

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОВСЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Карімов І. К.

КОНСПЕКТ ЛЕКЦІЙ

з дисципліни

**“ Комп’ютерні методи та засоби розв’язання
інженерних задач”**

для здобувачів вищої освіти першого

(бакалаврського) рівня зі спеціальностей:

101 “ Екологія ”, 161 “ Хімічні технології та інженерія ”,

162 “ Біотехнології та біоінженерія ”

(частина 1)

Затверджено:

Редакційно-видавничою секцією
науково-методичної ради ДДТУ

« 19 » 12 2019 рік

протокол № 10

Кам’янське
2019

Розповсюдження і тиражування без офіційного дозволу Дніпровського державного технічного університету заборонено.

Конспект лекцій з дисципліни “Комп’ютерні методи та засоби розв’язання інженерних задач” для здобувачів вищої освіти першого (бакалаврського) рівня зі спеціальностей 101 “Екологія”, 161 “Хімічні технології та інженерія”, 162 “Біотехнології та біоінженерія” (частина 1) / Укладач І.К. Карімов – Кам’янське: ДДТУ, 2019.- 73 с.

Укладач: Карімов І. К., канд. фіз.-мат. наук, доцент

Відповідальний за випуск

Карімов І. К., завідувач кафедри ВМ, к. ф.-м. н., доцент

Рецензент Дронов С.Г., к. ф.-м. н., доцент

(Дніпровський державний технічний університет)

Затверджено на засіданні кафедри вищої математики (протокол № 6 від 09.12.2019 р.).

Наведений виклад теоретичного матеріалу відповідно до програми дисципліни “Комп’ютерні методи та засоби розв’язання інженерних задач” для студентів спеціальностей 101, 161, 162 в обсязі першого модуля (семестру), включаючи окремі питання, що винесені на самостійну проробку студентами.

ЗМІСТ

1. ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ	5
1.1. Інформація, інформатика і інформаційні технології	5
1.2. Класична схема ЕОМ	6
1.3. Поняття апаратного і програмного забезпечення ЕОМ	7
1.4. Функціональний склад сучасного ПК	7
1.5. Програмне забезпечення ПК, його класифікація	9
1.6. Загальна характеристика операційних систем	11
1.7. Інтерфейс та основні принципи роботи з Windows	12
1.8. Техніка виконання основних операцій з об'єктами Windows	13
2. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ОСНОВА ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ	17
2.1. Інформаційні технології обробки даних	17
2.2. Класифікація інформаційних технологій	18
2.3. Мережні інформаційні технології	19
3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ НА ОСНОВІ MICROSOFT OFFICE	20
3.1. Пакет прикладних програм Microsoft Office, основні складові та сфери застосування	20
3.2. Інтерфейс та основні принципи роботи з MS Word	22
3.3. Редагування тексту	23
3.4. Форматування тексту	24
3.5. Інші можливості MS Word	28
3.6. Загальна характеристика MS Excel 2003	28
3.7. Створення та редагування електронних таблиць	30
3.8. Форматування електронних таблиць	31
3.9. Організація обчислень	33
3.10. Робота з функціями в MS Excel	35
4. ЧИСЕЛЬНІ МЕТОДИ ТА ЕЛЕМЕНТИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ	39
4.1. Поняття про чисельні методи	39
4.2. Сучасні інформаційні технології розв'язування задач обчислювальної математики	42
4.3. Загальні принципи використання системи MathCad	43
4.4. Робота з формульним редактором	45
4.5. Організація обчислень. Використання функцій в математичних виразах	47

4.6.	Матриці і ранжовані змінні	48
4.7.	Суть символьних обчислень	50
5.	ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ	52
5.1.	Поняття алгоритму, його властивості та способи зображення	52
5.2.	Блок-схеми алгоритмів	53
5.3.	Методи розробки алгоритмів	54
5.4.	Алгоритмізація типових обчислювальних процедур	56
5.5.	Типові елементи алгоритмічних мов програмування	59
5.6.	Найпростіші конструкції мови Паскаль	60
5.7.	Структура програми	62
5.8.	Оператор присвоєння	63
5.9.	Оператори введення-виведення	64
5.10.	Програмування алгоритмів лінійної структури	65
5.11.	Умовний оператор	65
5.12.	Приклад програми розгалуженої структури	67
5.13.	Оператори циклу	68
	СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	72

Тема 1. ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ

1.1. Інформація, інформатика і інформаційні технології

Інформація – одна з найважливіших сутностей нашого світу (поряд з речовиною та енергією). У найбільш широкому сенсі інформацією називають будь-які відомості про будь-який об'єкт, явище або процес. У прикладному розумінні *інформація – це сукупність певних даних (тексти документів, таблиці, малюнки, графіки і т. інше), які розміщені на матеріальних носіях і є об'єктами зберігання, передачі та перетворень.*

Сукупність операцій отримання, зберігання, передачі та перетворень (обробки) інформації складають *інформаційний процес*. Проведення інформаційного процесу має сенс тільки тоді, коли інформація має такі властивості, як *достовірність, зрозумілість, актуальність, корисність, повнота, однозначність.*

Діяльність людини нерозривно пов'язана з обробкою інформації. Саме потреба виразити і передати інформацію, зберегти її для подальшого використання призвела у свій час до появи мовлення і виникнення писемності, стимулювала створення друкарства, поштового зв'язку, телеграфу і телефону, радіо і телебачення. У свою чергу, кожне з зазначених відкриттів стало можливим тільки завдяки накопиченню і переробці інформації, яка вже мала.

Однак справжньою революцією в області накопичення, передачі та обробки інформації стала поява електронних обчислювальних машин (ЕОМ) – універсальних автоматичних пристроїв для обробки інформації. З появою ЕОМ, а отже народженням і поширенням нової автоматичної технології опрацювання інформації, фактично пов'язане формування нової наукової дисципліни – інформатики.

Інформатика – це наука про методи представлення, накопичення, передачі і обробки інформації за допомогою ЕОМ. Об'єктом інформатики є науково-технічна, соціально-економічна та інша інформація; предметом – науково-інформаційна діяльність; основним інструментом – ЕОМ.

В галузі металургії ЕОМ та їх різновид – персональні комп'ютери (ПК) – застосовуються для інформаційної підтримки та моделювання технологічних процесів, автоматизованого управління металургійними процесами, організації документообігу та комунікації відповідних служб тощо.

Для реалізації процесів обробки інформації в матеріальному світі в загальному випадку необхідні *носії інформації, передавач, канал зв'язку, приймач і одержувач інформації*. Сукупність цих елементів утворює так звану інформаційну систему. Склад і функції інформаційної системи визначаються, насамперед, специфічними особливостями і цілями функціонування. Як правило, інформація в інформаційній системі використовується для формування керуючих впливів. Тому, у найбільш загальному сенсі під *інформаційною системою* розуміється комплекс засобів і методів, що забезпечують процеси збору, збереження і передачі інформації, а також її

обробки відповідно до цілей функціонування.

Інформаційні системи функціонують на базі певної *інформаційної технології*. В своєму розвитку інформаційні технології пройшли ряд стадій. Так, інформаційні технології тридцятих-сорокових років будувалися на основі телефону, пошти, усних повідомлень, листів, звітів, різноманітних форм і бланків і т.п. Для сучасних інформаційних технологій характерне активне використання комп'ютерної техніки і засобів телекомунікації (комп'ютерних мереж), тобто сучасні інформаційні технології – це насамперед комп'ютерні технології. Тому надалі під *інформаційною технологією* ми будемо мати на увазі, насамперед, *сукупність методів виконання різноманітних операцій по обробці інформації за допомогою конкретних технічних і програмних засобів*.

Вивчення типових для галузі інформаційних технологій (відповідних методів та засобів), складає предмет дисципліни “Комп'ютерні методи та засоби розв'язання інженерних задач”.

1.2. Класична схема ЕОМ

Електронно-обчислювальні машини являють собою пристрої для автоматичної обробки інформації. В основі функціонування таких машин лежать ідеї цілого ряду вчених, що одержали логічне завершення в роботах відомого математика Джона фон Неймана. Саме він (разом з А. Берксом і Г. Голдстайном) опублікував у 1946 р. статтю "Попередній розгляд логічної конструкції електронного обчислювального пристрою", у якій дав опис структури ЕОМ, який не втратив своєї актуальності і нині.

Ключовими принципами функціонування ЕОМ є *принцип програмного керування* і *принцип збереженої програми*. Відповідно до першого принципу всі дії ЕОМ повинна виконувати по визначеній програмі, яка представляє собою послідовність інструкцій (команд), що реалізують заданий план дій.

Відповідно до другого принципу виконувана ЕОМ програма повинна зберігатися в спеціальному запам'ятовуючому пристрої (пам'яті). Для реалізації зазначених принципів потрібен набір пристроїв, показаний на рисунку 1.1.

Як же працює така машина? Насамперед, за допомогою пристрою введення в пам'ять ЕОМ вводиться програма. Пам'ять являє собою сукупність пронумерованих комірок, у кожній з яких знаходиться або команда виконаної програми, або оброблювані дані. Пристрій керування зчитує інформацію з комірки, у якій знаходиться перша команда програми, і організує її виконання. Після цього зчитується команда з наступної комірки і т.д. Кожна команда програми задає виконання певної операції з даними. До таких операцій можна віднести: арифметичні дії (додавання, віднімання, множення, ділення); логічні операції (наприклад, порівняння); операції введення-виведення даних; передачі керування і т.д. При виконанні арифметичних і логічних операцій використовується арифметико-логічний пристрій. Для введення в пам'ять ЕОМ необхідних даних застосовується пристрій введення, а для виведення

проміжних і кінцевих результатів роботи програми - пристрій виведення. Обробка команди передачі керування приводить до того, що пристрій керування чергову команду зчитує не з наступної комірки пам'яті, а з тієї, на яку вказує виконувана команда. Завершується виконання програми виведенням результатів на пристрій виведення, після чого ЕОМ переходить у режим очікування введення наступної програми.

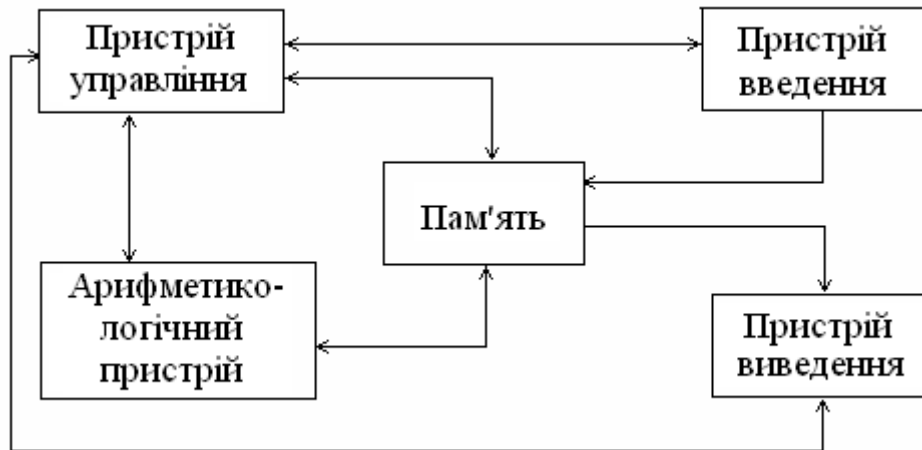


Рис.1.1. Класична схема ЕОМ

1.3. Поняття апаратного і програмного забезпечення ЕОМ

Розглядаючи ЕОМ як основний інструмент інформаційних технологій, слід виділяти дві рівноправні частини комп'ютера: *апаратну частину* (hardware) і *програмне забезпечення* (software).

Апаратну частину складають всі технічні засоби, які входять до складу ПК або під'єднуються до нього при необхідності. Під програмним забезпеченням мається на увазі сукупність програм системи обробки даних і програмних документів, необхідних для експлуатації цих програм.

1.4. Функціональний склад сучасного ПК

Серед сучасних персональних комп'ютерів виділяються два основних сімейства, що відрізняються конструктивними рішеннями і, відповідно, програмним забезпеченням: IBM-сумісні комп'ютери і комп'ютери Macintosh фірми Apple Computer. При цьому більше поширення в усім світі одержали IBM-сумісні комп'ютери. Одна з основних причин – *відкрита архітектура* таких комп'ютерів, яка зумовлена оприлюдненням фірмою IBM інформації про основні вузли розробленого ними ПК (*IBM PC*) та принципи їх взаємодії. Як наслідок, IBM-сумісний комп'ютер може складатися з вузлів, які виготовлені зовсім різними фірмами, що підтримують стандарт IBM PC. У нашій країні перевага таких комп'ютерів настільки велика, що зазвичай під терміном "комп'ютер" мається на увазі саме *IBM-сумісний комп'ютер*.

Структура сучасного IBM-сумісного комп'ютера представлена на рис. 1.2. В її основі лежать все ті ж принципи Джона фон Неймана. Однак, з урахуванням досягнень в області електроніки основні пристрої перетерпіли певні зміни. Арифметико - логічний пристрій і пристрій керування об'єднані в одному *мікропроцесорі*. Пам'ять розділена на *внутрішню пам'ять* і *зовнішню пам'ять*. Доступ до останньої організується за допомогою спеціальних пристроїв - *дисководів*. Внутрішня пам'ять представлена *оперативною пам'яттю* і *постійною пам'яттю*. Оперативна пам'ять застосовується для збереження виконуваних у даний момент програм і використовуваних ними даних. Постійна пам'ять служить для збереження програм, призначених для організації початкового завантаження комп'ютера. Постійна пам'ять доступна тільки для читання, змінювати її вміст користувач не може.

Пристрої введення на рис. 1.2 представлені *клавіатурою*, призначеною для введення символічної інформації, а пристрої виведення – *монітором* (*дисплеєм*) для виведення на екран і *принтером* для виведення на папір. Крім того є *дисководи*, які можна розглядати і як пристрої введення, і як пристрої виведення інформації. Крім зазначених пристроїв введення-виведення до комп'ютера можуть підключатися й інші зовнішні пристрої. Усі пристрої комп'ютера зв'язані за допомогою *системної шини*, до якої вони підключаються через спеціальні плати з мікросхемами, називані *адаптерами* або *контролерами*.

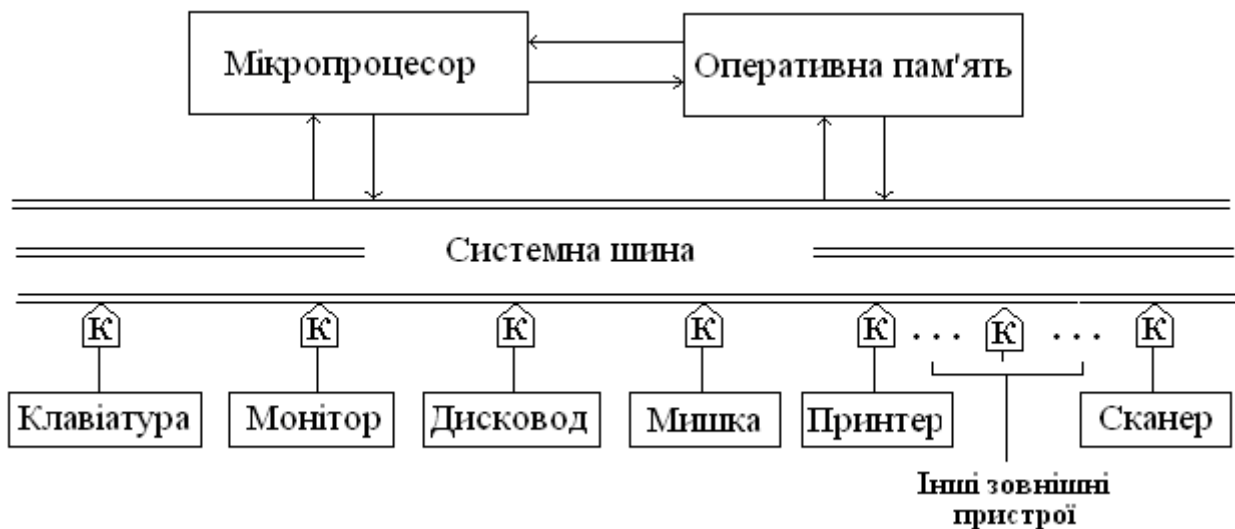


Рис. 1.2. Структурна схема сучасного IBM-сумісного комп'ютера

Конструктивно мікропроцесор і оперативна пам'ять являють собою мікросхеми, що разом із системною шиною, адаптерами і контролерами розташовані на центральній платі комп'ютера, називаною звичайно *материнською платою*. Материнська плата разом із блоком живлення і пристроями зовнішньої пам'яті розміщається усередині так званого *системного блоку*. Корпус системного блоку може мати різні розміри й орієнтацію. На задній стінці системного блоку розміщені *рознімні з'єднання* для підключення

до *паралельних і послідовних портів* (за допомогою спеціальних кабелів) зовнішніх пристроїв (дисплея, принтера й ін.).

Системний блок, дисплей (монітор) і клавіатура є обов'язковими компонентами будь-якого персонального комп'ютера. Інші пристрої називаються додатковими пристроями і можуть входити або не входити до складу конкретного ПК. Втім, вміст системного блоку також може змінюватися в досить широких межах. Це стосується як переліку включених у системний блок пристроїв, так і характеристик окремих з них.

1.5. Програмне забезпечення ПК, його класифікація

За призначенням програмне забезпечення поділяється на системне і прикладне (див. рис. 1.3).



Рис. 1.3. Класифікація програмного забезпечення ПК

Системне програмне забезпечення призначене для загального керування роботою ПК (операційні системи); створення і підтримки виконання прикладних програм кінцевого користувача (системи програмування); надання користувачеві сервісних послуг (сервісні програми); підтримки апаратної частини ПК (засоби діагностики і тестування) . Прикладами операційних систем можуть бути MS-DOS і Windows; прикладами систем програмування – Borland Pascal, Visual Basic, Delphi; прикладами сервісних програм - операційна оболонка Norton Commander, програми-архіватори WinZip і WinRAR, антивірусні програми DrWeb, AVP, Avast, пакет утиліт Norton Utilities. Зокрема, програма-детектор AntiViral Toolkit Pro (AVP), яка будучи 32-розрядним додатком для роботи в середовищі ОС Windows, має зручний інтерфейс, а також одну з найбільших у світі антивірусну базу. Ця програма шукає і вилучає

найрізноманітніші віруси, у тому числі:

- поліморфні, або віруси, що самошифруються;
- стелс-віруси, або віруси-невидимки;
- нові віруси для Windows;
- макровіруси, що заражають документи Word і таблиці Excel.

У процесі виконання програма AVP перевіряє оперативну пам'ять, файли, системні та завантажувальний сектори, а також таблицю поділу диска. Здійснює пошук вірусів в упакованих й архівних файлах, на гнучких, локальних, мережових і CD-ROM дисках. Крім того, вона контролює файлові операції системи у фоновому режимі, виявляє вірус до моменту реального зараження системи, а також за допомогою евристичного модуля визначає невідомі віруси.

Однією з найпопулярніших програма-архіваторів на сьогодні є програма *WinRar*, яка дозволяє працювати з архівними файлами різних типів, включаючи *.zip*; *.arj*; *.rar*.

Для користувачів ПК, що не є професіоналами в області ПК і програмування, найбільш важливим є прикладне програмне забезпечення. Воно призначене для вирішення певних класів задач користувача і підрозділяється на *окремі прикладні програми і пакети прикладних програм (ППП)*.

При цьому окремі прикладні програми призначені для рішення конкретних задач в умовах конкретного підприємства або установи. Як правило, такі програми доступні користувачам з будь-якою кваліфікацією в області обчислювальної техніки, однак вони забезпечують виконання тільки обмеженого кола робіт і не можуть бути використані в умовах іншого користувача.

Пакети прикладних програм відрізняються значно більш широкою областю застосування і вимагають визначених навичок користувача ПК. Це стосується, насамперед, пакетів загального призначення й інтегрованих пакетів програм. І ті, і інші призначені для рішення типових задач обробки даних різних типів (тексту, числових даних, графічних зображень і т.п.), причому інтегровані пакети передбачають можливість спільної обробки даних різних типів, а пакети загального призначення призначені, як правило, для обробки даних одного типу. Для роботи з такими програмами потрібні визначені навички користувача ПК, однак самі програми є найбільше гнучкими інструментами для непрограміста.

Що стосується проблемно-орієнтованих пакетів прикладних програм, то вони служать для вирішення визначеного кола задач з цілком визначеної області. Від окремих прикладних програм їх відрізняє більш широка область застосування і можливість налаштування на конкретного користувача.

Приведена вище класифікація є традиційною для інформатики і програмування. Поряд з нею застосовуються й інші, трохи спрощені, класифікації, розраховані на кінцевого користувача ПК. При цьому зазвичай виділяють:

- *програми офісного призначення* (текстові редактори, табличні процесори, програми підготовки презентацій, персональні інформаційні менеджери, програми планування, програми-перекладачі і програми-словники та ін.);

- *програми економічного призначення* (бухгалтерські програми, програми фінансового аналізу, програми підрахунку грошей, програми автоматизації банків, правові бази даних і ін.);

- *видавничі системи*, у т.ч. системи підготовки специфічних документів, наприклад, математичних;

- *програми комп'ютерної графіки, відео й анімації;*

- *системи керування базами даних;*

- *системи автоматизованого проектування;*

- *математичні і статистичні програми;*

- *комп'ютерні ігри;*

- *навчальні програми, електронні довідники й енциклопедії;*

- *авторські програми для створення навчальних мультимедіа-програм, Web-сторінок тощо;*

- *комунікаційні програми.*

1.6. Загальна характеристика операційних систем

Операційна система – це сукупність програмних засобів, призначених для управління апаратними і програмними ресурсами ПК, організації процесів оброблення інформації і взаємодії між різними апаратними пристроями, програмним забезпеченням та користувачем.

Таблиця 1.1. Основні функції операційних систем

№ пп	Функція	Пов'язані поняття
1	Забезпечення інтерфейсу користувача	Командний або графічний інтерфейс
2	Забезпечення автоматичного запуску ПК	BIOS
3	Організація файлової системи	Принципи розміщення інформації на дисках та доступу до неї
4	Обслуговування файлової системи	Створення каталогів (папок) і файлів (документів); копіювання, переміщення, вилучення і т.п.
5	Управління роботою прикладних програм	Встановлення та видалення програм, забезпечення одночасної роботи кількох програм та спільного використання ресурсів
6	Забезпечення взаємодії прикладних програм з апаратною частиною	Драйвери пристроїв
7	Обслуговування ПК	Перевірка дисків, стиснення (дефрагментація) і т.п.

Операційні системи можуть виконувати і додаткові функції (підтримка локальної мережі, забезпечення доступу до Internet, захист інформації тощо), а також надавати можливість виконувати найбільш поширені практичні задачі без додаткового програмного забезпечення (робота з текстом або графікою, калькулятор тощо).

Найбільш відомі операційні системи – MS-DOS, Windows, Unix (Linux).

1.7. Інтерфейс та основні принципи роботи з Windows

Типовим представником операційних систем сімейства Windows є *Windows XP*. Вона забезпечує виконання всіх основних і додаткових функцій, вказаних вище, підтримує багатозадачний режим роботи і зручний для користувача графічний інтерфейс.

Основним елементом інтерфейсу користувача *Windows XP* є фоновий екран, який називається *Робочим столом*. На *Рабочем столе* розміщуються окремі документи, папки з документами, інші об'єкти та інструменти для роботи з ними.

