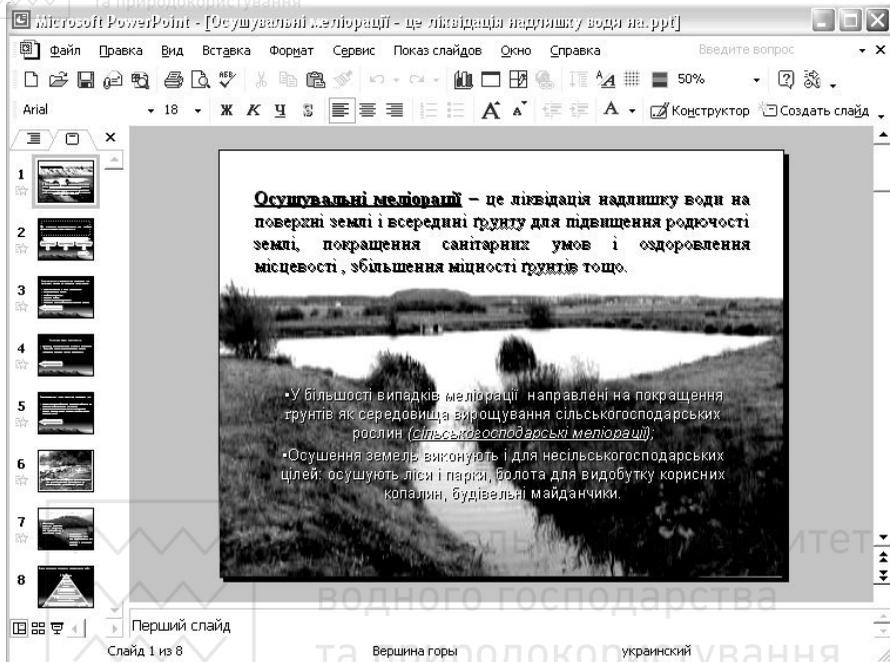
1. Який склад мережі?
2. Яка структура мережі?
3. Що називають хостом, сервером, вузлом?
4. Які базові принципи організації мережі?
5. Що таке протокол?
6. Як розшифрувати TCP?
7. Як розшифрувати IP?
8. Що визначає структура IP-адреси?
9. Що називають MAC-адресою?
10. Які функції провайдера?

- 
11. На якій кількості серверів може розташовувати свою інформацію провайдер?
12. Які найбільш популярні служби існують в Internet?
13. Яке мета UseNet?
14. Яке призначення FTP?
15. Розшифруйте WWW. Що називається гіпертекстом?
16. Що називається Web-сторінкою? Що називається Web-сайтом?
17. Що називають браузером?
18. Які є класи мережевих вирішень?
19. Що називають відкритою системою?
20. Які апаратні засоби потрібні для підключення до мережі?
21. Які основні принципи безпечної роботи в мережі?
22. Які існують класи мереж?
23. Яка структура доменних імен?
24. Які існують мережеві сервіси?
25. Яка структура протокольного стеку TCP/IP?
26. Які рівні включає в себе модель взаємодії відкритих систем?
27. Як проводити пошук інформації в мережі Інтернет?
28. Що називається концентратором, комутатором?
- 
- Національний університет
водного господарства
та природокористування



Додаток 1.1. Частина базової таблиці кодування ASCII

32	пропуск	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	“	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	‘	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	





Додаток 5.1. Типи даних

Тип	Розмір, байт	Діапазон значень і точність	Суфікс	Префікс	Примітка
Boolean	2	True, False		bln	логічне
Integer	2	$-2^{15} .. 2^{15}-1$	%	int	ціле
Byte	1	$0 .. 2^8-1$		byt	ціле
Long	4	$-2^{31} .. 2^{31}-1$	&	lng	ціле
Currency	8	$-2^{63} * 10^{-4} .. (2^{63} - 1) * 10^{-4}$	@	cur	дійсне, 4 знаки після коми
Date	8	1 січня 100 р. ... 31 грудня 9999 р.		dtm	дата і час
Single	4	$-3.4E+38 .. -1.4E-45$; $1.4E-45 .. 3.4E+38$!	sng	дійсні числа з плаваючою крапкою
Double		$- 1.79769313486232D+308 .. -4.94D-324$; $4.94D-324 .. 1.79769313486232D+308$	#	dbl	дійсні числа з плаваючою крапкою
String	10 б. + 1 б. на 1 символ, 1 б. на 1 символ	Змінна довжина: від 0 до ≈ 2 млрд. символів, фіксована довжина: від 1 до $\approx 654\,000$ символів	\$	str	рядок символів
Variant	16 б. + 1	Діапазон зале-		vnt	будь-який