Таблиця 1.2. Основні об'єкти та інструменти Windows

№ пп	Об'єкт або інструмент	Пояснення
1	Значок	Графічне зображення певного об'єкта
2	Документ	Іменована сукупність інформації певного виду (текст, графічне зображення, електронна таблиця і т.д.)
3	Папка	Іменована сукупність документів та інших папок
4	Прикладна програма (додаток)	Будь-яка програма або група програм, виконання яких ініціюється однією командою
5	Ярлик	Посилання на деякий об'єкт. Незалежно від того, де знаходиться сам об'єкт, для звернення до нього досить двічі клацнути по відповідному ярлику
6	Вікно	Прямокутна частина екрану, в якій може відображатись додаток, документ, вміст папки чи повідомлення. Розрізняють вікна папок, прикладних програм, документів, повідомлень та діалогові вікна
7	<i>Мой компьютер</i>	Засіб доступу до дискових пристроїв та інших ресурсів комп'ютера
8	<i>Корзина</i>	Інструмент, що складається зі спеціальної папки для тимчасового зберігання видалених користувачем об'єктів та механізму забезпечення можливості повернення видаленого об'єкта на своє місце (до тих пір, поки <i>Корзина</i> не очищена спеціальною процедурою)
9	<i>Сетевое окружение</i>	Засіб доступу до інших комп'ютерів локальної комп'ютерної мережі
10	<i>Буфер обмена</i>	Інструмент, який забезпечує можливість тимчасового розміщення певної інформації в спеціальній області оперативної пам'яті та подальшої вставки цієї інформації в інші місця

Вся інформація відображається в вікнах. Одночасно може бути відкрито кілька вікон, одне з яких є *активним*, тобто готовим до сприйняття команд користувача. Активне вікно завжди розташоване поверх інших, його заголовок виділений кольором. Для активізації іншого вікна досить клацнути мишкою на ньому або на його значку на *Панелі задач*.

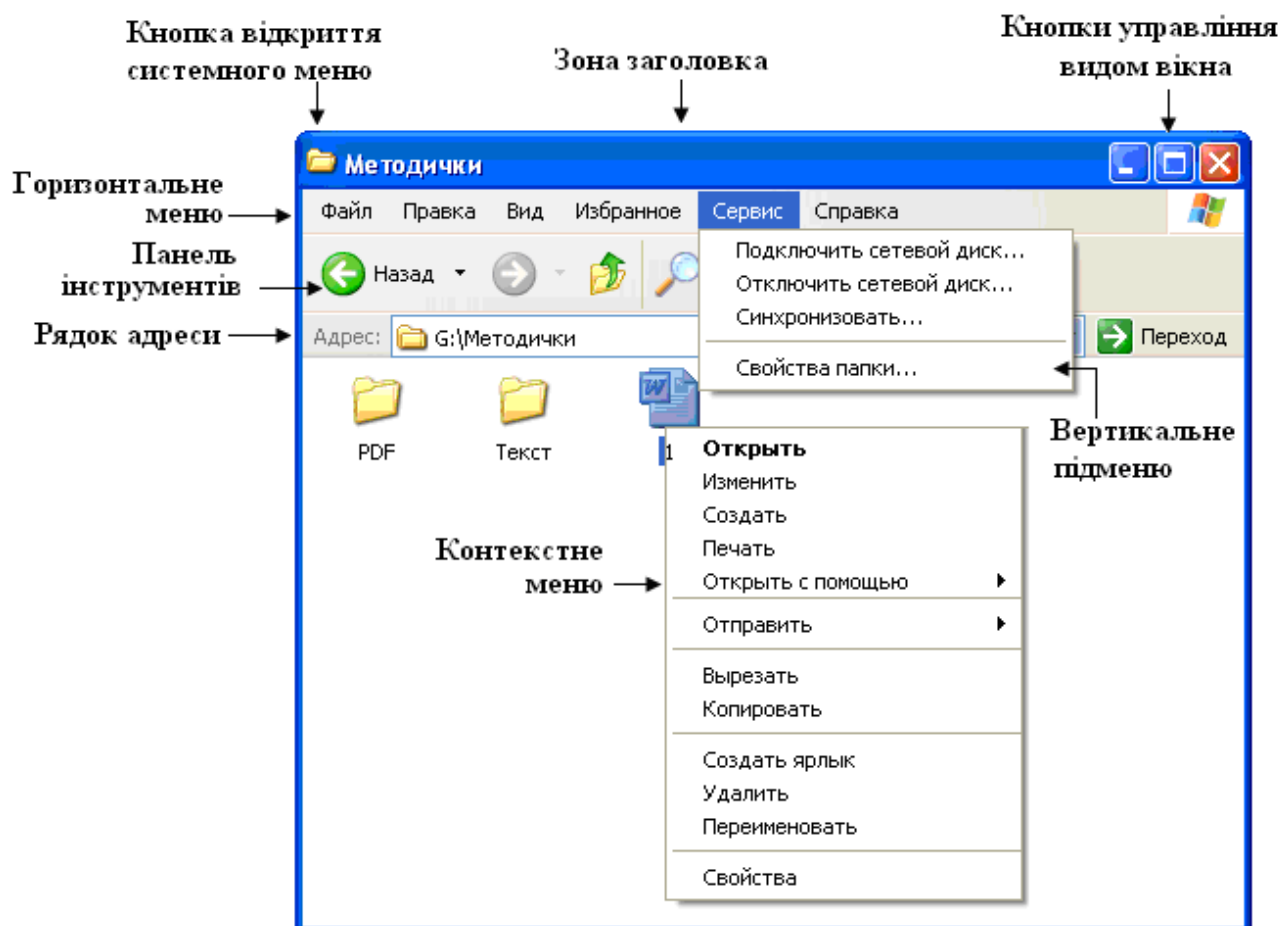


Рис. 1.4. Приклад вікна папки

Користувач для виконання певних дій в Windows використовує систему меню (головне меню, системне меню, контекстне меню, горизонтальне меню та вертикальні підменю), панелі інструментів, спеціальні комбінації клавіш та деякі маніпуляції мишкою (клацання лівою (ЛКМ) або правою (ПКМ) кнопкою мишки, подвійне клацання ЛКМ, переміщення об'єктів з одного місця в інше при натиснутій ЛКМ або ПКМ).

1.8. Техніка виконання основних операцій з об'єктами Windows

Операції виконуються, як правило, по відношенню до попередньо виділеного об'єкта (об'єктів). Одна й та ж операція може бути виконана різними способами. В таблиці 1.3 наведені деякі з них.

Таблиця 1.3. Типові способи виконання операцій з об'єктами Windows

№ пп	Операція	Спосіб виконання
1	Виділення окремого об'єкта	Клацнути ЛКМ на значку об'єкта
2	Виділення кількох об'єктів, які не розташовані поруч	Перший об'єкт як звичайно, кожний наступний - виділяти при натиснутій клавіші Ctrl
3	Виділення кількох об'єктів, які розташовані поруч	“Намалювати” контур навколо відповідної області шляхом переміщення мишки при натиснутій ЛКМ
4	Створення папок, документів та ярликів	Клацнути ПКМ на вільному місці <i>Робочого стола</i> або папки і вибрати в контекстному меню відповідну дію
5	Створення документів	Відкрити відповідну прикладну програму і після введення інформації зберегти документ на диску
6	Відкриття папки	Двічі клацнути ЛКМ на значку (ярлику) папки
7	Запуск прикладної програми	Двічі клацнути ЛКМ на значку (ярлику) програми або клацнути ЛКМ на назві програми в головному меню Windows
8	Закриття папки, припинення роботи програми	Клацнути ЛКМ на кнопці закриття відповідного вікна
9	Видалення об'єктів	Натиснути клавішу Del або відпрацювати команду <i>Удалить</i> контекстного меню, меню папки
10	Перейменування об'єктів	Відпрацювати команду <i>Переименовать</i> контекстного меню або меню папки
11	Переміщення та копіювання об'єктів	Див. рис. 1.5
12	Пошук папок та документів на дисках	Відпрацювати команду <i>Поиск</i> в головному меню Windows і уточнити параметри пошуку

На рис. 1.5 представлені варіанти копіювання та переміщення об'єктів з використанням *Буфера обмену*.

Зручним способом виконання операцій копіювання та переміщення, а також операції створення ярлика, є використання методу “перетягнути і залишити”. Він полягає в переміщенні за допомогою мишки значка об'єкта з вихідної папки в папку приймач (див. рис. 1.6). Зауважимо, що перетягування об'єктів мишкою при натиснутій ЛКМ без клавіш CTRL або SHIFT призводить до переміщення, якщо вихідна папка і папка-приймач знаходяться на одному диску і до копіювання – в іншому разі.

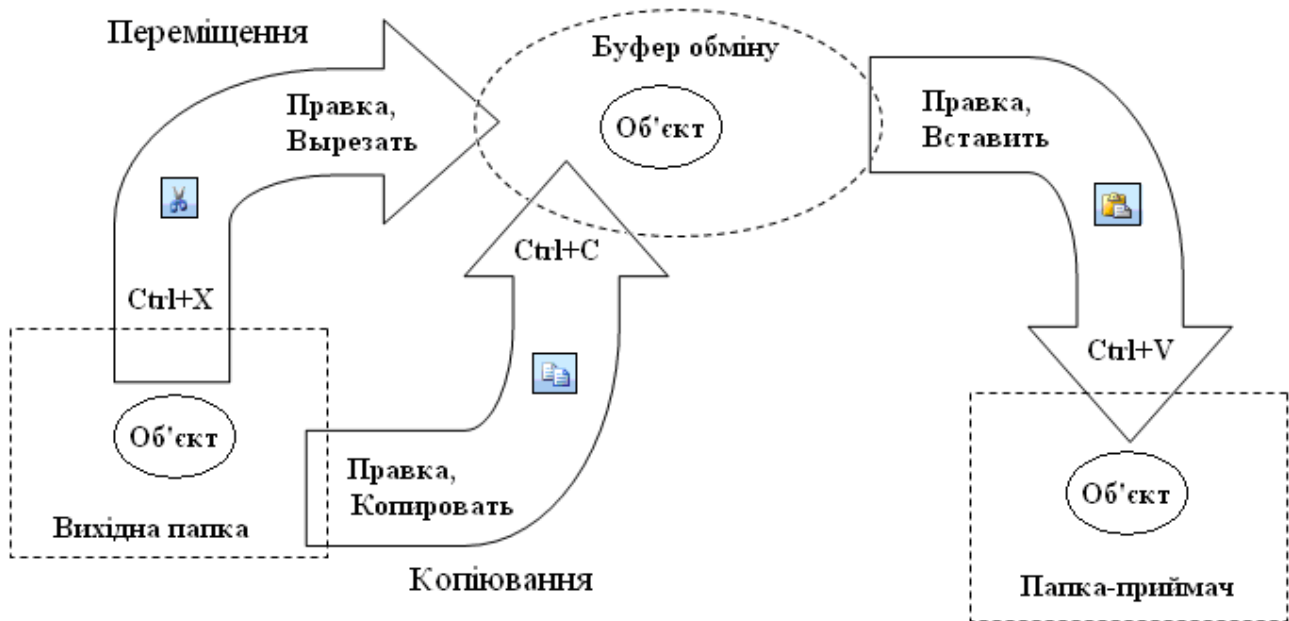


Рис. 1.5. Варіанти копіювання і переміщення об'єктів

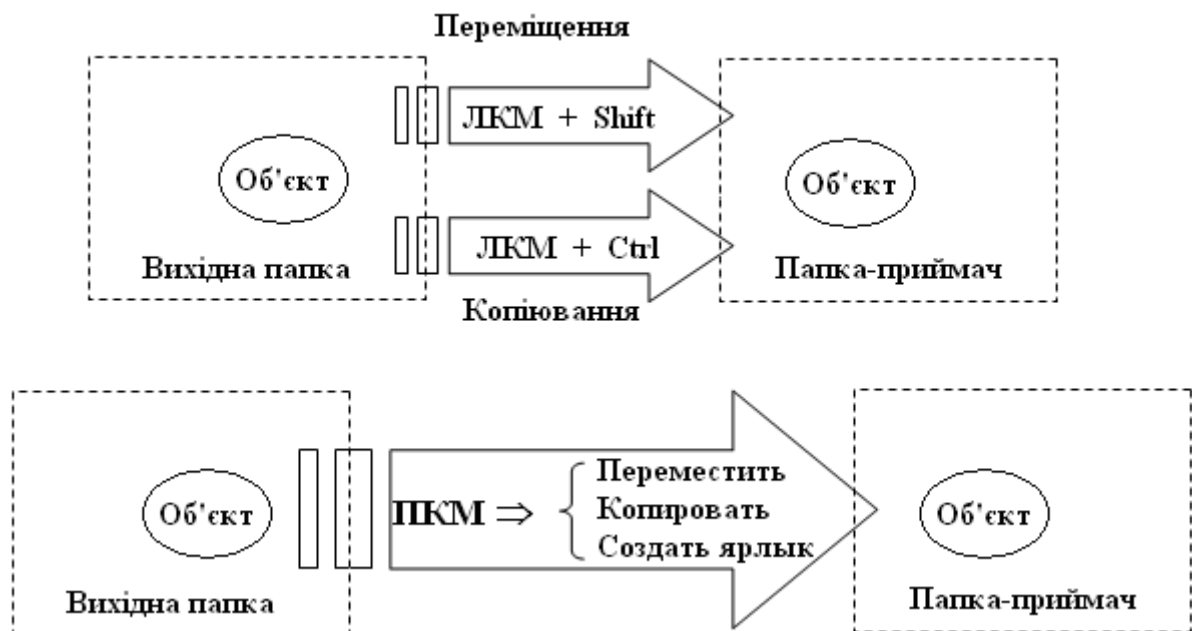


Рис. 1.6. Використання методу “перетягнути і залишити” для копіювання і переміщення об'єктів

Питання теми 1, які винесені на самостійну проробку і не включені в конспект

1. Архітектура та характеристика основних пристроїв ПК
2. Використання службових і стандартних програм Windows

Контрольні запитання до теми 1

1. Розкрийте поняття “інформація”. Наведіть конкретні приклади інформації.
2. Вкажіть основні властивості інформації.
3. Розкрийте поняття “інформаційний процес”. Наведіть конкретні приклади здійснення інформаційного процесу.
4. Розкрийте поняття “інформаційна система”. Перерахуйте основні елементи інформаційної системи.
5. Розкрийте поняття “інформаційна технологія”.
6. Розкрийте поняття “інформатика”. Вкажіть об’єкт, предмет та основний інструмент інформатики.
7. Яке значення для інформатики має двійкова система числення?
8. Поясніть поняття “біт”, “байт”, “Кбайт”, “Мбайт”.
9. Сформулюйте основні принципи, покладені в основу функціонування ЕОМ (принципи фон Неймана).
10. Наведіть класичну схему ЕОМ та поясніть, як вона працює.
11. Поясніть поняття “hardware” і “software”, вкажіть їх взаємозв’язок.
12. Перерахуйте основні пристрої сучасного комп’ютера та вкажіть їх призначення.
13. Порівняйте різні види пам’яті: оперативну, постійну, кеш-пам’ять (за призначенням, розмірами, залежністю від енергоживлення).
14. Охарактеризуйте призначення зовнішньої пам’яті. Назвіть основні види носіїв та порівняйте їх технічні характеристики.
15. Назвіть основні технічні характеристики (параметри) сучасних ПК, вкажіть яких значень вони можуть набувати.
16. Назвіть основні технічні характеристики (параметри) моніторів, вкажіть яких значень вони можуть набувати.
17. Як класифікується програмне забезпечення ПК?
18. Наведіть приклади прикладного програмного забезпечення.
19. Охарактеризуйте призначення та основні функції операційних систем.
20. Поясніть поняття Робочий стол, папка, документ, прикладна програма, ярлик.
21. Для чого використовуються такі інструменти, як Мой комп’ютер, Корзина, Сетевое окружение?
22. Поясніть поняття Буфер обмѐна. Для чого використовується цей інструмент?
23. Які види меню ви знаєте? Як вони викликаються і для чого використовуються?
24. Яке значення для Windows мають вікна? Якими вони бувають?
25. Опишіть структуру вікна папки в Windows.
26. Які маніпуляції мишкою найчастіше використовуються при роботі в Windows? Наведіть приклади їх застосування.
27. Опишіть процедуру створення папок та документів на робочому столі. Чим відрізняється від описаної процедури створення папок та документів в папці, яка вже є на диску?
28. Які варіанти створення ярликів ви знаєте?
29. Які варіанти перегляду змісту папок ви знаєте?
30. Охарактеризуйте систему меню папок. Як можна змінювати зовнішній вигляд інформації в папках?
31. Охарактеризуйте панель інструментів папок. Як вона викликається та які кнопки містить?
32. Перерахуйте основні варіанти пошуку документів та папок на дисках.
33. Опишіть процедуру автоматичного пошуку документів та папок на дисках.
34. Як в Windows виконуються операції виділення окремих об’єктів, групи об’єктів, всіх об’єктів папки? Як можна відмінити виділення об’єктів?

35. Які ви знаєте способи копіювання окремих об'єктів?
36. Чим відрізняється операція переміщення об'єктів від копіювання? Опишіть основні способи виконання цієї операції.
37. Як в Windows виконуються операції перейменування об'єктів?
38. Опишіть процедуру роботи зі стандартним засобом Корзина.
39. Які способи запуску на виконання прикладних програм ви знаєте ?
40. Назвіть та коротко охарактеризуйте:
основні стандартні програми Windows;
основні службові програми Windows.

Тема 2. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ОСНОВА ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

2.1. Інформаційні технології обробки даних

Інформаційні системи функціонують на основі певної інформаційної технології. Загальноприйнятим тлумаченням терміну *технологія* є таке: *технологія* — це сукупність методів обробки, виготовлення, зміни стану, форми сировини, матеріалів або напівфабрикатів, які здійснюються в процесі виробництва певної продукції. Для інформаційних технологій (ІТ) характерним є те, що і вихідною сировиною і остаточною продукцією в них є інформація. Тому основними складовими інформаційної технології є принципи обміну інформацією; методи її накопичення, збереження і переробки; використовувані носії і технічні пристрої. Звернемо увагу на ще одну принципову різницю між інформаційними та виробничими технологіями: виробничі технології містять в основному рутинні операції, а інформаційним технологіям притаманні елементи творчого характеру, які не піддаються регламентації та формалізації.

У своєму розвитку інформаційні технології пройшли ряд стадій, обумовлених засобами передачі й обробки інформації (див. табл. 2.1).

Таблиця 2.1. Стадії розвитку інформаційних технологій

Стадії ІТ	Засоби обробки інформації	Засоби зв'язку
"Ручна" ІТ	Ручка, рахівниця, бухгалтерська книга	Передача пакетів, листів
"Механічна" ІТ	Друкуюча машинка, диктофон, механічні рахункові машини	Телефон, телеграф
"Електрична" ІТ	Електрична друкуюча машинка, копіювальні машини, калькулятори	Телефон, телеграф, телетайп
Комп'ютерна ІТ	ПЕОМ	Комп'ютерні мережі

На кожній з цих стадій засоби передачі й обробки інформації відігравали вирішальну роль у функціонуванні відповідної інформаційної системи. Однак, саме поняття "інформаційна технологія" виникло тільки після появи

персональних комп'ютерів і комп'ютерних мереж, коли комп'ютер перетворився в настільний інструмент управлінського працівника, а наявне програмне забезпечення дало можливість автоматизувати велику частину рутинної роботи і визволити основний час для творчої роботи. Інформаційні технології фактично стали комп'ютерними технологіями. Тому надалі під *інформаційною технологією* ми будемо мати на увазі, насамперед, *сукупність конкретних технічних і програмних засобів, за допомогою яких виконуються різноманітні операції по обробці інформації*.

Таким чином, основу сучасних інформаційних технологій складають розподілена комп'ютерна техніка, "дружнє" програмне забезпечення, розвинені засоби комунікацій.

Для сучасних ІТ характерні такі особливості:

- безпаперовий процес обробки інформації (на папері фіксується тільки остаточний варіант документа);
- наскрізна інформаційна підтримка на всіх етапах обробки інформації за рахунок використання інтегрованих баз даних;
- робота користувача в режимі маніпулювання даними (без програмування);
- можливість колективної обробки інформації на основі мереж.

2.2. Класифікація інформаційних технологій

Інформаційні технології можна класифікувати за різними ознаками, зокрема, за *типом оброблюваної інформації* (див. рис. 2.1).

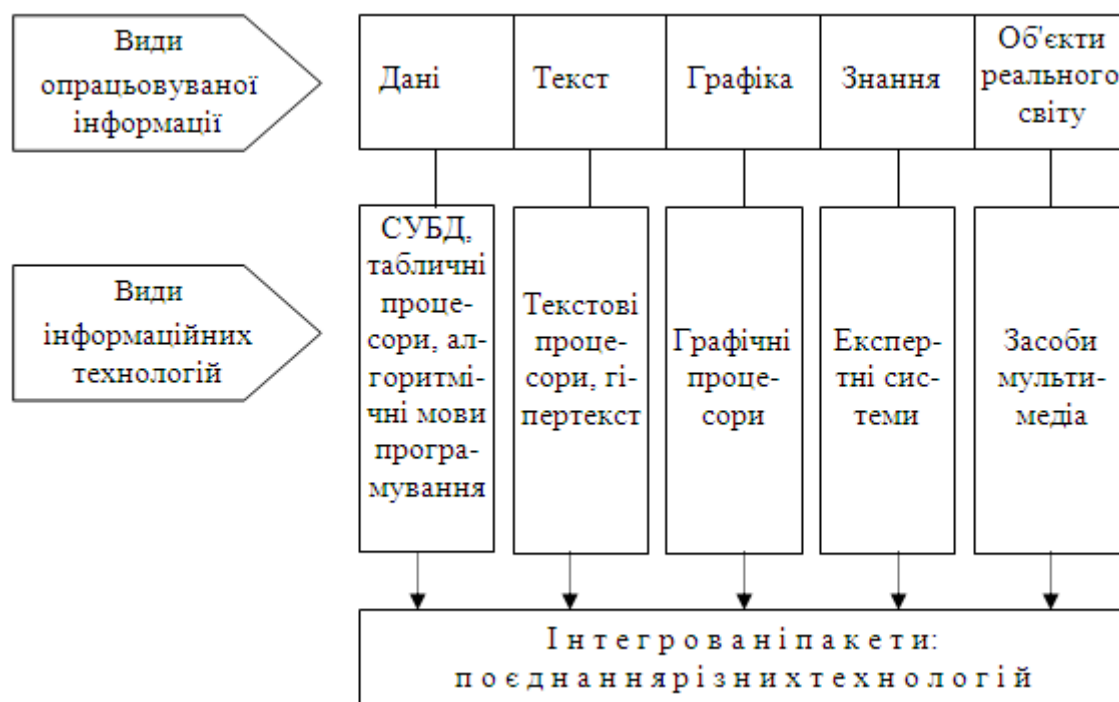


Рис. 2.1. Класифікація інформаційних технологій

Наведена вище класифікація орієнтована на історично сформовану класифікацію стандартних програмних засобів, призначених для обробки певних видів інформації.

В той же час за організацією роботи користувача розрізняють *інформаційні технології з командним інтерфейсом, графічним інтерфейсом (WIMP-інтерфейс) і мовним інтерфейсом (SILK-інтерфейс)*. Виділяють також *спеціалізовані й інтегровані інформаційні технології; мережні інформаційні технології і технології, реалізовані на різних платформах (у смислі використовуваних технічних засобів) і т.д.*

2.3. Мережні інформаційні технології

Під комп'ютерною мережею зазвичай розуміють кілька комп'ютерів, з'єднаних між собою кабелем, через телефонну мережу або в інший спосіб. Кількість комп'ютерів у мережі й відстані між окремими комп'ютерами можуть коливатися в досить широких межах. У зв'язку із цим розрізняють локальні, розподілені й глобальні мережі.

Локальна мережа - це мережа, що з'єднує персональні комп'ютери й інше офісне обладнання, яке розташоване, як правило, в одному або декількох сусідніх будинках і належить одній установі (підрозділу). Основне призначення локальної мережі полягає в забезпеченні колективної обробки даних, включаючи спільне використання програм; обмін даними; спільне використання принтерів, модемів і інших пристроїв. Для роботи локальної мережі крім персональних комп'ютерів потрібно спеціальне мережне устаткування й мережне програмне забезпечення.

Розподілена мережа за своєю ідеологією близька до локальної мережі, але окремі комп'ютери в ній можуть розташовуватися на значному віддаленні від інших. У розподілені мережі звичайно поєднують локальні мережі різних установ того самого відомства. Особливістю розподілених мереж є застосування спеціальних ліній зв'язку (у першу чергу, телефонних каналів), що використовуються, як правило, і для інших цілей.

У розподілених мережах часто реалізується не тільки колективний доступ до даних і ресурсів, але й обмін службовою інформацією (директивною та звітною), яка має конфіденційний характер. Для позначення таких мереж останнім часом використовується термін *корпоративні мережі*.

Локальні, розподілені й, тим більше, корпоративні мережі є мережами закритого типу в тому розумінні, що доступ до їхніх ресурсів обмежений колом осіб, що мають безпосереднє відношення до даного об'єднання, установи або підрозділу. Поряд із цим є й *глобальні мережі*, доступ до яких відкритий практично будь-якому користувачеві комп'ютера. Найбільш відомою глобальною мережею є мережа *Internet*. Саме із глобальною мережею *Internet* пов'язані найбільш значні зміни у використанні інформаційних ресурсів людства.

*Питання теми 2, які винесені на самостійну проробку
і не включені в конспект*

1. Локальні комп'ютерні мережі
2. Технологія використання інформаційних ресурсів глобальної комп'ютерної мережі

Контрольні запитання до теми 2

1. Перерахуйте основні стадії розвитку інформаційних технологій. Які саме фактори були визначальними на кожній зі стадій?
2. За якими ознаками можна класифікувати інформаційні технології і як саме?
3. Як класифікуються інформаційні технології за видом опрацьовуваної інформації?
4. Що таке комп'ютерна мережа? Якими бувають комп'ютерні мережі?
5. В яких випадках доцільно використовувати локальні мережі. Обґрунтуйте відповідь.
6. Поясніть поняття *сервер, робоча станція, однорангова мережа, багаторангова мережа*.
7. Перерахуйте базові топології локальних мереж та коротко охарактеризуйте їх.
8. Охарактеризуйте основні схеми опрацювання даних в мережах з виділеним сервером.
9. Які специфікації комп'ютерних мереж вам відомі? Чим вони відрізняються?
10. Які апаратні засоби необхідні для організації комп'ютерних мереж?
11. Охарактеризуйте програмне забезпечення комп'ютерних мереж.
12. Поясніть поняття *Internet*.
13. Яке значення для Internet мають *принцип комутації пакетів* та сімейство *протоколів TCP/IP* ?
14. Перерахуйте основні служби Internet та поясніть, для чого вони використовуються.
15. Поясніть поняття *провайдер*. Яке значення для користувача має провайдер?
16. Які варіанти підключення до Internet ви знаєте?
17. Що таке програма-броузер? Які програма-броузери ви знаєте?
18. Опишіть процедуру роботи з однією з програм-броузерів.
19. Наведіть приклади пошукових запитів (пошук точної фрази, пошук за окремими фрагментами фрази, виключення окремих слів, пошук за синонімами, пошук на вибір з кількох варіантів тощо).
20. Поясніть поняття *пошукова машина, Web-каталог, контекстний пошук, спрямований пошук*.

Тема 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ НА ОСНОВІ MICROSOFT OFFICE

3.1. Пакет прикладних програм Microsoft Office, основні складові та сфери застосування

Стандартні інформаційні технології можуть бути реалізовані за допомогою прикладних програмних програм пакету Microsoft Office, який відрізняється єдиними підходами до інтерфейсу програм, простою процедурою їх сумісного використання. Основними програмами пакету є *Microsoft Word* ,

Microsoft Excel, Microsoft Access, Microsoft PowerPoint.

Microsoft Word (*MS Word* або просто *Word*) - це текстовий редактор (текстовий процесор), за допомогою якого можна швидко і якісно підготувати документ будь-якої складності. В *Word* реалізовані всі без виключення традиційні операції над текстом, включаючи найхарактерніші для текстових редакторів функції:

- початкове формування тексту в пам'яті ПК, тобто його введення за допомогою клавіатури;
- редагування тексту, тобто внесення смислових змін, виправлення помилок і інші подібні операції (вставка, видалення і заміна окремих символів, слів або текстових фрагментів, в т.ч. з інших текстових документів);
- форматування документа, тобто підготовка до друку на папері відповідно до певних вимог по оформленню (застосування різних шрифтів і стилів, виділення заголовків, вирівнювання рядків тексту, встановлення розмірів поля друку, формування абзаців, розбивка на сторінки і т.п.);
- збереження тексту на магнітних носіях;
- друк тексту на папері.

Крім того, *Word* дає можливість використати специфічні переваги Windows-програм:

- зв'язування і впровадження об'єктів, підготовлених іншими програмами (таблиць, рисунків, формул і т.п.);
- можливість надання документам розширених назв на українській (російській) мові;
- використання графічного інтерфейсу, зокрема, панелей інструментів;
- можливість переміщення фрагментів документа за допомогою методу "перетягнути і залишити";
- використання масштабованих шрифтів і т.п.

Слід також відзначити наявність в *Word* готових шаблонів і стилів форматування; *Майстрів* для підготовки різних текстових документів.

Microsoft Excel (*MS Excel* або просто *Excel*) є представником табличних процесорів. *Електронні таблиці* або *табличні процесори* — це спеціальні пакети програм, призначені для опрацювання інформації, поданої у виді таблиць. При цьому частина клітин таблиці містить вихідну або первинну інформацію, а інша частина — похідну інформацію, одержувану шляхом виконання деяких операцій над первинною інформацією. Одна з найбільше привабливих властивостей електронних таблиць полягає в тому, що будь-яка зміна первинної інформації призводить до автоматичного оновлення похідної інформації. Це дуже зручно при розв'язанні задач моделювання, у яких шукаються відповіді на питання типу “ що буде, якщо... ”. Більшість табличних процесорів мають також засоби створення графіків і діаграм, великий набір стандартних функцій (математичних, статистичних і інших); дають можливість створювати й експлуатувати невеликі бази даних.

Microsoft Access (*MS Access* або просто *Access*) - це система управління

реляційними базами даних (СУБД), яка призначена для роботи в середовищі Windows і входить до складу професійної версії Microsoft Office.

Інформація в СУБД *Access* зберігається у вигляді файлів типу *.mdb*. При цьому кожний файл містить сукупність об'єктів бази даних, включаючи *таблиці, запити, форми, звіти, макроси, модулі*.

Працюючи з *Access* можна вирішувати такі задачі:

- створювати бази даних;
- вводити, змінювати і знаходити потрібні дані;
- знаходити підмножини даних за умовами, що задаються;
- створювати форми і звіти;
- автоматизувати виконання стандартних задач;
- вставляти малюнки й інші об'єкти у форми і звіти;
- створювати власні цілком готові до роботи з базами даних програми, що містять меню, діалогові вікна і командні кнопки і багато чого іншого.

Microsoft PowerPoint призначена для створення та демонстрації комп'ютерних презентацій, тобто системи слайдів з певним інформаційним змістом, орієнтованих на певну аудиторію. Презентації поширені в торгово-економічній діяльності (реклама товарів чи послуг; представлення нових проектів), а також в публічних виступах науковців та політиків, при захисті курсових або дипломних робіт тощо.

Презентація в середовищі PowerPoint складається з базових елементів, які мають назву слайдів. При створенні нової презентації програма пропонує користувачу для редагування її перший слайд з певним макетом. Макет слайда – це шаблон розташування об'єктів на слайді. У макетах містяться рамки-контейнери – місцезаповнення, у яких розташований текст – заголовки слайдів або маркіровані списки, і графічний вміст слайда: таблиці, діаграми, малюнки, фігури тощо.

Робота користувача при створенні презентацій полягає в заповненні та редагуванні окремих елементів слайду.

Створені презентації за замовчуванням зберігаються у форматі *.ppt*. Такі файли при подвійному клацанні мишею на їх назві відкриваються в режимі *Обычный* з метою редагування. В той же час повністю готові демонстрації можна зберігати у вигляді файлів демонстрації з розширенням *.pps*. Такі файли відразу відкриваються в режимі показу файлів на екрані монітора.

3.2. Інтерфейс та основні принципи роботи з MS Word

Текстовий редактор *Word* завантажується звичними для Windows способами, зокрема:

- через головне меню (**Пуск, Программы, Microsoft Office, Microsoft Word**);
- разом із документом, створеним за допомогою *Word*;
- за допомогою ярлика програми на робочому столі.

Основні елементи інтерфейсу показані на рис. 3.1.

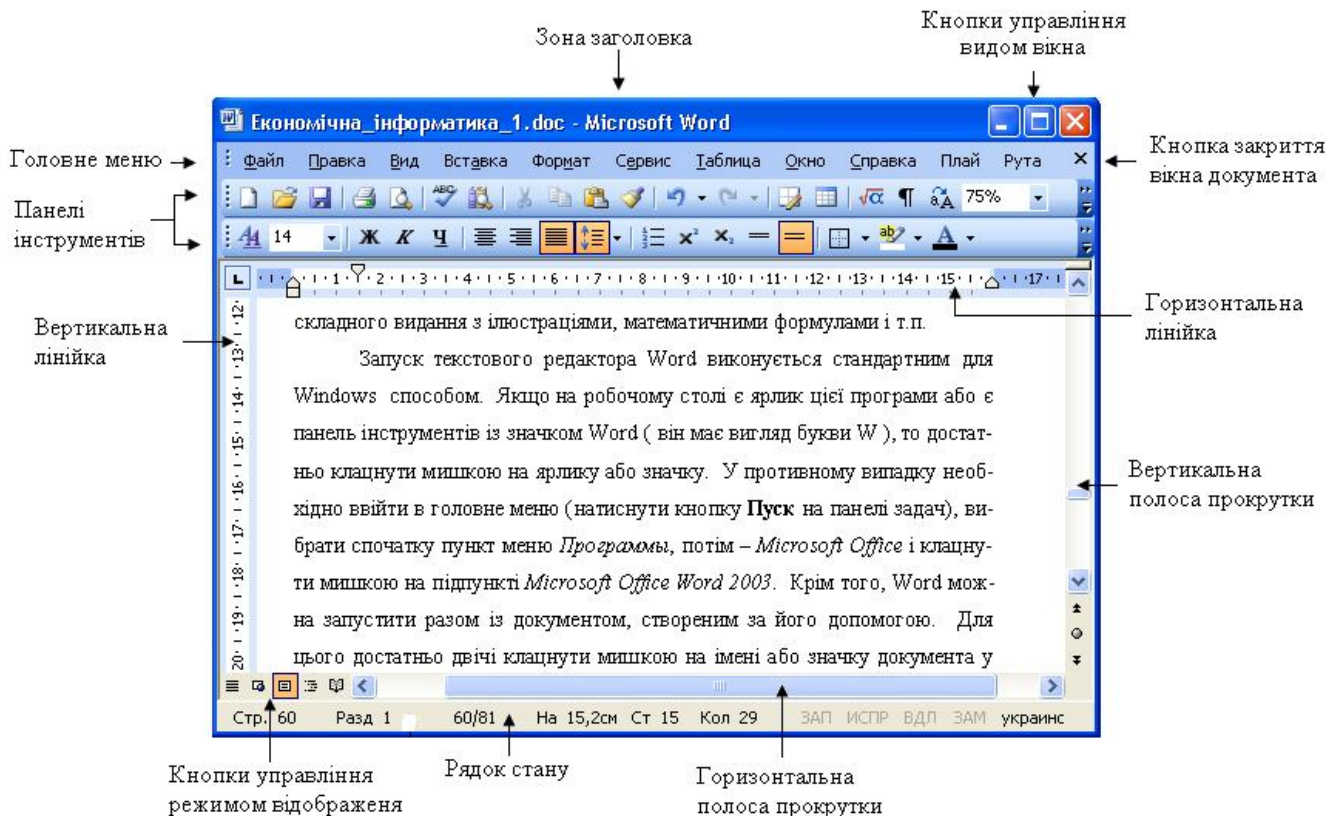


Рис.3.1. Загальний вигляд вікна Microsoft Office Word 2003

Режими відображення інформації у вікні: *Обычный, Веб-документ, Разметка страниц, Структура, Режим чтения.*

Початкове формування тексту виконується за допомогою клавіатури звичайним для текстових редакторів способом. Редагування й оформлення документа, а також інші операції проводяться за допомогою панелей інструментів або системи меню. Перший спосіб – швидший, другий – відрізняється більшими можливостями, оскільки надає користувачу право уточнення низки параметрів виконання операції.

У будь-який момент можна скористатися допомогою, натиснувши клавішу **F1** або клацнувши мишкою по відповідній кнопці панелі інструментів.

3.3. Редагування тексту

Редагування передбачає внесення смислових змін в текст, що реалізується шляхом копіювання, переміщення та видалення окремих фрагментів тексту, а також введенням нових фрагментів. При цьому під фрагментом можна розуміти символ або декілька символів, слово, речення, рисунок або інший об'єкт, рядок або кілька рядків, абзац, кілька абзаців, розділ тексту, увесь текст.

Таблиця 3.1. Типові операції редагування

№ пп	Операція	Спосіб виконання
1	Виділення окремих символів, довільної послідовності символів	“Протягнути” по символам мишкою при натиснутій
2	Виділення окремих слів	Двічі клацнути ЛКМ на будь-якому символі слова
3	Виділення окремих рядків	Клацнути ЛКМ на лівому полі поруч з рядком
4	Виділення речення	Клацнути ЛКМ у тексті речення, утримуючи затисненою клавішу Ctrl
5	Виділення абзацу	Двічі клацнути ЛКМ на лівому полі поруч з абзацом
6	Виділення кількох фрагментів, які не розташовані поруч	Перший фрагмент як звичайно, кожний наступний – виділяти при натиснутій клавіші Ctrl
7	Зняття виділення з фрагмента	Клацнути ЛКМ поза виділеним фрагментом
8	Видалення окремих символів	Клавішами Del (символ за курсором) або Backspace (символ перед курсором)
9	Видалення фрагментів	Виділити фрагмент і натиснути Del
10	Переміщення фрагментів тексту	“Перетягнути” мишкою попередньо виділений фрагмент в потрібне місце при натиснутій ЛКМ або через буфер обміну
11	Копіювання фрагментів тексту	“Перетягнути” попередньо виділений фрагмент в потрібне місце при натиснутій ЛКМ і натиснутій клавіші Ctrl або через буфер обміну
12	Скасування раніше виконаних команд	Пункти меню <i>Правка</i> ⇒ <i>Отменить</i> або за допомогою відповідної кнопки панелі інструментів <i>Стандартная</i>

3.4. Форматування тексту

Форматування передбачає зміну зовнішнього вигляду тексту.

Форматування символів - зміна шрифту, стилю написання (підкреслений, напівжирний, курсив) і розміру символів, а також установлення

спеціальних ефектів (верхній або нижній індекс, перекреслення, інтервал між символами в слові і т.д.).

Форматування абзаців - встановлення способу вирівнювання рядків абзацу (вліво, праворуч, по центру, по всій ширині); розмірів відступу в новому рядку, ширини і положення абзацу, міжрядкового інтервалу і відстані між абзацами. Крім того, можна взяти один або декілька абзаців у рамку тієї або іншої форми і заповнити рамку візерунком; оголосити декілька розташованих поруч абзаців списком-перерахуванням.

Форматувати символи й абзаци можна або за допомогою панелі інструментів *Форматирование* або за допомогою системи меню (пункт головного меню *Формат*). Перший спосіб - більш швидкий і простий у застосуванні, проте він реалізує не всі можливості Word по форматуванню символів і абзаців. Другий спосіб - універсальний, за його допомогою користувач може використовувати абсолютно всі можливості даного редактора.

При використанні системи меню використовуються підпункти *Шрифт* і *Абзац* пункту *Формат* головного меню. У результаті їх відпрацювання з'являються діалогові вікна (див. приклади на рис. 3.2-3.3), в яких і встановлюються відповідні параметри.

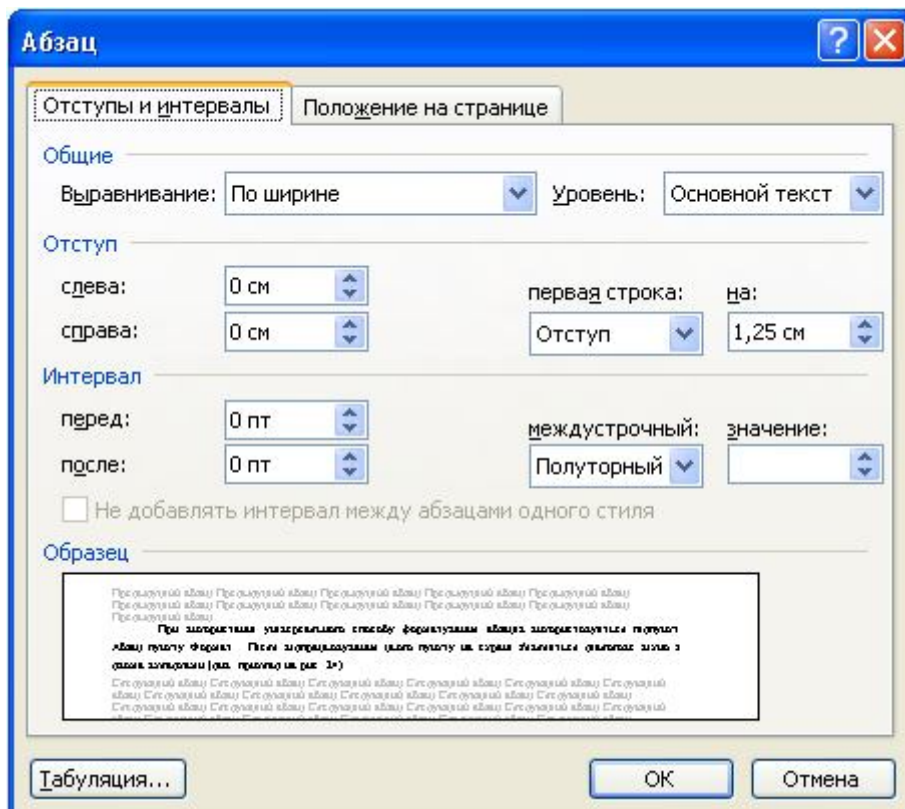


Рис. 3.2. Приклад діалогового вікна *Абзац*

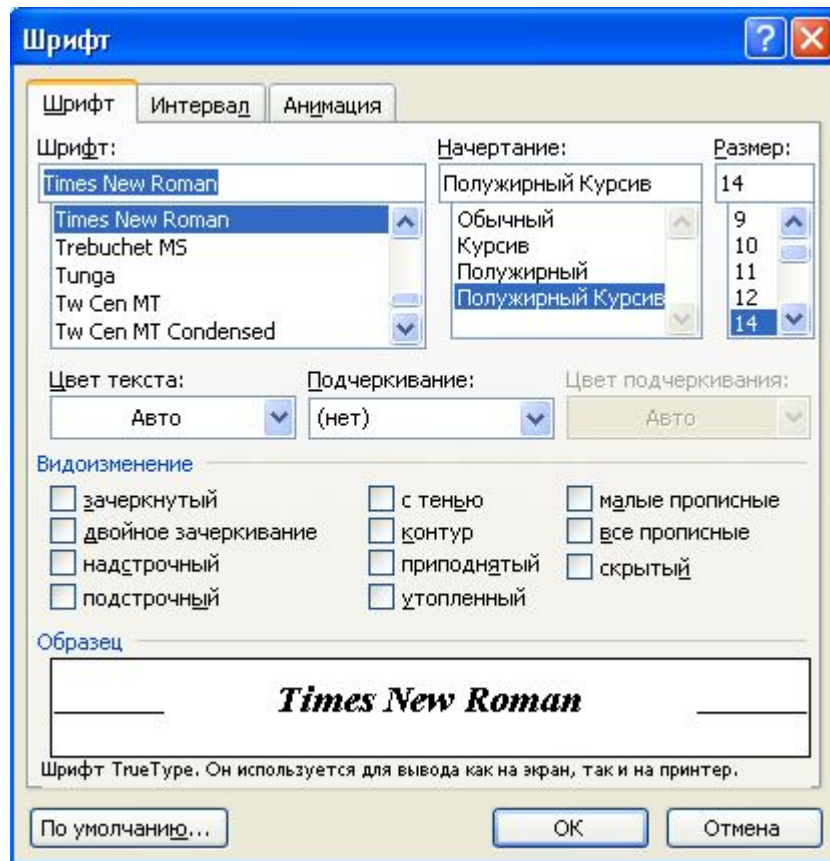


Рис. 3.3. Приклад діалогового вікна *Шрифт*

При використанні спрощеного способу форматування достатньо на панелі форматування (див. приклад на рис. 3.4) клацнути мишею на відповідній кнопці, щоб установити стиль символів, спосіб вирівнювання рядків абзацу або інші параметри тексту. Відповідна кнопка залишиться виділеною, тим самим демонструючи поточне значення даного перемикача.

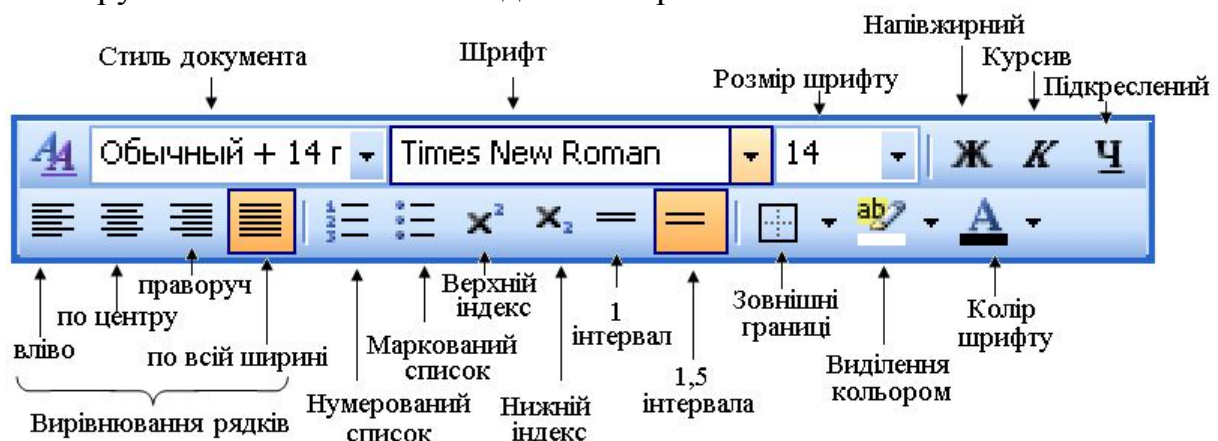


Рис. 3.4. Кнопки панелі інструментів *Форматирование*

Стиль документа, шрифт і розмір символів відображається в спеціальних демонстраційних вікнах (див. рис. 3.4). Для зміни цих параметрів використовуються списки альтернативних варіантів, що з'являються після

натиснення кнопок вигляду ▼, розташованих праворуч від відповідного демонстраційного вікна. Потрібне значення параметра встановлюється клацанням миші на необхідному варіанті.