	б. на 1 символ	жить від фактично збережених даних			тип даних
Object	4	Адреса об'єкту		obj	забезпечує доступ до об'єктів

Додаток 5.2. Таблиця істинності логічних операторів

Операнди		Операції над операндами <i>bytX1</i> та <i>bytX2</i>				
<i>bytX1</i>	<i>bytX2</i>	And	Or	Xor	Imp	Eqv
True	True	True	True	False	True	True
True	False	False	True	True	False	False
False	True	False	True	True	True	False
False	False	False	False	False	True	True

Додаток 5.3. Функції роботи з рядками символів та датами

Рядкові функції	
Asc(<i>strX</i>)	Повертає код першого символу рядка <i>strX</i>
Chr(<i>x</i>)	Повертає символ, ASCII-код якого дорівнює числовому аргументу <i>x</i>
InStr([<i>intStart</i> ,] <i>strX1</i> , <i>strX2</i>)	Повертає номер позиції у рядку <i>strX1</i> , починаючи з якої рядок <i>strX2</i> перший раз входить у <i>strX1</i> . Пошук у рядку <i>strX1</i> проводиться починаючи із символу <i>intStart</i> (якщо вказаний цей аргумент), в іншому випадку – з 1-го
Lcase(<i>strX</i>)	Повертає рядок <i>strX</i> , записаний символами тільки нижнього регістра
Left(<i>strX</i> , <i>intI</i>)	Повертає <i>intI</i> перших символів рядка аргументу <i>strX</i>
Len(<i>strX</i>)	Повертає число символів у рядку <i>strX</i>
Ltrim(<i>strX</i>)	Повертає рядок <i>strX</i> із вилученими початковими пробілами
Mid(<i>strX</i> , <i>intStart</i> [,	Повертає підрядок рядкового аргументу <i>strX</i> , що починається символом із номером <i>intStart</i> <i>i</i> / або продовжується до кінця рядка, або (якщо вказано

intLen)	необов'язковий аргумент intLen) має довжину intLen символів.
Right(strX, intI)	Повертає intI завершальних символів рядка strX
Rtrim(strX)	Повертає рядок strX, в якому видалено всі завершальні пробіли
Str(x)	Перетворює числове значення x на рядок символів
Ucase(strX)	Повертає рядок strX, записаний символами тільки верхнього регістра
Val(strX)	Перетворює рядок strX на число
Функції роботи з датами	
Date	Повертає поточну системну дату, яка встановлена на комп'ютері
Time	Повертає поточний системний час
Now	Повертає поточну системну дату та час
DateAdd (interval, intX, dtmDat)	Повертає дату dtmDat, яка змінена на заданий числовий інтервал intX часу. Значенням аргументу interval є наступні рядки символів: "ууу" – рік; "q" – квартал; "m" – місяць; "у" – день року; "d" – день; "w" – день тижня; "ww" – тиждень; "h" – година; "n" – хвилина; "s" – секунда
DateDiff (interval, dtmDat1, dtmDat2)	Повертає різницю між датами dtmDat1 та dtmDat1 у вказаних часових інтервалах (аргумент interval)

Додаток 5.4. Константи кнопок та вікна функції MsgBox

Константа	Значення	Набір кнопок у діалоговому вікні
vbOkOnly	0	ОК
vbOkCancel	1	ОК, Отмена
vbAbortRetryIgnore	2	Стоп, Повтор, Пропустить

1	vbYesNoCancel	3	Да, Нет, Отмена
	vbYesNo	4	Да, Нет
	vbRetryCancel	5	Повтор, Отмена
n1	vbDefaultButton	0	Встановлює фокус на 1-й кнопці
n2	vbDefaultButton	256	Встановлює фокус на 2-й кнопці
n3	vbDefaultButton	512	Встановлює фокус на 3-й кнопці
n4	vbDefaultButton	768	Встановлює фокус на 4-й кнопці
Modal	vbApplicationModal	0	Призначає модальність додатку. Діалогове вікно буде модальним, тобто буде вимагати переходу в інші вікна
1	vbSystemModal	4096	Призначає модальність системи. Діалогове вікно буде модальним на рівні системи, тобто поки воно не закрито, перейти у будь-який інший додаток неможливо
	vbMsgBoxHelpButton	16384	Добавляє у діалогове вікно кнопку Справка
	VbMsgBoxSetForeground	65536	Оголошує діалогове вікно фоновим
t	vbMsgBoxRight	524288	Вирівнює текст у діалоговому вікні по правому краю
	vbMsgBoxRTLReading	1048576	Перевертає текст для читання справа наліво



Значок	Константа	Значення	Тип повідомлення
	vbExclamation	48	Попередження
	vbQuestion	32	Запит
	vbInformation	64	Інформація
	vbCritical	16	Помилка

Додаток 5.6. Оператори та функції роботи з файлами

Назва	Опис
ChDir (<шлях>)	Встановлює новий робочий каталог на поточному диску
ChDrive (<диск>)	Встановлює новий робочий диск
CurDir	Функція. Повертає шлях до робочого каталога
MkDir (<шлях>)	Створює новий каталог
Rmdir (<шлях>)	Знищує існуючий каталог. Каталог повинен бути порожнім
Kill (<файл_змінна>)	Знищує файл на диску. Файл повинен бути закритим
Name <назва_файлу1> As <назва_файлу2>	Переіменовує файл1 на файл2
FileCopy <назва_файлу1>, <назва_файлу2>	Копіює файл1 у файл2



1. Ананьев Александр. Самоучитель Visual Basic 6.0 / Александр Ананьев, Аркадий Федоров. – СПб : БХВ-Петербург, 2005. – 624 с. : ил. – ISBN 5-8206-0060-6.
2. Армстронг Джеймс. Секреты UNIX. / Джеймс Армстронг ; – М. : Вильямс, 2001. – 1072 с. – ISBN 5-8459-0068-9.
3. Информатика. Комп'ютерна техніка. Комп'ютерні технології : підручник / Баженов Віктор Андрійович, Венгерський Петро Сергійович, Горлач Віталій Михайлович та ін. – К. : Каравела, 2004. – 464 с. – ISBN 966-8019-05-9.
4. Информатика: Комп'ютерна техніка. Комп'ютерні технології : посібник / Браткевич В. В., Бутов М. В., Золотарьова І. О. та ін. ; за ред. О. І. Пушкаря. – К. : Академія, 2002. - 704 с. – ISBN 966-580-085-X.
5. Буров Євген. Комп'ютерні мережі / Євген Буров ; [за ред. проф. Пасічника]. – Львів : БаК, 2003. - 584 с. – ISBN 966-7065-41-3.
6. Войтюшенко Н. М. Информатика і комп'ютерна техніка. – К. : ЦНЛ, 2006. – 568 с.
7. Глинський Ярослав Миколайович. Практикум з інформатики : навч. посіб. / Я. М. Глинський. – Львів : СПД Глинський, 2007. - 296 с. – ISBN 978-966-2934-00-7.
8. Глушаков Сергей Владимирович. Программирование на Visual Basic 6.0 : учебный курс / С. В. Глушаков, А. С. Сурядный – Харьков : Фолио, 2004. - 497 с. – ISBN 966-03-2437-5.
9. Дейт Крис Дж. Введение в системы баз данных : пер. с англ. / Крис Дж. Дейт – М. : Вильямс, 2005. - 1328 с. – ISBN 5-8459-0788-8.
10. Долженков Виктор. Microsoft Excel 2000 / Виктор Долженков, Юлий Колесников. – СПб: БХВ, 1999. - 1088 с. – ISBN 5-8206-0027-4.
11. Зубик Людмила Володимирівна. Информатика та комп'ютерна техніка у водному господарстві : навч. посіб. / Людмила Володимирівна Зубик, Ярослав Ярославович Зубик, Іван Миколайович Карпович. – Рівне : НУВГП, 2008. – 306 с. – ISBN 978-966-327-103-3.
12. Информатика. Базовый курс : учебное пособие для вузов / С. Симонович, Г. Евсеев, В. Мураховский, С. Бобровский ; под ред.



С. Симоновича. - 2-е изд. - СПб. : Питер, 2005. - 640с. : ил. - ISBN 5-94723-752-0.