Всі операції форматування виконуються стосовно попередньо виділеного фрагмента тексту, а у випадку відсутності такого - встановлюються параметри чергових символів, що вводяться, або параметри поточного абзацу.

До операцій по форматуванню сторінок слід віднести розмітку сторінки, вставку номерів сторінок, верхніх і нижніх колонтитулів, розривів сторінок.

Для розмітки сторінки використовується підпункт *Параметры страницы* пункту *Файл*. Після відпрацьовування цього пункту на екрані з'являється вікно (див. рис. 3.5) з трьома вкладками, на кожній із яких знаходяться демонстраційні вікна (*Образец*), кнопки *ОК*, *Отмена*, *По умолчанию* і перемикач *Применить*, який визначає область дії встановлених параметрів (до всього документа чи до кінця документа).

На вкладці *Размер бумаги* можна вибрати один із стандартних розмірів або встановити власні розміри (*Ширина*, *Высота*) аркуша паперу, задати порядок подачі паперу, а також встановити деякі параметри друку, наприклад, режим чорнового друку або режим фоновому друку.

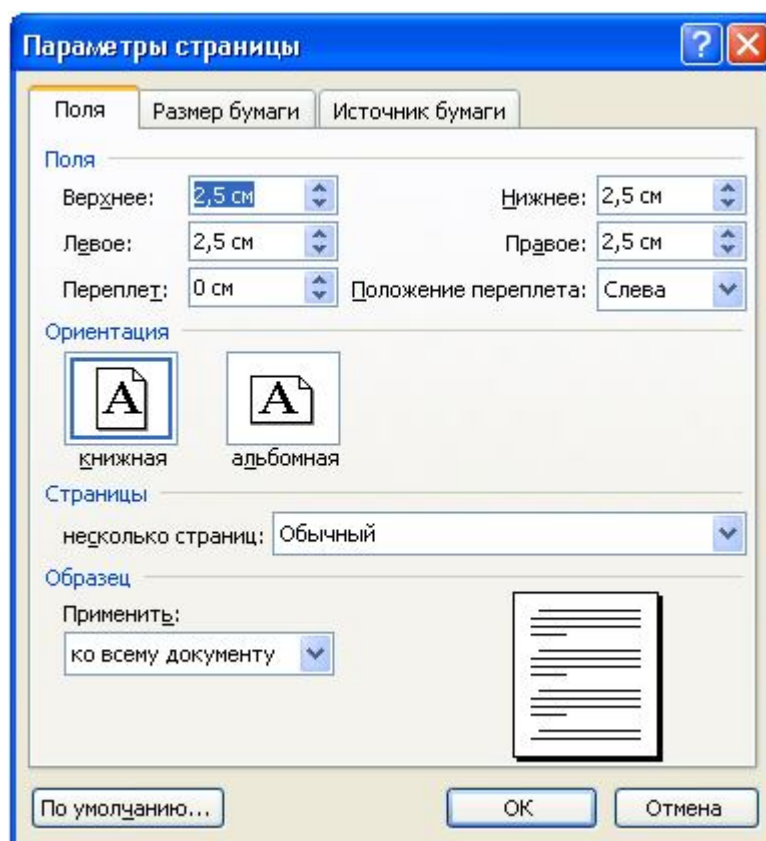


Рис. 3.5. Приклад діалогового вікна *Параметры страницы*

Для вставки примусового розриву сторінки (переходу до нової сторінки при незакінченій поточній сторінці) використовується підпункт *Разрыв* пункту

Вставка. При цьому з'являється діалогове вікно, що дозволяє уточнити характер розриву (початок нової сторінки, нової колонки, нового розділу). Потрібний варіант встановлюється клацанням миші, після чого досить натиснути кнопку ОК для вставки розриву.

3.5. Інші можливості Word

Текстовий редактор Word надає користувачеві і інші можливості, зокрема, такі:

- створення та оформлення таблиць;
- пошук і заміна фрагментів тексту;
- друк і попередній перегляд документів;
- використання буквиць;
- подання тексту сторінки у вигляді кількох колонок;
- перевірка орфографії та граматики;
- використання шаблонів документів;
- створення графічних об'єктів засобами MS Word;
- вставка автофігур, фігурного тексту та математичних формул;
- вставка об'єктів, підготовлених іншими програмами;
- застосування самостійно створених стилів для автоматичного форматування тексту у відповідності з певними правилами;
- можливість виводити на панелі або прибирати з панелей інструментів окремі кнопки;
- можливість створювати власні панелі інструментів, розміщуючи на них кнопки найбільш вживаних інструментів;
- можливість створювати гіперпосилання для швидкого доступу до документів або окремих частин документа.

3.6. Загальна характеристика MS Excel 2003

MS Excel 2003 – це табличний процесор, що входить до складу пакету офісних програм Microsoft Office 2003.

Об'єктом опрацювання в MS Excel є *документ*, який зберігається в файлах типу *.xls*. За термінологією Excel такий файл називається *робочою книгою*. Робоча книга складається з окремих електронних таблиць або *робочих листів* (в MS Excel 2003 їх може бути від 1 до 255). Кожний робочий лист складається з 256 стовпчиків і 65536 рядків, причому рядки пронумеровані числами від 1 до 65536, а стовпчики позначені латинськими літерами A, B, C, ..., Z, AA, AB, ..., IV. На перетині стовпчиків і рядків розташована *комірка* (клітинка), яка і є одним з основних структурних елементів Excel. У будь-яку комірку можна ввести первинну інформацію (число або текст) або формулу для визначення похідної інформації. Для посилання на комірку використовується її *адреса*, яка складається з позначення стовпчика і номеру рядка, на перетині

яких стоїть комірка, наприклад, A23, C1 і т.д. Для посилення на *діапазони комірок* (області прямокутної форми) використовуються адреси комірок, розташованих на головній діагоналі такої області, наприклад, A2:E13.

Запускається на виконання MS Excel 2003 звичайним для Windows способом, тобто через головне меню, за допомогою ярличка або ж разом з вже існуючим документом. Після завантаження програми на екрані з'явиться робоче вікно програми з новою робочою книгою і декількома чистими листами.

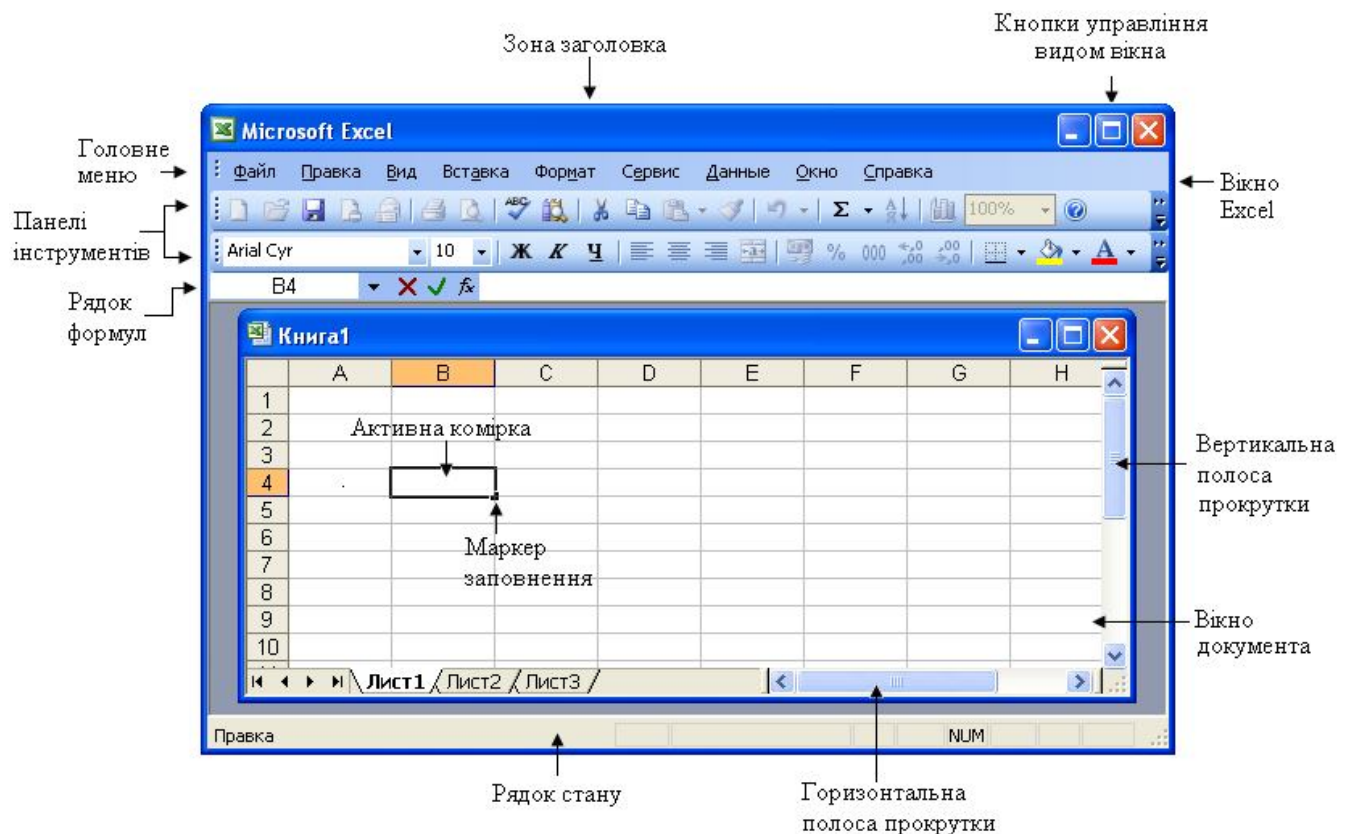


Рис. 3.6. Загальний вигляд вікна MS Excel 2003

За замовченням в робочій книзі 3 робочих листи, але є можливість додавати нові листи, видаляти непотрібні, переставляти їх місцями, перейменовувати тощо. Виконуються ці дії зазвичай за допомогою контекстного меню, яке з'являється після клацання мишкою на назві одного з листів.

Рядок формул використовують для введення і перегляду формул. У першій його частині (її ще називають полем імені) міститься адреса активної комірки і кнопка, що відкриває список іменованих комірок або список функцій (під час створення формул). У другій частині при введенні даних з'являються три допоміжні кнопки: *Отмена* (✕), *Ввод* (✓) і *Вставка функции* (fx). У третій частині завжди відображується вміст активної комірки; вона ж використовується для введення та редагування формул.

Створення нової робочої книги (документа), *відкриття* вже існуючої книги та *збереження* книги на зовнішніх пристроях, а також *закриття* документа і *вихід з Excel* виконується стандартними для Windows способами.

При цьому нову робочу книгу можна створювати на основі вже існуючого шаблону з готовими до використання формулами та іншими елементами.

3.7. Введення даних та редагування електронних таблиць

В комірках робочого листа можуть знаходитися формули або постійні дані різних типів.

Таблиця 3.2. Основні типи даних Excel

№ пп	Тип даних	Приклади	Пояснення
1	<i>Числовой</i>	1235 -234	Найбільш загальний спосіб представлення чисел
2	<i>Денежный</i>	1 234 грн.	Відображення грошових сум
3	<i>Финансовый</i>	12 356,00 грн.	Відображення грошових сум з вирівнюванням по розділювачу цілої та дробової частин
4	<i>Дата</i>	21.01.2012 12 січня 2012 р.	Відображення дат і часу, представлених числами, у вигляді дат
5	<i>Время</i>	13:30 21.01.12 13:30	Відображення дат і часу, представлених числами, у вигляді часу
6	<i>Процентный</i>	34,25 %	Значення в комірці помножується на 100 і виводиться з символом відсотків
7	<i>Экспоненциальный</i>	1,54E+06	Відображення чисел у формі з порядком
8	<i>Текстовый</i>	Ціна 1234 +/-	Значення відображаються точно так, як вводяться. Опрацьовуються як рядки, незалежно від змісту

Для введення даних в певну комірку її необхідно спочатку виділити (зробити активною), для чого досить клацнути у ній лівою кнопкою миші або перейти до неї, використовуючи клавіші управління курсором.

Інформацію можна вводити за допомогою клавіатури безпосередньо в комірку або ж в рядку формул. В першому випадку для фіксації інформації досить натиснути клавішу *Enter*, в другому - натиснути мишкою кнопку ✓, яка розташована в рядку формул.

Редагування даних в попередньо виділеній комірці виконується відповідно до табл. 3.3.

Таблиця 3.3. Типові операції редагування

№ пп	Операція	Спосіб виконання
1	Виділення окремої комірки	Клацнути ЛКМ на комірці
2	Виділення діапазону комірок	“Протягнути” мишкою від одного кута діапазону до протилежного при натиснутій ЛКМ
3	Виділення стовпчика	Клацнути ЛКМ на назві стовпчика
4	Виділення рядка	Клацнути ЛКМ на номері рядка
5	Виділення всього робочого листа	Клацнути ЛКМ на перетині заголовків стовпчиків та нумерації рядків
6	Виділення кількох комірок (діапазонів, стовпчиків, рядків), не розташованих поруч	Перший елемент як звичайно, кожний наступний – виділяти при натиснутій клавіші Ctrl
7	Зняття виділення з елемента	Клацнути ЛКМ поза виділеним елементом
8	Видалення вмісту комірок, діапазонів, стовпчиків, рядків	Виділити відповідний елемент і натиснути клавішу Del
9	Копіювання окремих комірок	“Перетягнути” попередньо виділений фрагмент в потрібне місце при натиснутій ЛКМ і натиснутій клавіші Ctrl або через буфер обміну (аналогічно Word)
10	Переміщення окремих комірок	“Перетягнути” мишкою попередньо виділений фрагмент в потрібне місце при натиснутій ЛКМ або через буфер обміну
11	Копіювання (переміщення) діапазонів комірок, стовпчиків, рядків	Виділити відповідний елемент, скопіювати або перемістити в буфер обміну (аналогічно Word); виділити в потрібному місці елемент точно такої структури і виконати операцію вставки з буферу обміну
12	Вставка нової комірки	Виділити комірку, перед якою відбувається вставка, і відпрацювати команди <i>Вставка</i> $\Rightarrow$ <i>Ячейки</i>
13	Вставка нового стовпчика	Виділити стовпчик, перед яким відбувається вставка, і відпрацювати команди <i>Вставка</i> $\Rightarrow$ <i>Столбцы</i>
14	Вставка нового рядка	Виділити рядок, перед яким відбувається вставка, і відпрацювати команди <i>Вставка</i> $\Rightarrow$ <i>Строки</i>
15	Об’єднання кількох комірок в одну	Виділити комірки, відпрацювати команди <i>Формат</i> $\Rightarrow$ <i>Ячейки...</i> , на вкладці <i>Выравнивание</i> встановити прапорець <i>Объединение ячеек</i>
16	Відміна об’єднання кількох комірок	Зняти прапорець <i>Объединение ячеек</i> (див. п.15)

3.8. Форматування електронних таблиць у MS Excel

Фрагмент електронної таблиці з розв’язком задачі можна відформатувати у відповідності з певними вимогами. Форматування можна виконувати автоматично або вручну.

Для автоматичного форматування необхідно виділити відповідний діапазон таблиці; виконати команди *Формат* $\Rightarrow$ *Автоформат*; обрати за зразком один зі стандартних способів оформлення та уточнити (при необхідності) деякі параметри. Після клацання мишкою на кнопці ОК фрагмент буде автоматично оформлений відповідно до обраного зразка.

Самостійне форматування таблиці користувачем потребує більше часу, але надає значно більше можливостей. Для виконання більшості операцій може бути використане діалогове вікно *Формат ячеек*, для появи якого необхідно клікнути по клітинці чи діапазону клітинок таблиці та виконати команду *Ячейки...* з підменю *Формат* або ж команду *Формат ячеек...* контекстного меню.

Вікно *Формат ячеек* має вкладки *Число*, *Выравнивание*, *Шрифт*, *Граница*, *Вид*, *Защита*.

Вкладка *Число* використовується для надання кліткам певних форматів (див. табл. 3.2) та уточнення їх параметрів (наприклад, для чисел - кількості знаків після коми).

Вкладка *Выравнивание* використовується для встановлення способу вирівнювання даних по горизонталі та вертикалі, орієнтації даних в клітинці, а також об'єднання кількох суміжних клітинок в одну. Крім того, можна встановити режими розташування тексту у клітках у декілька рядків (*переносить по словам*) та автоматичної зміни ширини стовпчиків, якщо дані не вміщуються в ньому.

Вкладка *Шрифт* дозволяє вибирати параметри шрифту для виділеного діапазону, включаючи гарнітуру, розмір шрифту, стиль накреслення символів (напівжирний, курсив, підкреслений), колір символів. Крім того, можна встановити вид підкреслення (одинарна або подвійна лінія, тільки дані або вся клітинка) і деякі ефекти (закреслений символ, верхній або нижній символ).

Вкладка *Граница* дає можливість оформлення границь клітинок. При цьому можна вибрати тип та колір лінії і вказати, до якої саме границі застосувати вибрані параметри.

Вкладка *Вид* використовується для кольорового оформлення клітинок. Вкладка *Защита* використовується для захисту змісту клітинок у діапазоні (спрацьовує тільки після захисту робочого листа в цілому).

Ширину стовпчиків і висоту рядків можна встановити за допомогою мишки так само, як і в таблицях Word. Точні значення цих параметрів можна задати за допомогою системи меню (*Формат* $\Rightarrow$ *Столбец* $\Rightarrow$ *Ширина...* або *Формат* $\Rightarrow$ *Строка* $\Rightarrow$ *Высота*). Крім того, підменю пунктів *Формат* $\Rightarrow$ *Столбец* і *Формат* $\Rightarrow$ *Строка* надають можливість приховати або відобразити стовпчик (рядок) і встановити режими автоматичного підбору ширини стовпчика або висоти рядка.

В багатьох випадках форматування попередньо виділених клітинок можна виконати за допомогою панелі інструментів *Форматирование* (див. рис. 3.7), зокрема - змінити шрифт (1), розмір шрифту (2), стиль накреслення символів (3); встановити спосіб горизонтального вирівнювання даних в клітинці (4), задати деякі формати даних (5), збільшити або зменшити розрядність чисел (6),

зменшити або збільшити відступи (7), задати вид границь (8), колір заливки (9) та колір шрифту (10).

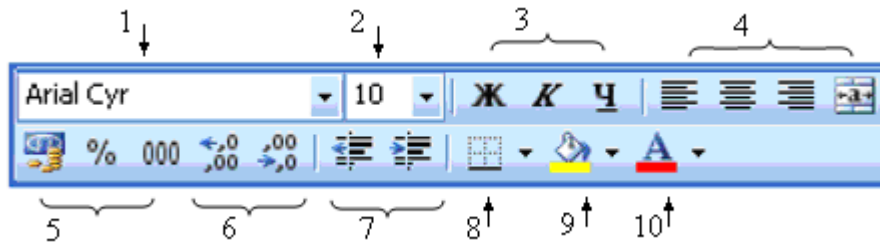


Рис. 3.7. Приклад панелі *Форматирование*

Всі параметри форматування виділених комірок можна зняти за допомогою команд *Правка* $\Rightarrow$ *Очистить* $\Rightarrow$ *Форматы*.

3.9. Організація обчислень

Обчислення в MS Excel реалізуються за допомогою формул.

Формула в MS Excel може містити посилання на адреси або імена комірок, константи, оператори визначення дій, дужки і вбудовані функції. При цьому посилання на адреси або імена комірок фактично є аналогами змінних у звичайних математичних формулах. Прикладами операторів можуть бути знаки арифметичних дій: «^» - піднесення до степеня, «*» - множення, «/» - ділення, «+» - додавання, «-» - віднімання. Крім того, існують логічні оператори і оператори порівняння. Дужки () допомагають змінити послідовність виконання обчислень, якщо це потрібно. *Пріоритет виконання обчислень* в формулах наступний: перш за все виконуються дії в дужках, всередині дужок спочатку обчислюються значення функцій, потім виконуються операції піднесення до степеня, далі – множення або ділення, і в останню чергу – додавання або віднімання. Операції одного пріоритету виконуються зліва направо, у звичайній послідовності.

Введення формул завжди починається зі знака =, який вказує системі на необхідність збереження наступного за ним виразу у вигляді формули.. Для включення у формулу адреси деякої комірки не обов'язково вводити її за допомогою клавіатури, достатньо просто клацнути мишкою на цій комірці.

При виділенні комірки з формулою Excel завжди відображає формулу у рядку формул, але в комірці замість формули зазвичай виводиться результат обчислень, оскільки саме він потрібний користувачеві. При необхідності перегляду всіх формул робочого листа Excel можна встановити режим відображення формул в комітках (*Сервис* $\Rightarrow$ *Параметры* $\Rightarrow$ *Вид*).

Після введення формули в одну з комірок її можна копіювати в інші за допомогою буферу обміну. Для цього при активній першій комірці з формулою слід відпрацювати команду *Копировать* одним зі способів, потім виділити іншу комірку і відпрацювати команду *Вставить*.