13. Інформатика для економістів : навч. посіб. для студ. ВНЗ / Беспалов В.М., Вакула А.Ю., Гострик А.М. та ін. – К. : ЦУЛ, 2003. – 788 с. – ISBN
14. Кузьменко В. Г. Базы данных в Visual Basic и VBA: Самоучитель. – М. : ООО “Бином-Пресс”, 2004. – 416 с.
15. Леонтьев Виталий Петрович. Новейшая энциклопедия персонального компьютера 2003 / В. П. Леонтьев. – М. : ОЛМА-ПРЕСС, 2003. – 957 с. : ил. – ISBN 5-224-04035-3.
16. Лук’янова Валентина В’ячеславівна. Комп’ютерний аналіз даних : посібник / В. В. Лук’янова. – К. : Академія, 2003. - 344 с. – ISBN 966-580-160-0.
17. Мамченко Світлана Дмитрівна. Економічна інформатика. Практикум : навч. посіб. / С. Д. Мамченко, В. А. Одинець. – К. : Знання, 2008. – 710 с. – ISBN 978-966-346-437-4.
18. Марков Александр Сергеевич. Базы данных. Введение в теорию и методологию : учебник / А. С. Марков, К. Ю. Лисовский – М. : Финансы и статистика, 2004. - 512 с. – ISBN 5-279-02298-5.
19. *Пройдаков Е. М. Англо-український тлумачний словник з обчислювальної техніки, Інтернету і програмування / Е. М. Пройдаков, Л. А. Теплицький. – Київ : СофтПрес, 2006. - 824 с. – ISBN 966-530-070-9.*
20. Інформатика та комп’ютерна техніка : навчально-методичний посібник для самостійного вивчення дисципліни / Д. Рзаєв, О. Шарапов, В. Ігнатенко, Л. Дибкова. К. : КНЕУ, 2006. – 486 с. – ISBN 966-574-458-5.
21. Таненбаум Эндрю. Архитектура компьютера / Эндрю Таненбаум. СПб : Питер, 2007. – 844 с. – ISBN 5-469-01274-3.
22. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта и др. : Пер. с англ. А. К. Белоцкого и др. – М. : Машиностроение, 1991. - 560 с. : ил. – ISBN 5-217-00617-X(СССР).
23. Фигурнов Виктор Эвальдович. IBM PC для пользователя / В.Э. Фигурнов. – М. : ИНФРА, 2006. – 640 с. – ISBN 5-86225-292-4.
24. Чаповська Роксоляна Богданівна. Робота з базами даних Microsoft Access 2000 / Р. Б. Чаповська, А. Є. Жмуркевич. – К. : ЦНЛ, 2004. - 324 с. – ISBN 966-8365-10-0.



Алфавітний покажчик термінів

Термін	Стор.	QBE-запит	163
ARPANET	275	RAM	30
ASCII	9	Software	31
BIOS	30	SOHO	280
CD-Rewritable	29	SQL	163
CD-ROM	28	TCP/IP	285
CMOS	31	TelNet	291
Cookie	283	UNICODE	10
DNS	288	Unix	37
dpi	25	WAIS	292
DVD	29	WAN	281
e-mail	291	Web	292
Extranet	285	Webster	292
Firewall	283	Windows	44
FTP	291	WWW	292
Gopher	292	WYSIWYG	84
Hardware	31	Zip-накопичувач	28
HiFD	28	Адреса клітини	126
host	276	Адресат	255
HTML	293	Алгоритм	211
http	286	Алфавітно-цифрові клавіші	20
IBM PC	17	Аналітична машина	12
ICQ	291	Антивірус	76
Intel	17	Арифметико- логічний пристрій	18
Internet	285	Архів	70
Intranet	285	Атрибут файлу	41
IPv4	286	База даних	160
IPv6	288	Базовий клас	272
IRC	292	Байт	7
Jaz	28	Біт	6
LAN	281	Бітрейт	80
Linux	37	Блок живлення	31
MAC-адреса	278	Блок клітин	127
mdb	162		
Net-meeting	291		
News	292		

Блок-схема	212	Електронні таблиці	122
Брандмауер	284	Елемент списку	104
Браузер	293	Запис	161
Буфер обміну	53	Запит	163
Вираз (Access)	203	Звіт	164
Вираз (VB6)	224	Змінна	219
Віддалений доступ	291	Значок	47
Відкрита система	282	Ідентифікатор	218
Відношення	161	Інтегральна схема	14
Вікно	49	Інтерпретатор	32
Вінчестер	31	Інтерфейс	11
Вірус	73	Інформатика	10
Вказівник файлу	238	Інформація	6
Властивості інформації	7	Ітераційний алгоритм	214
Волоконно-оптичний кабель	278	Кампусна мережа	281
Вузол	276	Канал передавання даних	276
Гігабайт	7	Каталог	40
Гіперпосилання	154	Категорії функцій	130
Грабер	80	Кеш-пам'ять	30
Двійкова система числення	6	Кільцева топологія мережі	277
Дефрагментація	60	Клавіатура	20
Джерело	255	Клавіші керування	20
Джойстик	25	Клас	272
Дигітайзер	25	Кластер	40
Дисководи	31	Клієнт-сервер	295
Діаграма	139	Клітина	126
Діалогові вікна	235	Ключ	169
Довжина вірусу	75	Ключове слово	218
Додаток	45	Коаксіальний кабель	278
Документ	47	Кодер	80
Домен	289	Кодування даних	8
Драйвер	33	Колекція	273
Евристичний аналізатор	77	Колонтитул	114
Екземпляр класу	272	Коментар	219
		Комп'ютер	13