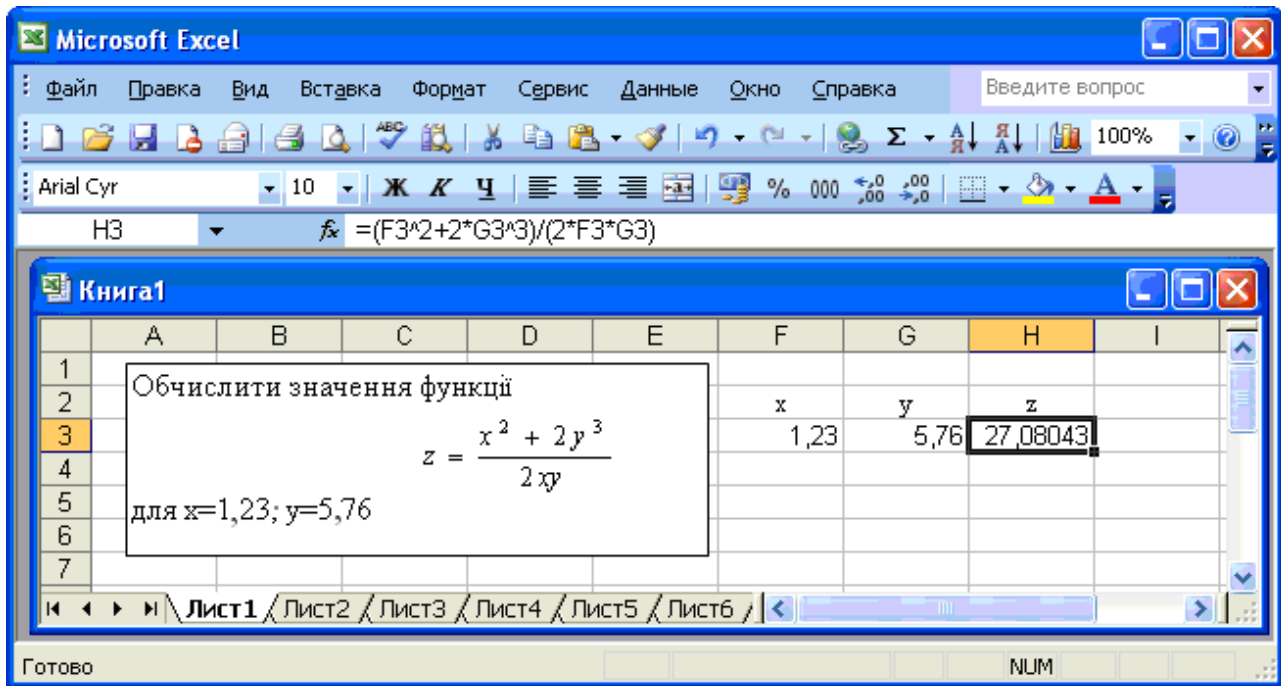


Рис. 3.8. Приклад організації обчислень

Розрізняють *абсолютні* і *відносні адреси комірок* в формулах. Відносна адреса комірки складається з назви стовпчика і номера рядка, наприклад, A1, D4. В абсолютних адресах перед номером рядка та (або) позначенням стовпчика ставиться знак \$, який вказує на те, що номер даного рядка або назва стовпчика не повинні змінюватися при копіюванні формули. В процесі копіювання формули у нову комірку посилання на відносні адреси комірок в формулі будуть автоматично змінені пропорційно зміщенню комірки-приймача формули відносно комірки-джерела формули (див. рис. 3.9).

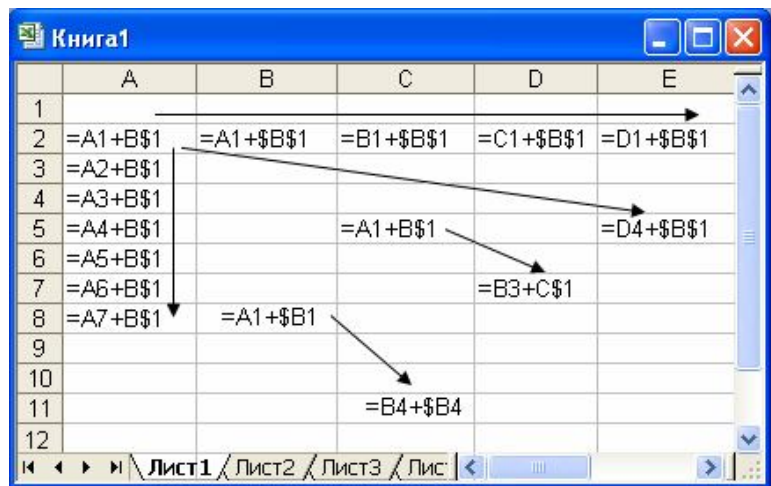


Рис. 3.9. Приклади копіювання формул

У випадку, коли формула копіюється в розташовані поруч комірки, можна скористатися іншою процедурою: досить схопити за маркер заповнення першої комірки і при натиснутій ЛКМ протягнути курсор по решті комірок. При копіюванні вправо (вліво) по рядку зсув на одну комірку призведе до збільшення (зменшення) номера кожного стовпця у формулі на одиницю, а при копіюванні униз (нагору) по стовпцю відповідним чином будуть змінюватися номери рядків. (саме так в прикладі на рис. 3.9 формула з комірки A2 скопійована в діапазоні B2:E2 і A3:A8).

Якщо формула має помилки, то після завершення введення формули у комірку може бути виведене повідомлення про це.

Таблиця 3.4. Повідомлення про помилки

Повідомлення	Пояснення
#ССЫЛКА	у формулі задано посилання на неіснуючі комірки
#ДЕЛ/0	у формулі є ділення на нуль
#ЧИСЛО!	у формулі використовуються недопустимі значення аргументів функції
#ИМЯ?	невірно записано ім'я функції
#ПУСТО!	невірно заданий діапазон комірок
#Н/Д	у формулі використовується посилання на комірку, в якій дані відсутні
#ЗНАЧ!	у формулі використовується аргумент недопустимого типу

Редагування формул (виправлення помилок) проводиться в рядку формул звичайними для текстових редакторів способами, в т.ч. шляхом копіювання або переміщення окремих фрагментів цієї ж або інших формул через буфер обміну.

Зауважимо, що в MS Excel існують і інші стилі адресації комірок.

Стиль тривимірної адресації передбачає додавання до адреси комірки імені листа робочої книги з позначкою !, наприклад, *Лист1!A1*. Тривимірна адресація застосовується при використанні в формулах даних з кількох листів робочої книги.

Стиль R1C1 передбачає нумерацію як рядків, так і стовпчиків. При цьому адреса комірки позначається буквою R, за якою наведений номер рядка, і буквою C, за якою наведений номер стовпчика. У вигляді R1C1 надається абсолютна адресація комірок; для відображення відносної адресації використовуються квадратні дужки. Наприклад: R1C1 - абсолютна адреса комірки на перетині 1 рядка і 1 стовпчика (A1); R[2]C[2] - відносна адреса комірки, яка розташована на 2 рядки нижче і на 2 стовпчики правіше від комірки, в якій адреса використовується.

Для вмикання або вимикання стилю адресації R1C1 досить виконати команди *Сервис⇒Параметры⇒Общие* і встановити або зняти прапорець *Стиль ссылок R1C1*.

3.10. Робота з функціями в MS Excel

Обчислювальні можливості Excel багато в чому визначаються великим набором стандартних функцій. При необхідності використання такої функції у формулі достатньо записати її стандартне ім'я і зазначити в дужках аргумент (якщо аргументів декілька, вони відокремлюються один від одного комою або крапкою з комою у залежності від установок Windows). Як аргументи можуть використовуватися числа, адреси і діапазони комірок, арифметичні вирази і т.д. Конкретний вид кожного аргументу, так само як і порядок запису аргументів,

визначається описом функції; довільно змінювати їх не можна. Обов'язковою приналежністю функції є дужки (вони записуються навіть тоді, коли функція аргументів не має).

Як приклад функції наведемо функцію підсумовування СУММ. Ця функція обчислює суму значень доданків, що перераховуються в дужках. Окремі доданки можуть бути числами, адресами або іменами комірок, діапазонами комірок. Приклади виразів із використанням функції СУММ:

=СУММ(A1,A2,A3,A4,A5)

=СУММ(A1:A5)

=СУММ(A1,B3:E8,23.4)

У перших двох прикладах обчислюється сума значень чисел, розташованих у перших п'ятих комірках стовпця А. У третьому прикладі до вмісту комірки А1 додається сума значень усіх чисел, розташованих у блоці комірок В3:Е8, а також постійне число 23.4.

Ще один приклад - логічна функція ЕСЛИ. Ця функція має три параметри: перший - логічний вираз; другий - значення, яке повинно бути результатом функції, якщо логічний вираз істинний; третій - значення, яке повинно бути результатом функції, якщо логічний вираз хибний. Таким чином, функція ЕСЛИ дозволяє реалізовувати алгоритми розгалуженої структури (див. приклад на рис. 3.10).

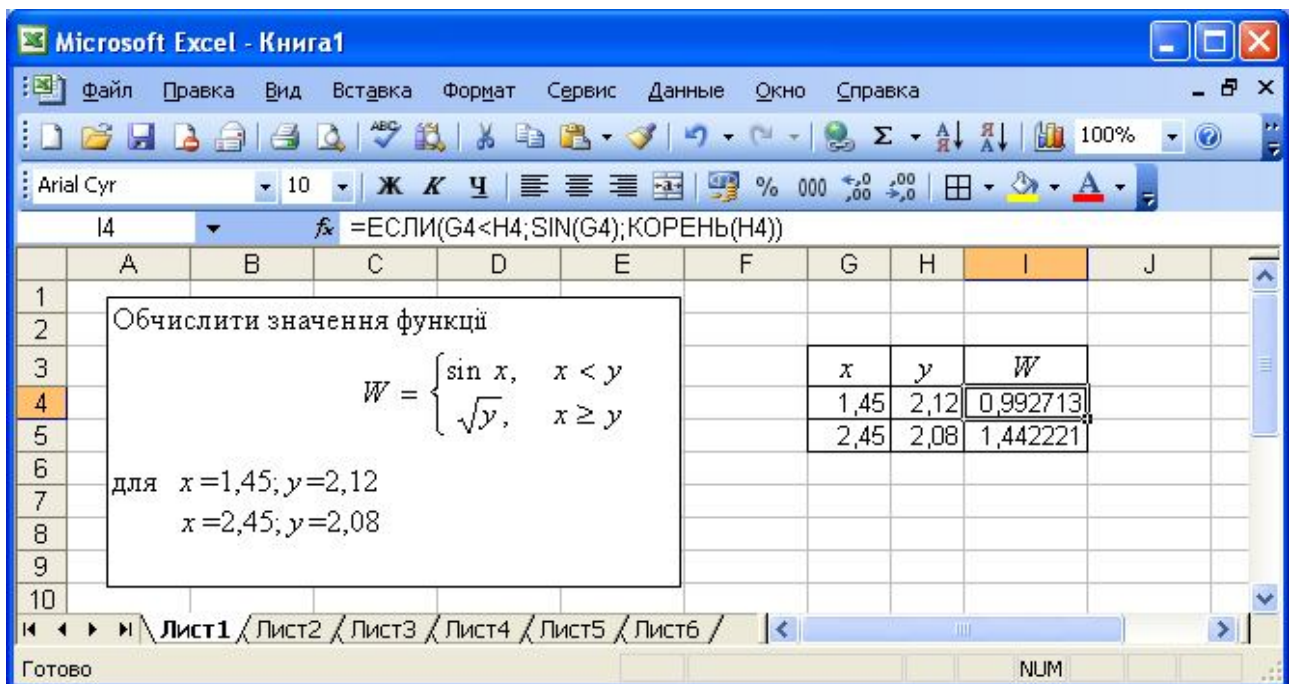


Рис. 3.10. Приклад використання функції ЕСЛИ

Усього в Excel визначено декілька сотень стандартних функцій. Вони розподілені по окремих категоріях: математичні, статистичні, фінансові, текстові, логічні, дати і часу, для роботи з базами даних тощо.

Таблиця 3.5. Найбільш поширені стандартні функції

Функція	Значення функції
СРЗНАЧ(число1;число2;...)	середнє арифметичне аргументів
КОРЕНЬ(число)	квадратний корінь числа
МАКС(число1;число2;...)	найбільший з аргументів
МИН(число1;число2;...)	найменший з аргументів
ABS(число)	абсолютне значення числа
LN(число)	натуральний логарифм числа
EXP(число)	експонента числа
LOG(число, основа)	логарифм числа за заданою основою
СЧЕТЕСЛИ(діапазон; умова)	кількість не порожніх комірок діапазону, які задовольняють умову
СУММЕСЛИ(діапазон1;умова; діапазон2)	сума значень тих комірок діапазону2, для яких відповідні комірки діапазону1 задовольняють умову (в разі відсутності діапазону2 - сума значень діапазона1)

Ознайомитися з іншими функціями можна за допомогою довідкової системи Excel. Крім того, введення функцій можна виконувати за допомогою спеціального *Майстра функцій*, що викликається клацанням миші на кнопці зі значком fx (на панелі інструментів *Стандартная* або в рядку формул). Майстер функцій працює в два кроки: на першому кроці можна вибрати потрібну функцію, на другому - ввести значення параметрів. На кожному кроці у вікнах є інформація (про призначення функції, кількість і смисл параметрів і т.д.), що дозволяє використовувати будь-яку функцію будь-якому користувачу.

Слід також відзначити, що в деяких випадках для виконання обчислень не обов'язково використовувати формули. В Excel діє функція *Автовычисление*: для виділеного діапазону комірок в рядку стану (в нижній частині вікна) можна відразу побачити середнє арифметичне, кількість значень, кількість чисел, максимум, мінімум або суму чисел (що саме - вибирається з контекстного меню, що викликається клацанням правої кнопки в області рядка стану).

*Питання теми 3, які винесені на самостійну проробку
і не включені в конспект*

1. Робота з таблицями в MS Word
2. Створення та редагування діаграм в MS Excel
3. Бази даних і системи управління базами даних
4. Технологія створення і демонстрації презентацій

Контрольні запитання до теми 3

1. Які способи завантаження текстового редактора Microsoft Word ви знаєте?
2. Назвіть основні елементи вікна програми Word та вікна документа.
3. Як виконується введення тексту та переміщення по тексті?
4. Як здійснюється виділення окремих елементів тексту?
5. Які дії можуть виконуватися з виділеним фрагментом?
6. Опишіть способи копіювання частин тексту з одного місця в інше, з одного документа в інший.
7. Опишіть способи перенесення частин тексту з одного місця в інше, з одного документа в інший.
8. Якими способами можна змінити шрифт, розмір символів, вид підкреслення?
9. Вкажіть параметри абзацу, які можна змінювати.
10. Якими способами можна виконати форматування абзаців?
11. Які можливості оформлення сторінок документа реалізовані в Word ?
12. Опишіть процедуру встановлення орієнтації сторінки та розмірів її полів.
13. Опишіть процедуру завантаження вже існуючого файлу для перегляду або редагування.
14. За допомогою яких дій виконується збереження файлів на дисках?
15. Як у текстовому редакторі застосовується *Буфер обмену*?
16. Які основні способи створення таблиць в Word ви знаєте?
17. Як виділити в таблиці: текст в комірці, комірку, кілька комірок, стовпчик, кілька стовпчиків, рядок, кілька рядків, всю таблицю?
18. Як вставити у таблицю один або кілька рядків (стовпчиків)?
19. Опишіть процедури переміщення, копіювання та видалення рядків, стовпчиків або окремих клітинок таблиці.
20. Як змінити ширину стовпчиків або висоту рядків таблиці?
21. Як об'єднати кілька комірок в одну?
22. Як розбити комірку на кілька частин?
23. Як розбити таблицю на дві?
24. Як об'єднати таблиці в одну?
25. Як видалити вміст таблиці і всю таблицю в цілому?
26. Які дії по форматуванню вмісту таблиці можна виконувати?
27. Опишіть процедуру автоформатування таблиць.
28. Опишіть призначення і основні можливості табличного процесора MS Excel.
29. Якими способами можна завантажити табличний процесор MS Excel?
30. Назвіть основні елементи вікна програми MS Excel та вікна робочої книги.
31. Вкажіть основні способи створення нової робочої книги в середовищі MS Excel.
32. Опишіть процедуру створення нового листа робочої книги.
33. Як здійснюється перейменування листів робочої книги ?
34. Як виконується виділення різних діапазонів та елементів електронної таблиці ?
35. Яким чином здійснюється введення даних у комірки таблиці?
36. Як встановити та відмінити об'єднання комірок?
37. Як відформатувати елементи таблиці (комірку, рядок, стовпчик, діапазон комірок)?
38. Які можливості форматування таблиці надаються на вкладнці *Выравнивание*?
39. Які можливості форматування таблиці надаються на вкладнці *Шрифт*?
40. Як використовується панель інструментів *Форматирование* при форматуванні елементів робочої книги?
41. Опишіть процедуру кольорового оформлення комірок.
42. Як виконується форматування таблиць при використанні команди *Автоформат*?
43. Перерахуйте основні типи даних, що використовуються при роботі в Excel.

44. Для чого використовуються формули? Які елементи може містити формула?
45. Опишіть процедуру введення формули в комірку.
46. В чому полягає різниця між відносними і абсолютними адресами комірок? Коли у формулах використовують абсолютну адресацію комірок?
47. Опишіть процедуру редагування формул.
48. Перерахуйте основні категорії функцій, наведіть приклади математичних функцій.
49. Опишіть процедуру роботи з *Мастером функцій*.
50. Опишіть процедуру створення послідовностей чисел, що змінюються з певним кроком.
51. Вкажіть призначення та опишіть синтаксис функцій ЕСЛИ, СУММ, СРЗНАЧ, СЧЕТЕСЛИ.
52. Опишіть можливі процедури копіювання формул.
53. Поясніть поняття "база даних", "система управління базами даних".
54. Які об'єкти входять до бази даних Microsoft Access ?
55. Опишіть призначення та принципи використання програми PowerPoint.
56. Поясніть поняття *презентація, слайд, макет слайда*.
57. Опишіть процедуру створення презентацій за допомогою *Мастера автосодержання*.
58. Опишіть процедуру самостійного створення презентацій.
59. Як можна оформити презентацію оригінальним дизайном?
60. Опишіть процедури перегляду та демонстрації презентацій.
61. Опишіть можливі процедури додавання до слайдів ефектів анімації.

Тема 4. ЧИСЕЛЬНІ МЕТОДИ ТА ЕЛЕМЕНТИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

4.1. Поняття про чисельні методи

Чисельні методи складають особливий клас методів розв'язання математичних задач. У традиційній математиці зазвичай шляхом перетворення математичних виразів одержують так зване аналітичне розв'язання у виді остаточної формули або сукупності формул. При використанні чисельних методів застосовується інший підхід: алгоритм розв'язання являє собою ланцюжок обчислень, проведених із конкретними числами; саме розв'язання завжди представляється тільки у вигляді числа або таблиці чисел.

Покажемо це на прикладі розв'язання задачі про обчислення площ фігур. Легко і просто вирішується ця задача, якщо фігура являє собою трикутник, паралелограм або коло. Якщо ж фігура більш складна — наприклад, обмежена віссю Ox , прямими $x = a$ і $x = b$ та графіком деякої функції $y = f(x)$ (див. рис. 4.1), то можна спробувати замінити реальну фігуру деякою іншою, площа якої легко обчислюється. Кілька варіантів такої заміни показані на рис.4.2.

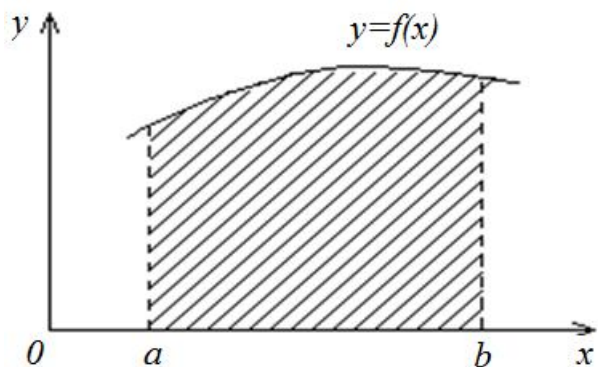


Рис. 4.1. Обчислення площ фігур

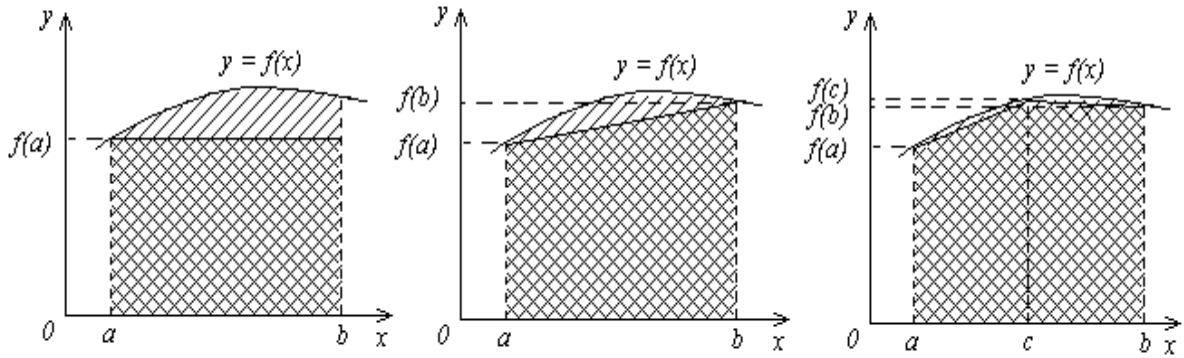


Рис. 4.2. Варіанти наближеної заміни реальної фігури

У першому варіанті реальна фігура замінена прямокутником. Висота цього прямокутника лежить на кривій $y = f(x)$ і в силу цього легко обчислюється: вона дорівнює $f(a)$. З огляду на те, що основа прямокутника дорівнює $(b - a)$, остаточно отримаємо

$$S = f(a) (b - a).$$

В другому варіанті реальна фігура замінена трапецією з основами $f(a)$, $f(b)$ і висотою $(b - a)$. Площа трапеції така

$$S = \frac{f(a) + f(b)}{2} (b - a).$$

У третьому варіанті реальна фігура замінена двома трапеціями. Їхня загальна площа обчислюється за аналогією з попереднім:

$$S = \frac{f(a) + f(c)}{2} (c - a) + \frac{f(c) + f(b)}{2} (b - c).$$

Як бачимо, у всіх трьох випадках обчислення не представляють труднощів, всі формули можна використовувати для обчислення наближеного значення площі реальної фігури. Проте, одного погляду на рисунок 4.2 достатньо, щоб зрозуміти, що найближче до точного значення одержується по третій з формул. Інтуїтивно зрозуміло також, що розбіжність між наближеним і точним значеннями площі буде ще менше, якщо розбити фігуру не на дві, а на три або більш трапецій. Цей факт доведений і теоретичним шляхом: при $n \rightarrow \infty$ розбіжність між наближеним і точним значеннями прямує до нуля. Звідси і практичний висновок: при збільшенні кількості розбивок n наближення стає усе більш точним.

Узагальнюючи все сказане, сформулюємо остаточно ідею наближеного обчислення площі фігури, показаної на рисунку 4.1.

Розіб'ємо відрізок $[a, b]$ на n рівних частин точками $x_i = a + i h$, де $h = (b - a)/n$, $i = 0, 1, \dots, n$. Після цього за допомогою ліній $x = x_i$ реальну фігуру розіб'ємо на n елементарних смужок і кожну таку смужку замінимо трапецією.

Тоді шукану площу можна приблизно обчислити як суму площ маленьких трапецій, що утворилися:

$$S = S_1 + S_2 + \dots + S_n,$$

де площу однієї трапеції, наприклад S_4 , можна обчислити як добуток напівсуми основ $f(x_3)$ і $f(x_4)$ на висоту $h = (b - a)/n$, тобто

$$S_4 = \frac{f(x_3) + f(x_4)}{2} h.$$

Після підстановки виразів для $S_1, S_2, \dots, S_n$ у формулу для S отримаємо

$$S \approx \left[\frac{f(x_0)}{2} + f(x_1) + f(x_2) + \dots + f(x_{n-1}) + \frac{f(x_n)}{2} \right] \cdot h$$

або

$$S \approx \left[\frac{f(a) - f(b)}{2} + \sum_{i=1}^n f(x_i) \right] \cdot h.$$

Ця формула називається *формулою трапецій*. Використовуючи її, можна обчислити площу фігури, обмеженої будь-якою функцією $f(x)$ при будь-яких a, b, n незалежно від складності самої фігури.

Саме це характерне для всіх чисельних методів: один метод можна застосовувати для розв'язання великої сукупності конкретних задач (з аналізованого класу задач), причому сам алгоритм вирішення задачі базується на виконанні звичайних арифметичних операцій. Однак, для забезпечення прийнятної точності кількість операцій може бути досить великою, що зумовлює необхідність застосування комп'ютерної техніки.

Практичне значення наведеної вище формули трапецій не обмежується обчисленням площ. Якщо ми пригадаємо геометричну інтерпретацію інтеграла, то зауважимо, що чисельне значення площі аналізованої нами фігури збігається зі значенням визначеного інтеграла

$$I = \int_a^b f(x) dx$$

Таким чином, формула трапецій фактично стає наближеною формулою обчислення визначених інтегралів.

Зазначимо, що чисельні методи придатні не тільки для обчислення інтегралів. До переліку задач, розв'язуваних за їх допомогою, входять апроксимація функцій, диференціювання, розв'язання нелінійних рівнянь і систем рівнянь, пошук екстремумів, розв'язання звичайних диференціальних рівнянь і рівнянь з частинними похідними, розв'язання інтегральних рівнянь і багато чого іншого. Проблеми такого типу виникають як при розв'язанні чисто наукових задач, так і в конкретних практичних застосуваннях. Без персональних комп'ютерів ці проблеми ще довго залишалися б невирішеними.

4.2. Сучасні інформаційні технології розв'язування задач обчислювальної математики

Моделювання і аналіз технологічних процесів або економічних явищ призводить до необхідності застосування досить серйозного математичного апарату. Враховуючи той факт, що більшість фахівців (хіміки, теплоенергетики, металурги, менеджери і інші) не спеціалісти в галузі математики, для них розроблені стандартні пакети прикладних програми, що дають можливість виконувати типові математичні операції, причому не тільки в числовому, але і в символьному виді. Такі пакети програм називають математичними пакетами.

Всі математичні пакети можна розділити на дві групи. До першої групи відносяться порівняно прості пакети *Derive*, *Eureka* та інші. До другої групи відносяться професіональні математичні пакети *MATLAB*, *Maple*, *MathCad*, *Mathematica*.

Серед математичних пакетів другої групи найбільш поширеними є пакети сімейства *MathCad*. Вони відрізняються стандартним інтерфейсом, легкістю освоєння і використання, великим набором функцій для інженерних розрахунків. Саме через функції реалізується обчислювальна потужність *MathCad*. Щоб не бути голослівними, наведемо перелік основних груп вбудованих функцій пакета *MathCad*: функції Беселя; функції комплексних чисел; функції розв'язання диференціальних рівнянь і систем (задача Коші, крайова задача, рівняння в часткових похідних); функції типу виразу; функції роботи з файлами; функції перетворень Фур'є; гіперболічні функції; функції опрацювання образів; функції інтерполяції й екстраполяції; логарифмічні і експоненціальні функції; функції теорії чисел і комбінаторики; функції щільності ймовірності; функції розподілу ймовірності; функції випадкових чисел; функції регресії і згладжування; функції опрацювання сигналів; функції розв'язання алгебраїчних рівнянь і систем, а також розв'язання оптимізаційних задач; функції сортування; спеціальні функції; статистичні функції; текстові функції; тригонометричні функції; функції округлення і роботи з частиною числа; функції роботи з векторами і матрицями; функції хвилястого перетворення. Як бачимо, наявні не тільки стандартні математичні функції і спеціальні функції, застосовувані в науково-технічних розрахунках, але і функції, що реалізують алгоритми розв'язання типових математичних задач.

Інші математичні пакети поки ще не одержали в нашій країні такого поширення, як *MathCad*, хоча кожний із них має свої переваги. Так, пакет *Mathematica* по праву вважається самим потужним математичним пакетом. У ньому реалізована найбільша кількість функцій із різних розділів вищої математики, у тому числі й вузькоспеціалізованих. Зокрема, даний пакет спроможний вирішувати диференціальні рівняння не тільки в чисельному, але й у символьном вигляді. Втім, явним лідером по можливостях саме символьних обчислень слід вважати пакет *Maple*. Він дозволяє виконувати найбільш складні аналітичні перетворення й одержувати аналітичні розв'язання найширшого класу математичних задач.

Обидва пакети (*Mathematica* і *Maple*) підтримують структуру документа у вигляді сукупності комірок, що робить документи більш упорядкованими і дозволяє створювати гіперпосилання між окремими комірками і цілими сторінками. У результаті редагування документів здійснюється простіше, ніж у *MathCad*. До того ж, *Mathematica* і *Maple* підтримують формат *.tex, що є стандартом де-факто для публікацій на математичні теми. Файли цього формату створюються й опрацьовуються в текстовому редакторі TeX, у якому дуже просто набирати математичні формули. Ще зовсім недавно статті в американські математичні журнали приймалися тільки у форматі TeX, і сьогодні ситуація змінилася не дуже, хоча на зміну TeX поступово приходить мови розмітки текстів SGML (Standart Generalized Markup Language- стандартна узагальнена мова опису документів) і його різновиди, наприклад HTML. У документ, створений за допомогою пакета *Mathematica*, можна включати не тільки будь-які об'єкти OLE і графічні файли *.wmf, *.bmp, *.eps, але і звукові файли формату *.wav (це дозволяє створювати справжні мультимедійні сторінки).

Система *MatLab* (MATrix LABoratory) споконвічно була призначена для чисельних обчислень і опрацювання матриць. З часом можливості *MatLab* істотно зросли, і сьогодні *MatLab* однаково добре підходить не тільки для чисельних і символьних обчислень, але і для моделювання, аналізу, візуалізації даних.

Фірмовою відмінністю *MatLab* є наявність *toolboxes* - спеціалізованих бібліотек, що реалізують унікальні для математичних пакетів функції. Наприклад, широко відома бібліотека *Simulink* дозволяє, не написавши ні строчки коду, побудувати функціональну схему системи управління зі стандартних блоків (підсилювач, суматор, інтегратор і т.д.) і проаналізувати її роботу. Крім того, є бібліотеки для фінансових розрахунків, медичних досліджень і т.д.

Підводячи підсумок огляду математичних пакетів, слід зазначити, що кожний із них має свою область переважного використання і свого основного користувача. *MathCad* - це інструмент для широкого кола рядових користувачів, включаючи студентів. *Mathematica* більше підійде для наукових працівників. *MatLab* можна вважати інструментом для фахової роботи конкретних спеціалістів (відповідно до наявними *toolboxes*). Пакету *Maple* можна віддати перевагу, якщо основна робота пов'язана із символьними перетвореннями, а також у випадках, коли доводиться працювати на застарілому ПК (цей пакет дуже компактний і найменш вибагливий до ресурсів комп'ютера).

4.3. Загальні принципи використання системи MathCad

Для запуску *MathCad 14* досить подвійного клацання мишки на ярличку цієї програми або ж на піктограмі вже існуючого документа. Універсальним способом запуску є використання системи меню: необхідно клацнути мишкою

на кнопці *Пуск*, обрати пункт меню *Программы*, а потім клацнути мишкою на підпункті *MathCad14*. Як наслідок вказаних дій на екрані з'явиться основне вікно системи з зоною заголовка, рядком горизонтального (головного) меню програми; панелями інструментів; вікном документу; рядком міні-статусу зі службовою інформацією.

Вся робота в середовищі MathCad проводиться за допомогою системи меню та панелей інструментів. Крім головного меню можна використовувати контекстно-залежне меню, склад команд якого залежить від того, в якій саме ситуації воно викликане.

Основними панелями інструментів є панелі *Стандартная*, *Форматирование* та *Математическая*. Кнопки панелі інструментів *Форматирование* добре знайомі всім користувачам Word. Це саме можна сказати і про цілий ряд кнопок панелі *Стандартная*. Інші кнопки відображають специфіку MathCad. Панель інструментів *Математическая* фактично є набором окремих математичних панелей. Перелік їх назв (*Калькулятор*, *График*, *Матрица*, *Вычисление*, *Исчисление*, *Логический*, *Программирование*, *Греческая*, *Символьная*) відразу нагадує користувачу, що ми маємо справу саме з математичним пакетом. Клацання мишкою на будь-якій з кнопок панелі *Математическая* призводить до появи в вікні документа окремої панелі інструментів з відповідною назвою.

Документ MathCad являє собою сукупність математичних виразів, текстових фрагментів і графічних областей (див. приклад на рис. 4.3).

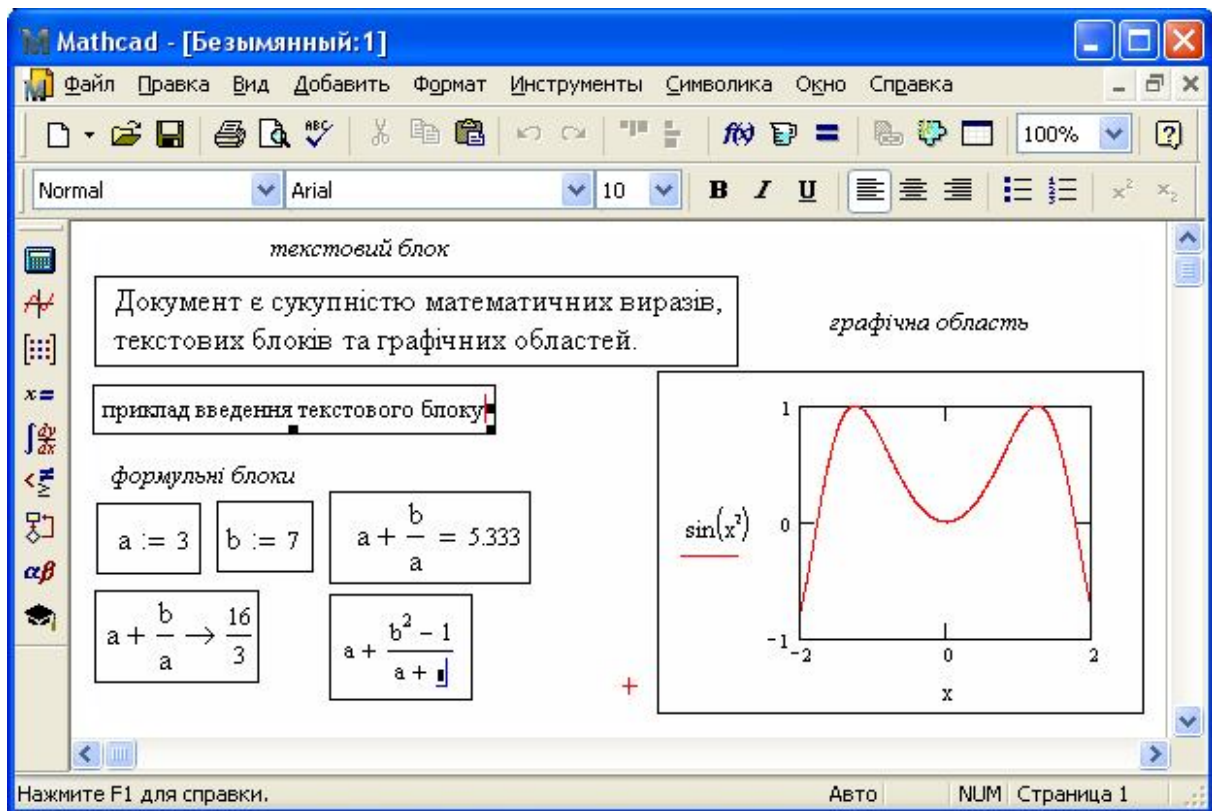


Рис. 4.3. Типові блоки документа MathCad

Після формування відповідного блоку його можна переміщувати за допомогою мишки або через буфер обміну в будь-яке місце документа. Така структура документа дає можливість організувати обчислення, прокоментувати та проілюструвати їх.

Текстові блоки вводяться, редагуються і форматуються звичайними для текстових редакторів способами. Створення формул і графіків виконується за допомогою формульного та графічного редактора MathCad.

Після створення документа MathCad його можна зберегти на диску у вигляді файлу типу *.xmcd*. Надалі будь-який збережений файл можна завантажити для перегляду, редагування або друку. Основні команди для роботи з файлами містяться в підменю пункту *Файл*.

4.4. Робота з формульним редактором

Формульний редактор призначений для введення математичних виразів і організації обчислень. Точка введення встановлюється клацанням мишки в потрібному місці документа (там з'являється червоний хрестик). Після цього можна приступати до набору виразу за допомогою клавіатури. При введенні першого ж символу на екрані формується область формульного блока, обмежена прямокутною рамкою, що розширюється в міру введення наступних символів.

Як і в будь-якій мові програмування, математичний вираз являє собою сукупність чисел, імен змінних і імен функцій, з'єднаних знаками арифметичних дій і круглими дужками. При запису чисел для відділення дробової частини від цілої використовується точка. У іменах змінних дозволяється використовувати великі і маленькі букви латинського і грецького алфавіту, а також цифри і деякі інші символи. На першому місці обов'язково повинна бути буква, усередині імені не допускаються пробіли. Великі і маленькі букви в іменах змінних розрізняються. Довжина імені практично не обмежена. Допускається використання імен з індексами (X_1 , $A_{\max}$ і т.д.). Перехід до набору нижніх індексів здійснюється натисканням клавіші з крапкою, повернення до звичайного режиму - натисканням клавіші пробілу.

Арифметичні операції задаються символами $+$, $-$, $*$, $/$, $^$. При обчисленні значень математичних виразів застосовується звичайний пріоритет операцій. Інший порядок дій задається за допомогою круглих дужок. У деяких випадках використання дужок не обов'язкове, треба лише уважно стежити за курсором введення. Останній являє собою синій кут, що вказує напрямок і місце введення. Звичайно він охоплює останній операнд, проте натисканням клавіші пробілу можна перемістити курсор введення так, щоб він охоплював деяку частину раніше введенного виразу. У залежності від конкретної ситуації буде різним і результат введення деякого символу операції (див. таблицю 4.1). Зауважимо, що в наведених прикладах перехід від ситуації $2 + 3$ до ситуації $2 + 3$ здійснений натисканням клавіші пробілу.

Таблиця 4.1. Приклади введення виразів

Вихідний стан	Введені символи	Відображення на екрані
$2 + 3$	$* 4$	$2 + 3 \cdot 4$
$2 + 3$	$* 4$	$(2 + 3) \cdot 4$
$2 + 3$	$/ 7$	$2 + \frac{3}{7}$
$2 + 3$	$/ 7$	$\frac{2 + 3}{7}$
$2 + 3$	$^ 2$	$2 + 3^2$
$2 + 3$	$^ 2$	$(2 + 3)^2$

З наведених прикладів видно також, що є певна специфіка введення виразів, пов'язана із символами $*$, $/$, $^$. У процесі введення формульний редактор автоматично замінює символ $*$ на більш звичний звичайному користувачу знак множення у вигляді крапки. При введенні знака $/$ на екрані формується зображення чисельника і знаменника як у звичайному дробу. Введення символу $^$ призводить до переміщення курсору в положення показника степеню (повернення до основного рівня рядка здійснюється натисканням клавіші пробілу).

При введенні математичних виразів можна використовувати палітру математичних знаків *Калькулятор* (вона з'явиться в вікні *MathCad* після клацання мишкою на відповідній кнопці панелі палітр). На ній крім цифр і звичайних знаків арифметичних дій є шаблони деяких поширених операцій (добування кореня, обчислення факторіалу і т.п.), а також шаблони математичних функцій, що часто зустрічаються (див. рис. 4.4). При виборі деякого шаблону з палітри (клацанням миші на ньому) у точці введення з'являється зображення відповідного знака і чорні квадрати на місці необхідних параметрів (операндів). Наприклад, при введенні знака квадратного кореня з'явиться зображення $\sqrt{\quad}$, при введенні шаблону функції $\sin x$ з'явиться $\sin(\quad)$ і т.д. Само собою зрозуміло, що на кожному місці введення (замість квадратиків) потрібно ввести відповідні параметри (числа або вирази).

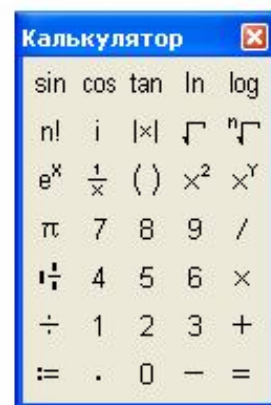


Рис. 4.4. Палітра Калькулятор

Всі змінні у математичних виразах, що обчислюються, попередньо повинні бути визначені, тобто їм повинні бути присвоєні певні числові значення. Для цього використовується операція присвоєння, що задається введенням символу $:$ (на екрані відображається $:=$). Перше присвоювання значення змінної можливо також за допомогою символу $=$, проте повторне присвоювання (переприсвоєння) за допомогою цього символу неможливе.

При спробі використання у виразі невизначеної змінної *Mathcad* видає повідомлення про помилку, а сама змінна зафарбовується в яскраво червоний колір.

4.5. Організація обчислень. Використання функцій в математичних виразах

У режимі *Автоматическое вычисление* (див. пункт головного меню *Инструменты*, підпункт *Вычислить*) значення математичного виразу визначається і виводиться на друк, якщо відразу після введення виразу натиснути клавішу з символом $=$. Наприклад, якщо за допомогою клавіатури ввести $2^2=$, то на екрані відобразиться $2^2 = 4$. Таким чином, у певному смислі оператор $=$ можна вважати оператором виведення. Якщо ж режим *Автоматическое вычисление* відключений, то для обчислення виразу після введення оператора $=$ потрібно ще натиснути клавішу **F9**.

Якщо обчислення тривають довго, то їх можна перервати, натиснувши клавішу **Esc**, а потім *OK* або *Cancel* для підтвердження або скасування переривання. Після переривання можна продовжити обчислення, натиснувши клавішу **F9**.

Точність обчислень задається значенням системної змінної **TOL**. Для зміни точності слід відпрацювати команди меню *Инструменты* і *Параметры рабочего листа*, а потім на вкладці *Встроенные переменные* встановити потрібне значення. Описане визначення **TOL** є глобальним, тобто діє для всього документа. Якщо ж потрібно змінити точність в конкретному місці документа, то використовується звичайне присвоєння типу $TOL:=0.0001$.

В *Mathcad* визначено багато стандартних функцій (математичних, статистичних та інших). Окремі приклади наведені в таблиці 4.2.

Таблиця 4.2. Деякі стандартні функції *Mathcad*

Функція	Опис	Функція	Опис
$\cos(z)$	косинус	$\sin(z)$	синус
$\ln(z)$	натуральний логарифм	$\log(z)$	десятковий логарифм
$\log(z, b)$	логарифм за основою b	$\exp(z)$	експонента
$\tan(z)$	тангенс	$\min(A)$	найменший елемент матриці A
$\text{trunc}(x)$	ціла частина дійсного числа	$\max(A)$	найбільший елемент матриці A

Принципи використання функцій у математичних виразах такі ж, як і в будь-якій мові програмування, тобто після імені функції в дужках обов'язково записуються фактичні параметри (аргументи), причому їхній тип, кількість і порядок запису повинні збігатися з передбаченими в описі функції.

Вводити функції в вирази можна звичним способом, тобто за допомогою клавіатури. Найбільш вживані математичні функції можуть бути введені також за допомогою панелі елементів *Калькулятор* (див. рис. 4.4). Крім того, можна

використати універсальний спосіб введення функцій, натиснувши кнопку $f(x)$ на панелі інструментів. З'явиться діалогове вікно, в якому можна буде вибрати певну категорію функцій і конкретну функцію з вибраної категорії (див. рис. 4.5). В цьому ж вікні дається короткий опис функції і можливість одержання детальнішої інформації про неї (кнопка ?).

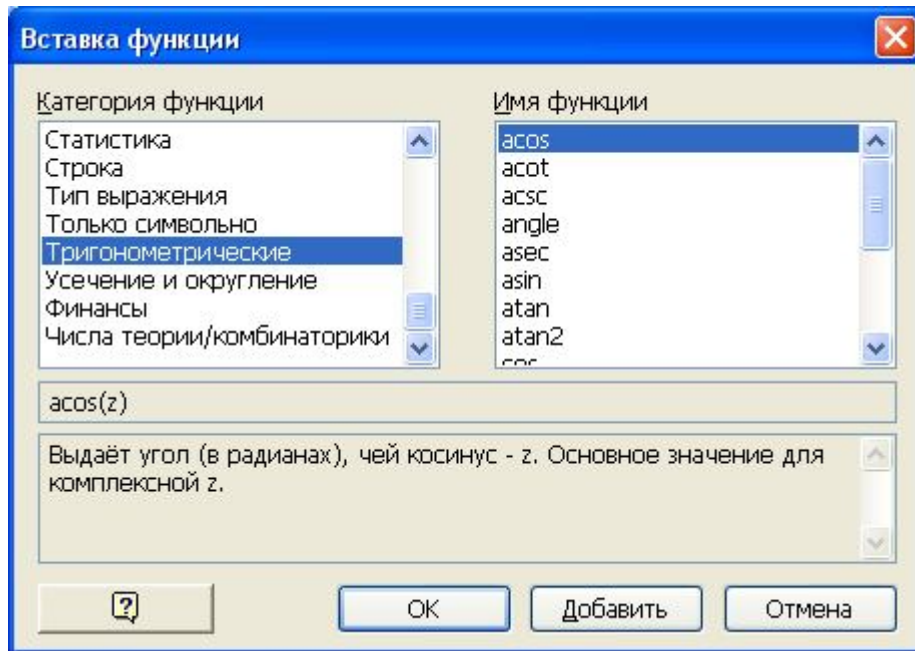


Рис. 4.5. Вікно введення функцій

Крім стандартних функцій користувач має можливість ввести (визначити) свою власну функцію, яка надалі (в межах даного документу) буде використовуватися нарівні зі стандартними функціями. Визначення функції користувача в найпростішому випадку має вид:

$$\langle \text{ім'я} \rangle (\langle \text{список аргументів} \rangle) := \langle \text{вираз} \rangle ,$$

причому при запису виразу можна використовувати як стандартні функції Mathcad, так і визначені раніше функції користувача. Наприклад, так:

$$f(x, y) := \sin(x) + e^y \cdot \cos(x + y)$$

4.6. Матриці і ранжовані змінні

Крім простих змінних у Mathcad використовуються масиви, тобто сукупності скінченного числа елементів, об'єднані за деякою ознакою під одним спільним іменем. Розрізняють одновимірні масиви (вектори) і двовимірні масиви (матриці). Доступ до конкретних елементів масиву здійснюється за допомогою цілочисельних індексів. Для елемента вектора вказується один індекс - порядковий номер елемента в сукупності. Для елемента матриці вказуються два індекси: перший - номер рядка, другий - номер стовпчика, на

перетині яких знаходиться даний елемент. Розмір матриці характеризується кількістю рядків і стовпчиків. Вектор може розглядатися як окремий випадок матриці, що складається з одного стовпчика або одного рядка.

Для введення матриць у Mathcad необхідно спочатку ввести шаблон, вказавши кількість рядків і стовпчиків, а потім заповнити його конкретними елементами. Введення шаблону здійснюється одним із таких способів:

- через меню (пункти *Добавить, Матрицу*);
- за допомогою математичної палітри *Матрицы* (кнопка );
- за допомогою комбінації клавіш **Ctrl+M**.

Приклади порожніх і частково заповнених шаблонів показані на рис. 4.6.



$$\begin{pmatrix} 1 \end{pmatrix} \quad \begin{pmatrix} 7 \\ 9 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 5 & 6 & 8 \\ 0 & 9 & 1 \\ 1 & 1 & 8 \end{pmatrix}$$

Рис. 4.6. Приклади шаблонів векторів і матриць

Відзначимо, що після визначення вектора або матриці в цілому доступними є також їхні окремі елементи (наприклад, A_5 , $M_{2,3}$). Перехід до запису індексів здійснюється за допомогою математичної палітри *Матрицы* (кнопка ), а до нормального рівня рядка – клавішею пробілу. Для запису операторів виведення або присвоювання переходити до нормального рівня рядка не обов'язково.