Компілятор	32	Об'єкт (VB6)	271
Комп'ютерна мережа	276	Область видимості	221
Комутатор	278	змінної	
Константа	218	Однорангова мережа	281
Конструктор форм	260	Операнд	224
Контролери	30	Оператори VB6	228
пристроїв		Операційна система	15
Конфігурація	19	Організаційна	152
Концентратор	277	діаграма	
Координатні лінійки	86	Очистка диска	61
Корзина	59	Пароль	45, 72
Курсор	21		283
Лексема	217	Перезавантаження	23
Лінійка задач	47	комп'ютера	
Магістраль	277	Периферійні	19
Макрос	164	пристрої	
Маршрутизатор	276	Персональний	17
Масив	222	комп'ютер	
Маска	43	Піксель	19
Межа стискання	71	Плагін	69
Метод	272	Плата системна	29
Метод стискання	70	(материнська)	
Миша	24	Плейліст	80
Міжмережевий	285	Плотер	28
протокол		Подвійне слово	7
Мікропроцесор	17	Подія	272
Міра надлишковості	70	Поле	161
Модель даних	160	Поля сторінки	102
Модем	29	Порт	31
Модуль	164	Презентація	149
Монітор	19	Принтер	26
Монітор вікна	48	Принцип відкритої	17
Мультимедіа	79	архітектури	
Мультипрограмуван	15	Пристрій управління	18
ня		Пробіл	21
Нормалізація	172	Провайдер	293
Нотатки	150	Проверка диска	60
Об'єкт	47	Проводник	56

Проект	258	Спам	284
Протокол	282	Стан	271
Процедура у VB6	243	Стандартні функції	226
Процес	36	VB6	
Процесор	18	Сторінка	164
Рамка	153	Стример	28
Растр	10	Схема даних	177
Регістр	20	Таблиця символів	61
Редактор формул	118	Табуляція	108
Microsoft		Тег	293
Equation		Текстовий редактор	84
Реєстр	45	Телекомунікації	29
Режим вставки	22	Тип даних	168
(заміни)		Типи зв'язків	161
Ресурси комп'ютера	36	Топологія мережі	276
Робоча книга	123	Транслятор	32
Робоча станція	276	Трекбол	25
Робочий стіл	46	Утиліта	33
Роздільна здатність	19	Файл	7
Роздільник	217	Файли у VB6	237
Розшифрування	284	Файл-менеджер	68
Ряд даних	139	Файлова система	41
Сведения о системе	61	Файловий номер	238
Світлове перо	25	Факс-сервіс	292
Секвенсори	79	Фейрвол	283
Селектор	127	Фільтрація даних	192
Семпли	79	Форма	164
Сервер	276	Форматування диска	39
Синтезований звук	79	Форматування	98
Система баз даних	160	тексту	
Системний диск	39	Формула	128
Сканер	25	Функціональні	20
СКБД	162	клавіші	
Скролінг	22	Хост	276
Скручена пара	278	Цикл	213
Слово	7	Цифровий звук	79
Сніфер	284	Частота регенерації	19
Сокет	287	зображення	



Чіп
Шаблон
Шина

14
43
30

Шлюз
Шлях
Ярлик

276
43
47



Національний університет
водного господарства
та природокористування



Національний університет
водного господарства
та природокористування

Навчальне видання

*Зубик Людмила Володимирівна,
Зубик Ярослав Ярославович,
Карпович Іван Миколайович*

ІНФОРМАТИКА ТА КОМП'ЮТЕРНА ТЕХНІКА У ВОДНОМУ ГОСПОДАРСТВІ

Навчальний посібник



Друкується в авторській редакції

Дизайн обкладинки А.О. Харішина

Національний університет
водного господарства
та природокористування

Підписано до друку 30.05.2008 р. Формат 60x84 1/16.
Папір друкарський № 1. Гарнітура Times. Друк різнографічний.
Ум.-друк.арк.17,8. Обл.-вид. арк.18,6.
Тираж 150 прим. Зам. № 974.

*Редакційно-видавничий центр
Національного університету
водного господарства та природокористування
33028, Рівне, вул. Соборна, 11*

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготівників і розповсюджувачів видавничої
продукції РВ № 31 від 26.04.2005 р.*