Ще один спосіб введення векторів і матриць заснований на використанні ранжованих змінних. Він застосовується у випадках, коли між елементами масиву й індексами елементів можна встановити явну аналітичну залежність.

Ранжована змінна - це спеціальний тип даних із множинними значеннями. Фактично це сукупність значень, що змінюються в заданому інтервалі з деяким кроком. Для задання ранжованої змінної слід ввести її ім'я, оператор присвоювання, а потім - перше, друге й останнє значення. Наприклад, так:

$$x := 1, 1.5..3 \quad y := -2, -1.3..0.8 \quad n := 1..5$$

У цьому прикладі введена змінна x із значеннями 1; 1.5; 2; 2.5; 3, змінна y із значеннями -2; -1.3; -0.6; 0.1; 0.8 і змінна n із значеннями 1, 2, 3, 4, 5. Як бачимо, Mathcad самостійно визначає крок зміни і формує всі члени послідовності, що задається зазначеним вище способом. Відзначимо, що оператор присвоювання, як завжди в Mathcad уводиться клавішею **:** (двокрапка), а символ продовження послідовності **..** вводиться клавішею **;** (крапка з комою) або ж за допомогою математичної палітри *Матрицы* (кнопка ). Крок зміни значень може бути не тільки додатним, але і від'ємним. У окремому випадку, коли крок зміни дорівнює 1 або -1, запис спрощується (у наведеному прикладі для змінної n друге значення не зазначене).

Для виведення значень ранжованої змінної використовується звичайний оператор виведення **=** (див. приклад на рисунку 1.6).

Може здатися, що ранжована змінна - це вектор. Проте, це не так. Головна відмінність полягає в тому, що доступ до одного конкретного значення ранжованої змінної неможливий. У той же час, через ранжовану змінну можна визначати вектори і матриці.

Нехай, наприклад, елементи вектора A обчислюються за формулою $A_i = i^2$ ($i=1,2,\dots, 5$). Тоді задання вектора виглядає так, як показано на рис. 4.7. Там же відображений результат виведення на екран вектора A . Як видно, у першому випадку вектор має 6 елементів, незважаючи на те, що ранжована змінна i має 5 значень. Це викликане тим, що нумерація індексів може починатися або з 0, або з 1 у залежності від значення системної змінної $ORIGIN$. За замовченням $ORIGIN=0$, тому в першому випадку на друк виведений і нульовий елемент масиву. В другому випадку змінна $ORIGIN$ задана явно (її значення дорівнює 1, індекси починаються з 1 і вектор містить 5 елементів).

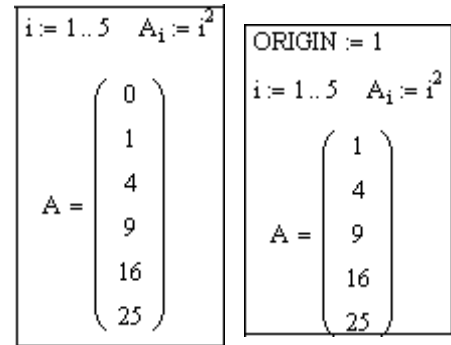


Рис. 4.7. Приклад вектора

4.7. Суть символьних обчислень

Символьними називають такі обчислення, результати яких представляються в аналітичному вигляді, тобто у вигляді формул. Такі обчислення носять узагальнений характер, в багатьох випадках дають більш повне уявлення про математичні чи фізичні закономірності задач, що розв'язуються. При наявності символьного рішення легко знаходяться числові значення шуканого параметра в конкретних випадках простою підстановкою відповідних значень вхідних величин в остаточну формулу.

В системах комп'ютерної математики символьні обчислення реалізуються за допомогою спеціальних процесорів. Основою такого процесора є ядро, в якому зберігається сукупність формул і алгоритмів формульних перетворень, за допомогою яких і проводяться символьні обчислення. Як наслідок, з'являється можливість виконання таких рутинних операцій, як спрощення виразів, розкладання на множники, інші алгебраїчні перетворення. В сучасних системах комп'ютерної математики, зокрема в MathCad, можна одержувати аналітичні розв'язки і більш складних задач, включаючи типові задачі математичного аналізу (границі функцій, похідні функції однієї незалежної змінної та частинні похідні функції багатьох незалежних змінних, невизначені і визначені інтеграли, невластні інтеграли, дослідження числових рядів на збіжність), розв'язування рівнянь та систем рівнянь тощо.

У середовищі *Mathcad* символьні обчислення можуть виконуватися або за допомогою системи меню (пункт *Символика*), або за допомогою оператора символьних перетворень $\rightarrow$. У першому випадку дії виконуються над явно заданою функцією, у другому — можна використовувати і визначені раніше

функції користувача. При використанні оператора символьних перетворень → деякі операції можуть бути задані директивами, а деякі — за допомогою шаблонів палітри *Исчисление*.

*Питання теми 4, які винесені на самотійну проробку
і не включені в конспект*

1. Побудова та опрацювання графіків в середовищі MathCad

Контрольні запитання до теми 4

1. В чому полягає різниця між аналітичними і чисельними методами розв'язання задач?
2. Опишіть процедуру обчислення визначеного інтегралу за допомогою формули трапецій.
3. Сформулюйте основні риси чисельних методів, найбільш привабливі для користувача.
4. Чому чисельні методи найбільшого поширення набули після появи персональних комп'ютерів?
5. Перерахуйте основні типи задач, які можна розв'язувати за допомогою чисельних методів.
6. В чому полягає особливість комп'ютерної математики порівняно зі звичайною математикою?
7. Перерахуйте основні математичні пакети прикладних програм. Чим вони відрізняються від інших програм обчислювального характеру (наприклад, Excel)?
8. Опишіть основні принципи роботи з пакетом MathCad .
9. Для чого в MathCad використовується палітра математичних знаків Калькулятор?
10. Опишіть правила введення арифметичних виразів та процедуру обчислення їх значень.
11. Які дії необхідні для обчислення значень виразів при ввімкнутому і вимкнутому режимі *Автоматическое вычисление*? Як перемикається вказаний режим ?
12. Для чого використовується системна змінна **TOL**? Як встановлюються її значення?
13. Опишіть процедуру введення векторів і матриць в Mathcad
14. Яке значення має системна змінна **ORIGIN** з точки зору використання векторів і матриць?
15. Що таке ранжована змінна? Чим вона відрізняється від вектора?
16. В яких випадках і як саме можна використати ранжовані змінні для введення векторів і матриць?
17. Які стандартні функції MathCad ви знаєте?
18. Опишіть можливі процедури введення стандартних функцій MathCad.
19. Як в MathCad описуються власні функції користувача? Як вони надалі використовуються?
20. Які види графіків можна будувати засобами Mathcad?
21. Опишіть процедуру побудови двовимірних графіків.
22. Які можливості редагування двовимірних графіків є в Mathcad і як вони реалізуються?
23. Для чого при побудові графіків використовуються ранжовані змінні?
24. Опишіть процедуру зміни масштабу відображення графіка. Для чого використовується ця можливість?

25. Для чого використовується трасування графіка? Як саме виконується ця процедура?
26. Опишіть процедуру побудови тривимірних графіків.
27. Які можливості форматування тривимірних графіків і як саме реалізуються в Mathcad?
28. Як можна змінити вигляд тривимірного графіка за допомогою мишки?
29. Що таке символні обчислення? Як вони виконуються MathCad?

Тема 5. ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

5.1. Поняття алгоритму, його властивості та способи зображення

Алгоритм - це скінченна послідовність точно визначених дій, що приводять до розв'язання поставленої задачі.

Основні властивості алгоритму: визначеність, масовість, результативність, дискретність.

Визначеність полягає в тому, що кожне правило алгоритму має бути чітким і однозначним; таким, що не припускає різного тлумачення.

Масовість - це властивість, згідно з якою алгоритм повинен бути придатним для розв'язання не однієї, а множини однотипних задач.

Результативність полягає в тому, що алгоритм обов'язково повинен приводити до кінцевого результату за скінчене число кроків.

Дискретність означає, що алгоритм являє собою набір елементарних дій, причому кожна з них виконується лише після завершення попередньої.

Якщо алгоритм складено добре, то його виконання може бути доручено будь-якому виконавцю, який вміє виконувати відповідні елементарні дії, незалежно від розуміння останнім задачі, що розв'язується. Таким чином, *комп'ютер також може виступати в ролі виконавця алгоритмів*. Для цього потрібно лише, щоб алгоритм був описаний за допомогою таких дій, які вміє виконувати комп'ютер.

Одна з властивостей алгоритму - масовість - передбачає його обов'язковий опис для того, щоб один раз розроблений алгоритм можна було використовувати і надалі. Існують різноманітні способи опису алгоритмів.

Словесний спосіб полягає в запису етапів виконання алгоритму на звичайній мові, до якої звикла людина. Основним недоліком словесного опису є відсутність наочності процесу опрацювання інформації.

Формульний спосіб опису алгоритмів полягає в тому, що етапи виконання задаються з використанням математичних символів та виразів з необхідними поясненнями. Цей спосіб більш компактний та наочний порівняно зі словесним описом, проте він не є строго формалізованим.

Графічний спосіб полягає в зображенні логічної структури алгоритму за допомогою графічних символів, які відповідають тій чи іншій обчислювальній операції.

5.2. Блок-схеми алгоритмів

Одним з найбільш поширених способів опису алгоритмів, призначених для ПК, є графічний спосіб. При цьому використовуються стандартні графічні символи для позначення певних дій. Сукупність таких символів (блоків) разом із лініями зв'язку утворює *блок-схему алгоритму*. Опис алгоритмів за допомогою блок-схем - компактний та наочний і до того ж строго формалізований. Існують навіть спеціальні стандарти, які регламентують використання символів блок-схем. Найуживаніші з символів зображені на рис. 5.1.

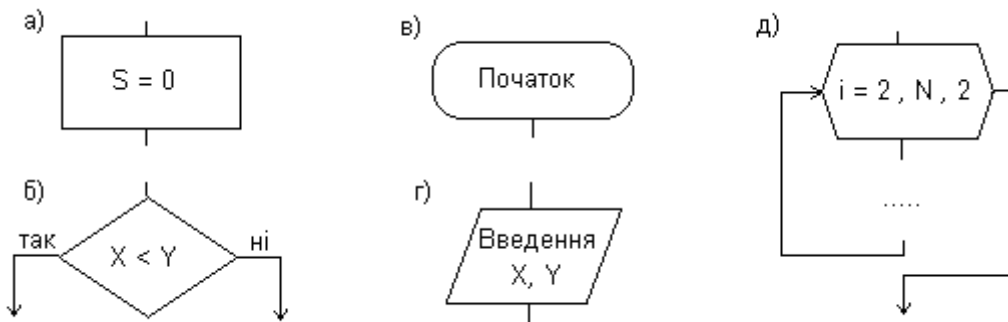


Рис. 5.1. Основні символи блок-схем

Етап опрацювання даних (наприклад, обчислення за формулами) позначається прямокутником (див. рис. 5.1,а), всередині якого записують зміст даного етапу

Перевірка умови зображується ромбом (рис. 5.1,б), всередині якого записують умову, що перевіряється. Від кутів ромба відходять дві стрілки, що вказують, які дії потрібно виконати, коли умова справджується, а які - коли ні.

Початок і кінець алгоритму зображується символом, який показано на рис. 5.1,в. Всередині цього символу відповідно вказується слово "Початок" або "Кінець".

Введення вихідних даних та виведення результатів роботи алгоритму зображується паралелограмом (рис. 5.1,г), всередині якого записують слово "введення" або "виведення" та вказують змінні, значення яких вводяться чи виводяться. Наведений символ не конкретизує спосіб введення або виведення, отже може бути використаним в будь-якій ситуації. В той же час для позначення операцій введення-виведення за допомогою конкретних зовнішніх пристроїв передбачено спеціальні символи.

Початок циклічного процесу, в якому одні і ті ж дії повторюються багато разів, зображується символом "модифікація" (рис. 5.1,д). Всередині символу записують закон, за яким змінюється значення параметрів циклу. Стрілочки, що входять та виходять з символу "модифікація", показують, які саме дії повторюються, та які дії виконуються після закінчення циклічного процесу.

За допомогою розглянутих символів (блоків) можна записувати різноманітні алгоритми. Порядок виконання блоків при цьому задається лініями зв'язку. Усі блоки мають одну точку входу і одну точку виходу (за винятком блока перевірки умови, в якому дві точки виходу). Блоки на схемі алгоритму можна нумерувати; стрілки на лініях зв'язку можна не зображувати, якщо лінія іде вниз або зліва направо. Як приклад графічного опису алгоритмів на рис. 5.2 наведена блок-схема алгоритму Евкліда для пошуку найбільшого спільного дільника двох натуральних чисел.

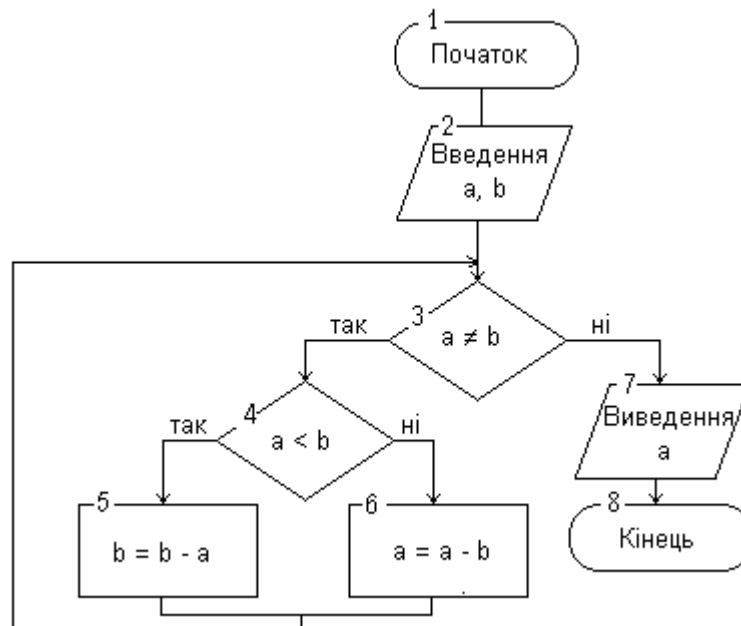


Рис. 5.2. Блок-схема алгоритму Евкліда

Зауважимо, що алгоритми, призначені для ПК, обов'язково мають містити дії по введенню початкових даних (ПК треба повідомити, з якими даними виконувати алгоритм) і виведенню результатів (користувач повинен дізнатися про результати виконання алгоритму комп'ютером).

5.3. Методи розробки алгоритмів

При розв'язуванні простих задач складання алгоритмів не викликає труднощів. В більш складних випадках використовуються спеціальні методи, за допомогою яких розробка алгоритмів проводиться швидше та якісніше. Одним з таких методів є метод покрокової деталізації.

Суть *методу покрокової деталізації* полягає в тому, що вихідна задача розбивається на декілька великих самостійних частин (підзадач) і спочатку складається укрупнена схема *основного алгоритму*, в якій відображаються найбільш важливі та суттєві зв'язки між виділеними частинами. На наступних етапах розкриваються (деталізуються) алгоритми розв'язання окремих підзадач. По відношенню до основного алгоритму такі алгоритми називаються допоміжними. Кожний *допоміжний алгоритм* повинен бути максималь-

но самостійним, щоб подальшу його деталізацію можна було виконати незалежно від деталізації інших допоміжних алгоритмів. При цьому, в свою чергу, можна використовувати допоміжні алгоритми для розв'язання більш простих задач і т.д. Така процедура значно полегшує процес розробки алгоритмів і дає можливість проводити цю роботу одночасно декільком спеціалістам (кожний з них працює над своєю частиною алгоритму).

Узагальненням методу покрокової деталізації є так званий *структурний підхід до розробки алгоритмів*. В його основу закладено такі принципи:

1. Алгоритм повинен складатися поетапно (по кроках).
2. Складна задача повинна розбиватися на дрібніші підзадачі, які легко алгоритмізуються; причому кожна така частина алгоритму повинна мати тільки один вхід та один вихід.
3. Логіка алгоритму повинна спиратися на мінімальне число досить простих типових структур.

Доведено, що алгоритм розв'язання задач будь-якого ступеню складності можна побудувати з використанням тільки трьох *типових (базових) структур* (див. рис.5.3).

Суть вказаних структур така:

- а) *лінійна структура* - "виконуй дію S_1 , потім дію S_2 ";
- б) *розгалуження* - "якщо умова P справджується, то виконуй дію S_1 , в противному разі - дію S_2 ";
- в) *цикл з передумовою* (цикл "Поки") - " доти, поки справджується умова P , виконуй дії S ".

Кожна з цих структур має один вхід і один вихід. При цьому блоки S , S_1 , S_2 самі можуть являти собою одну з розглянутих структур чи навіть їх комбінацію. Таким чином, весь алгоритм розв'язання задачі можна подати як лінійну послідовність типових структур.

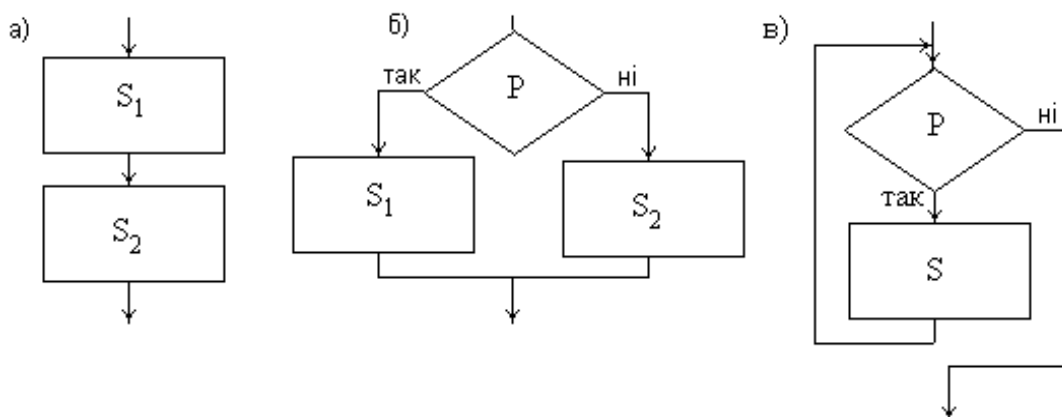


Рис. 5.3. Типові алгоритмічні структури

Крім розглянутих базових структур досить часто використовуються *додаткові типові структури* (див. рис. 5.4). Їх наявність в алгоритмах не обов'язкова, але з використанням цих структур можна в деяких випадках спростити складні алгоритми, зробити їх більш наочними.

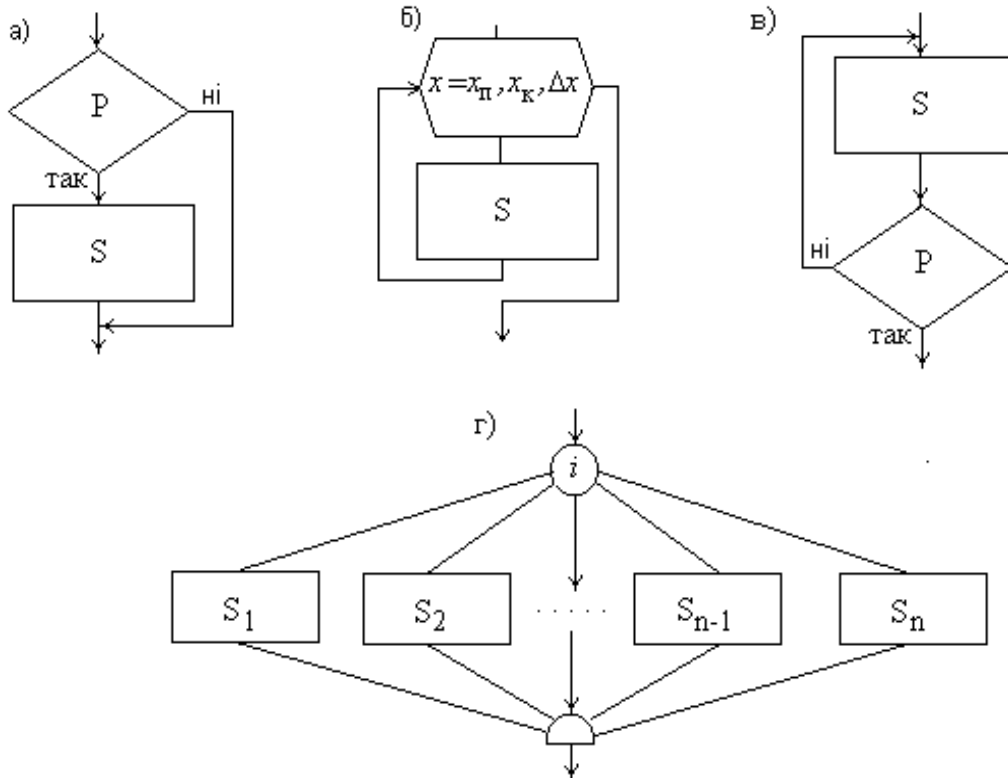


Рис. 5.4. Додаткові алгоритмічні структури

Суть додаткових структур, зображених на рис.5.4, така:

- а) *обхід* - "якщо умова P справджується, то виконуй дію S , в протилежному разі цю дію не виконуй";
- б) *цикл з параметром* - "для всіх значень x , що змінюється від $x_{\text{п}}$ до $x_{\text{к}}$ з деяким кроком Δx , виконуй дію S ";
- в) *цикл з післяумовою* (цикл "До") - "виконуй дії S до тих пір, поки не справдиться умова P ";
- г) *вибір* - "з сукупності дій $S_1, S_2, \dots, S_n$ виконуй тільки ту, порядковий номер якої співпадає зі значенням змінної i ".

В найпростіших випадках алгоритми можуть бути побудовані на основі однієї з розглянутих типових структур. Тоді їх називають алгоритмами лінійної, розгалуженої або циклічної структури, відповідно. Проте в більшості випадків алгоритм містить в собі кілька типових структур. Наприклад, в алгоритмі Евкліда (див. рис. 5.2) блоки 4,5,6 утворюють структуру "розгалуження". Спільно з блоком 3 та ж сама сукупність блоків утворює структуру "цикл з передумовою", а остання структура спільно з іншими блоками алгоритму утворює лінійну структуру.

5.4 Алгоритмізація типових обчислювальних процедур

В основі алгоритму обчислення суми скінченного числа доданків лежить ідея накопичення: спочатку деякій змінній S ; яка відповідає шуканій сумі, присвоюється значення нуля; потім організується цикл для перебору всіх

доданків і при кожному проходженні циклу значення змінної S збільшується на величину чергового доданка. Як приклад на рис. 5.5, а) наведено блок-схему алгоритму обчислення суми квадратів перших двадцяти членів натурального ряду.

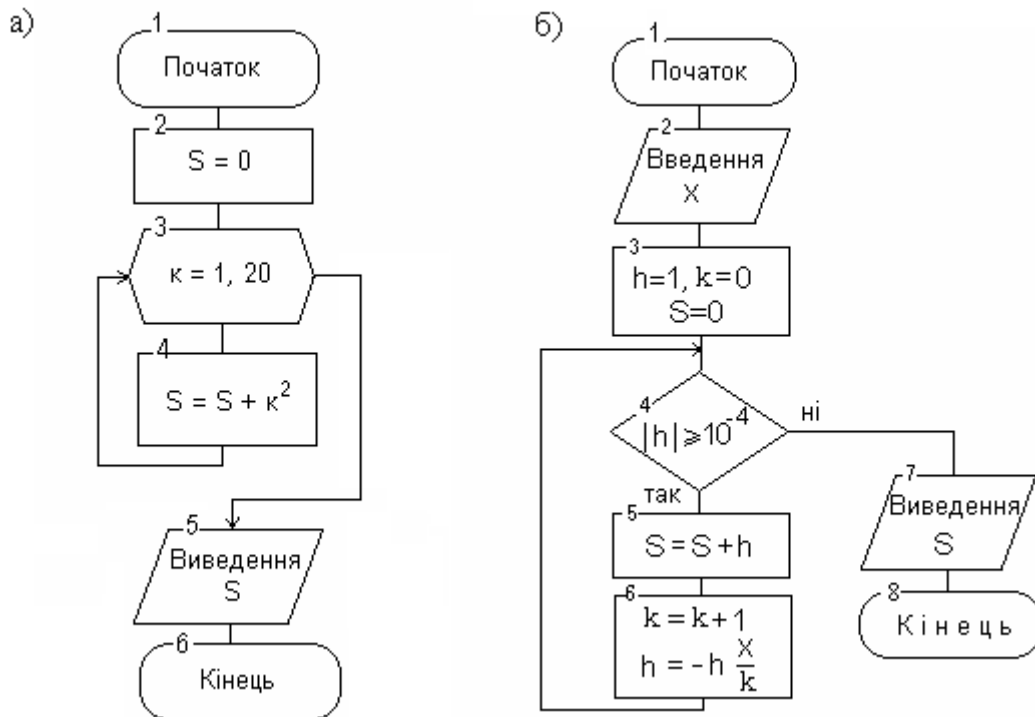


Рис. 5.5. Алгоритми обчислення сум

Схема проведення обчислень в цьому алгоритмі така:

$$\begin{aligned}
 & S = 0 ; \\
 k=1 & \quad S = S + k^2 = 0 + 1^2 = 1 ; \\
 k=2 & \quad S = S + k^2 = 1 + 2^2 = 5 ; \\
 k=3 & \quad S = S + k^2 = 5 + 3^2 = 14 ; \\
 k=4 & \quad S = S + k^2 = 14 + 4^2 = 30 \quad \text{і т.д.}
 \end{aligned}$$

Очевидно, після завершення циклу змінна S буде мати значення суми квадратів всіх перших двадцяти чисел натурального ряду.

Алгоритм обчислення суми скінченного числа доданків є типовим алгоритмом циклічної структури. В таких алгоритмах дії, що повторюються багаторазово, називаються *тілом циклу*, а змінна, що керує процесом повторювання, називається *параметром циклу*. Очевидно, що в цьому прикладі параметром циклу є змінна k , а тілом циклу - блок 4. Цикл завершується, коли параметр циклу набуде всіх своїх значень. Для відображення циклу в алгоритмі використано символ “модифікація” (блок 2 на рис.5.5, а).

Циклічні структури, в яких можна явно виділити параметр циклу та вказати закон його зміни, називаються *циклами з параметром*. Для будь-якого циклу з параметром можна заздалегідь підрахувати, скільки разів буде виконуватися тіло циклу. Іншим різновидом циклічних структур є *ітераційні цикли*. В таких циклах кількість повторень тіла циклу наперед невідома, і

завершення циклу відбувається не після перебору параметром циклу всіх своїх значень, а після того, як справдиться певна умова виходу з циклу.

В алгоритмі *обчислення суми нескінченного збіжного ряду* враховується той факт, що окремі члени ряду поступово зменшуються, прямуючи до нуля. Отже, накопичення суми проводиться до тих пір, поки члени ряду за абсолютною величиною більші від деякого досить малого числа ε , що називається точністю обчислення суми. Як тільки черговий член ряду стане меншим ε , то всіма наступними членами можна знехтувати і накопичення суми припинити. Таким чином, при визначенні суми нескінченного ряду наперед не відома кількість повторень тіла циклу, і алгоритм має структуру ітераційного циклу. Як приклад на рис. 5.5, б) наведено алгоритм обчислення з точністю $\varepsilon = 0,0001$ суми нескінченного ряду

$$S = 1 - x + \frac{x^2}{2!} - \dots + \frac{x^k}{k!} - \dots = \sum_{k=0}^{\infty} (-1)^k \frac{x^k}{k!}.$$

В алгоритмі врахований зв'язок між будь-яким членом ряду та попереднім членом

$$h_k = (-1)^k \frac{x^k}{k!} = (-1)^{k-1} \frac{x^{k-1}}{(k-1)!} (-1) \frac{x}{k} = -\frac{x}{k} h_{k-1}.$$

Обчислення добутку скінченного числа співмножників проводиться також на основі ідеї накопичення добутку в циклі у вигляді значення деякої змінної Р. Перед початком циклу змінній Р присвоюється значення одиниці, а при кожному проходженні циклу поточне значення Р помножується на черговий співмножник.

Табулювання функції $y = f(x)$ полягає в складанні таблиці значень даної функції на заданому проміжку $[x_{\text{п}}; x_{\text{к}}]$. Алгоритм розв'язання цієї задачі досить простий: необхідно організувати перебір значень x від $x_{\text{п}}$ до $x_{\text{к}}$ з деяким кроком Δx ; для кожного x обчислити відповідне значення y ; результати вивести на друк, щоб одержати таблицю. Як приклад на рис.5.6,а) наведено алгоритм табулювання функції $y = e^{2x} \cos x$ на проміжку $[0,2; 1,6]$ з кроком $\Delta x = 0,2$.

Якщо ж функція, яку треба протабулювати, залежить від двох або більше змінних, то необхідно використати *структуру вкладених циклів*, в якій всередині тіла одного циклу міститься ще один або кілька інших циклів. При організації таких циклів слід мати на увазі, що цикли (внутрішній та зовнішній) не повинні перехрещуватися, тобто першим повинен закритися той цикл, який відкрито останнім. Параметри циклів змінюються не одночасно - при кожному значенні параметра зовнішнього циклу параметр внутрішнього циклу приймає по черзі всі свої значення і тільки після цього змінюється значення параметра зовнішнього циклу.

Як приклад використання вкладених циклів на рис. 5.6,б) наведено алгоритм табулювання функції

$$Z = y e^{-x}; \quad 1 \leq x \leq 2; \quad \Delta x = 0,1; \quad 0,3 \leq y \leq 0,9; \quad \Delta y = 0,1.$$

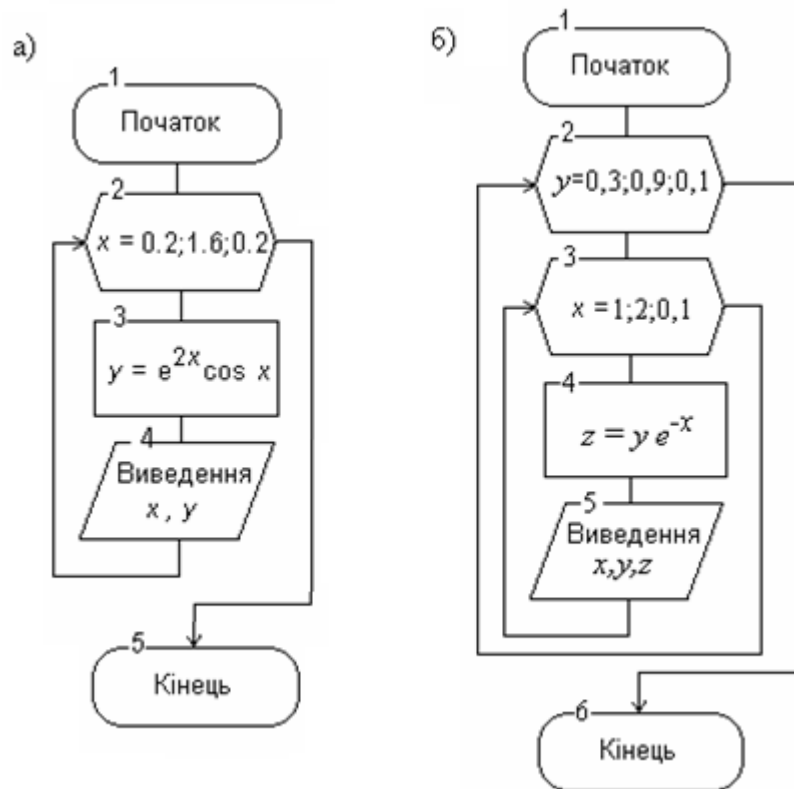


Рис. 5.6. Алгоритми табулювання функцій

5.5. Типові елементи алгоритмічних мов програмування

Мова програмування - це сукупність правил опису алгоритмів і даних для їх реалізації за допомогою комп'ютера.

Програма в алгоритмічних мовах програмування складається з послідовності *операторів*, що визначають певну дію або сукупність дій. Як правило, в кожній мові програмування є оператори присвоєння, введення-виведення даних, умовного та безумовного переходу, а також оператори циклу. Саме вони забезпечують можливість використання комп'ютера як виконавця алгоритмів, подібних до розглянутих нами раніше.

Для запису програм використовують *алфавіт мови*, тобто сукупність допустимих символів. До складу операторів входять службові (зарезервовані) слова та найпростіші конструкції мови (дані, стандартні функції, арифметичні та логічні вирази). Службові слова відображають смисл оператора, наприклад, INPUT (вхід), READ (читати) – це оператори введення; PRINT (друкувати), WRITE (писати) – це оператори виведення.

Дані зазвичай діляться на константи, змінні та масиви. *Константи* - це величини, значення яких у процесі обчислень завжди однозначно визначені і при виконанні програми не змінюються. *Змінні* - це величини, що утворюються в результаті виконання програми. На відміну від констант вони можуть змінюватися в ході обчислювального процесу, приймаючи значення тієї чи іншої константи. Змінні можуть бути простими та індексованими. *Індексована змінна* (або змінна з індексом) - це елемент масиву. В програмах

змінні представляються *ідентифікаторами* (іменами), які утворюються у відповідності з правилами мови програмування. Характерним атрибутом даних є тип, який визначає множину допустимих значень та обсяги пам'яті, необхідні для зберігання даних.

Стандартні функції в алгоритмічних мовах програмування спочатку передбачалися для обчислення основних елементарних функцій, наприклад, $\sin x$, $\sqrt{x}$. При необхідності використання в програмі таких функцій досить записати в відповідному виразі стандартне ім'я цієї функції та вказати в круглих дужках її аргумент. Аргументом може бути константа, змінна або ж будь-який інший вираз. Правила використання стандартних функцій виявилися настільки привабливими, що в подальшому набір таких функцій в багатьох мовах значно розширився за рахунок статистичних, інженерних та інших спеціальних функцій.

Арифметичні вирази - це аналоги алгебраїчних виразів в математиці. Вони задають правила для обчислення значення змінних величин і складаються з операндів (змінних, констант, назв функцій), об'єднаних знаками арифметичних операцій та круглими дужками.

Логічні вирази - це сукупність констант, змінних та функцій, об'єднаних знаками арифметичних та логічних операцій, а також операцій відношення ($<$, $=$, $>$ і інші). Логічний вираз може приймати лише одне з двох значень: істинно або хибно. Основне призначення логічних виразів – формування умов для операторів умовного переходу і операторів циклів.

Визначальними характеристиками для арифметичних і логічних виразів (можливо, різними в різних мовах програмування) є правила запису та пріоритет виконання операцій.

На закінчення огляду типових елементів мов програмування введемо ще поняття *структури програми*, як сукупності правил запису програми в цілому, включаючи порядок слідування окремих структурних одиниць та правила розташування операторів і інших елементів в рядках програми. Так, наприклад, для багатьох версій Бейсіка було характерне розташування кожного оператора в окремому рядку, який мав унікальний номер. В версіях Фортрана передбачався запис операторів, починаючи з сьомої позиції (перші п'ять призначалися для запису міток, а шоста – для запису символу продовження рядка).

5.6 Найпростіші конструкції мови Паскаль

Алфавіт мови програмування Паскаль складається з таких символів:

1. Цифри 0, 1, 2, ..., 9.
2. Великі та малі літери латинського алфавіту A, B, C, ..., Z, a, b, c, ..., z.
3. Знаки арифметичних операцій: $+$ (додати), $-$ (відняти), $*$ (помножити), $/$ (поділити).
4. Знаки відношень: $=$ (дорівнює), $>$ (більше), $<$ (менше).
5. Спеціальні символи: $.$ (крапка), $,$ (кома), $;$ (крапка з комою), $:$

(двокрапка), ' (апостроф), \$ (знак грошової одиниці), круглі, квадратні та фігурні дужки і ще деякі символи.

Крім того, для запису коментарів, пояснень в операторах виведення тощо, дозволяється користуватися великими та малими літерами української (або російської) мови і будь-якими іншими символами таблиці кодування.

Дані, що використовуються в алгоритмічній мові Паскаль, можна поділити на константи, змінні та масиви.

Ім'я змінної або масиву завжди починається з латинської літери, за якою слідує довільна комбінація літер та цифр. Довжина імені обмежується лише довжиною рядка програми, великі і малі літери сприймаються як еквіваленти; при написанні імен можна користуватися і тими, і іншими. Для наочності дозволяється використовувати символ підкреслювання. Приклади:

Z, KOP45, Limit, DigitWord, Serg_1.

В індексованих змінних індекси розміщуються в квадратних дужках, один від другого відокремлюються комами, наприклад: A[7], B[1,4], Z[I,J].

В мові Паскаль визначено чотири стандартних типи даних: *integer* (цілий), *real* (дійсний), *boolean* (бульовий або логічний), *char* (символьний або літерний). До типу *integer* належать усі цілі числа, тип даних *real* відповідає дійсним числам. Дані бульового типу можуть набувати одне з двох значень: *true* (істинно), або *false* (хибно). Значенням даних символьного типу може бути будь-який символ (тільки один!) алфавіту мови Паскаль.

Приклади запису числових констант

Число	145	-7	-3,58	$5,67 \cdot 10^{-8}$	1000	0,001
Запис мовою Паскаль	145	-7	-3.58	5.67E-8	1E3	1E-3

Символьна константа - це будь-який символ алфавіту, що міститься між апострофами: 'A', '+', 'Z' тощо. Констант бульового типу тільки дві, вони так і записуються: *true* та *false*.

Основними стандартними функціями, що набувають числових значень, в стандартній мові програмування Паскаль є такі функції:

$\text{abs}(x)$	-	$ x $	$\text{sqrt}(x)$	-	$\sqrt{x}$	$\text{sqr}(x)$	-	x^2
$\text{exp}(x)$	-	e^x	$\ln(x)$	-	$\ln x$	$\sin(x)$	-	$\sin x$
$\cos(x)$	-	$\cos x$	$\arctan(x)$	-	$\arctg x$	π	-	π

Аргументи тригонометричних функцій $\sin(x)$, $\cos(x)$ задаються обов'язково в радіанах. Результат функції $\arctan(x)$ обчислюється також в радіанах.

Основні правила запису арифметичних виразів в програмах на Паскалі:

- 1) Весь вираз записується в один рядок; багатоповерховий запис не допускається;
- 2) Всі знаки операцій обов'язково записуються явно (не $2a$, а $2*a$);
- 3) Будь-яке відхилення від установленого порядку виконання операцій

задається круглими дужками.

Для запису операції піднесення до степеню використовується формула $a^x = e^{x \ln a}$, добування кореня будь-якого степеня, крім кореня квадратного, замінюється піднесенням підкореневого виразу в дробовий степінь, тобто використовується запис $x^{1/n}$, а не $\sqrt[n]{x}$.

Приклади запису арифметичних виразів:

$\frac{-b + \sqrt{D}}{2a}$	$(-B + \text{sqr}(D)) / (2 * A)$
$\sin^2 x - \cos x^2$	$\text{sqr}(\sin(X)) - \cos(\text{sqr}(X))$
$x \cos x - \sqrt[3]{x}$	$X * \cos(X) - \exp(1/3 * \ln(X))$
$z_{i+1} - z_i z_{i-1}$	$Z[I+1] - Z[I] * Z[I-1]$
$\frac{\sqrt{x^2 \sin \sqrt{x} + 1} - 2}{2 x }$	$(\text{sqr}(\text{sqr}(X) * \sin(\text{sqr}(X)) + 1) - 2) / (2 * \text{abs}(X))$

5.7. Структура програми

Кожна програма, що написана мовою Паскаль, складається з таких частин: заголовок; розділи описів; розділ операторів.

Заголовок вказує ім'я програми і складається зі службового слова *program* та власне назви, наприклад:

program PRIMER;

В *розділах описів* визначаються (описуються) всі дані, що використовуються в програмі. При цьому розрізняють:

- 1) Розділ опису міток;
- 2) Розділ визначення констант;
- 3) Розділ визначення типів;
- 4) Розділ опису змінних;
- 5) Розділ опису процедур та функцій.

В конкретних програмах деякі розділи описів можуть бути відсутні, але розділ опису змінних зустрічається практично в кожній програмі. Це пов'язано з тим, що всі змінні в програмі на алгоритмічній мові Паскаль повинні обов'язково бути описаними.

Розділ опису змінних починається службовим словом *var*, за яким розташовується список, записи в якому мають таку структуру: спочатку через кому перелічуються один або кілька ідентифікаторів змінних, потім ставиться двокрапка, і після двокрапки вказується тип перелічених змінних. Різні записи відокремлюються один від одного крапкою з комою, наприклад:

```

var
  I, J, K: integer;
  X, Z: real;
  RESULT: boolean;

```

Імена в програмі на мові Паскаль можуть присвоюватися не лише змінним, але й константам. В подальшому замість самої константи (тобто її значення) в програмі можна вказувати лише її ім'я. Такі *іменовані константи* задаються в розділі визначення констант, наприклад, так:

```

const
  A = 5.8; K = 7; Sigma = 5.67E-8;

```

Тип іменованих констант визначається типом їх значень. В нашому прикладі A і Sigma - дійсні константи, а K - константа цілого типу.

За розділами описів розміщується *розділ операторів*. Він вказує дії, які повинна виконувати програма. Розпочинається розділ операторів службовим словом *begin* (початок). За ним слідує ряд операторів, відокремлених один від одного крапкою з комою. В одному рядку може бути кілька операторів; в той же час один оператор може займати й кілька рядків (між окремими елементами оператора може бути будь-яка кількість пропусків). В будь-якому місці програми (як в окремих рядках, так і в рядках з операторами) можна розміщати в фігурних дужках коментарі, тобто послідовність будь-яких символів мови, наприклад, текст пояснення українською мовою. В кінці розділу операторів ставиться службове слово *end* (кінець) і крапка. Перед *end* крапку з комою можна не ставити.

5.8. Оператор присвоєння

Загальний вигляд *оператора присвоєння*:

$$X := A ,$$

де X - ім'я змінної (простої або індексованої),

A - арифметичний вираз.

Виконується оператор присвоєння так: обчислюється значення арифметичного виразу і обчислене значення присвоюється змінній X. Наприклад, після виконання послідовності операторів

$$F:=5; V:=8; X:=F*V/(2*V-F-1); V:=V+X$$

змінна X набуде значення 4.0; змінна F збереже своє значення, що дорівнює 5; змінна V набуде нового значення, яке буде дорівнювати 12.0.

Тип змінної і тип арифметичного виразу в операторі присвоєння обов'язково повинні співпадати. Винятком є випадок, коли змінна має тип *real*, тоді вираз може бути як дійсного, так і цілого типу. При виконанні такого оператора обчислене ціле значення виразу буде автоматично перетворене на дійсне значення.

5.9. Оператори введення-виведення

Введення даних в алгоритмічній мові Паскаль може здійснюватись або за допомогою оператора присвоєння та опису відповідних констант, або за допомогою спеціальних операторів введення.

Оператори введення мають вигляд:

read (X,Y,Z,...); *readln* (X,Y,Z,...)

де X,Y,Z,... - список змінних, значення яких необхідно ввести.

При виконанні одного з таких операторів комп'ютер зупиняється і чекає, коли користувач ПК введе за допомогою клавіатури відповідні дані. При цьому одне значення від іншого повинно бути відокремлене одним або кількома пропусками або ж дані повинні розміщуватися в різних рядках. Оператори *read* та *readln* відрізняються лише тим, що після введення всіх даних, вказаних в операторі *readln*, відбувається перехід до нового рядка. Таким чином, наступний оператор введення буде звертатися до нового рядка з даними, навіть якщо в попередньому рядку використані ще не всі дані.

Дані типу *boolean* можна вводити в програму тільки за допомогою оператора присвоєння або шляхом опису іменованої константи. Оператори введення в цьому випадку не придатні.

Виведення даних на екран дисплея здійснюється за допомогою операторів виведення, що мають такий вигляд:

write (X,Y,Z,...); *writeln* (X,Y,Z,...);

де X,Y,Z,... - список змінних, значення яких необхідно вивести на екран. Елементами списку виведення можуть бути також послідовності будь-яких символів, обмежені з двох сторін апострофами, наприклад:

write ('x=', X, 'y=', Y, 'z=', Z)

В цьому випадку на екрані дисплея може з'явитись такий напис

x=12.3 y= -14.5 z=7.5

(звичайно, якщо змінні X, Y, Z набудуть саме таких значень). Різниця між операторами *write* і *writeln* така сама, як і між операторами *read* і *readln*: після виконання оператора *writeln* наступні дані будуть виводитися в новий рядок.

Елементом списку виведення може бути і арифметичний вираз. Крім того, в операторах виведення можна задавати кількість позицій, що відводяться для запису числа та його дробової частини. Так, наприклад, після виконання оператора

write ('A=', A:8:2, 'I+J=', I+J:3)

на екрані при A=123.74, I=10, J=3 з'явиться напис

A = 123.74 I+J= 13

Якщо довжина відведеного поля більша кількості цифр числа, що виводиться, то зайві позиції зліва заповнюються пропусками; якщо довжина поля менша,

ніж потрібно, то вона автоматично збільшується до необхідної. Якщо при виведенні дійсного числа кількість позицій для дробової частини не задана, то таке число буде виведене в експоненційній формі.

Зауважимо, що оператор *writeln* без списку виведення просто переводить курсор на початок нового рядка. Звернемо увагу також на те, що при виконанні операторів *read* або *readln* не з'являється ніяких ознак необхідності введення даних. Тому доцільно для організації введення даних використовувати послідовність операторів виведення і введення, наприклад, так:

write ('Введи значення X'); *readln* (X)

В такому випадку на екрані з'явиться текст, що нагадає про необхідність введення значення змінної X.

5.10. Програмування алгоритмів лінійної структури

Лінійні алгоритми - це найпростіші алгоритми, всі дії яких виконуються послідовно одна за одною. Для програмування таких алгоритмів використовуються оператори присвоєння та введення-виведення даних. Як приклад, наведемо алгоритм та програму обчислення площі трикутника, для якого відомі довжини сторін.

```

program Pr_1;
var
  A,B,C: real;      { сторони трикутника }
  P : real;         { на півпериметр }
  S : real;         { площа трикутника }
begin
  writeln ('Введіть довжини сторін a, b і c');
  readln (A,B,C);
  P:=(A+B+C)/2;
  S:= sqrt(P*(P-A)*(P-B)*(P-C));
  writeln ('Площа трикутника S=',S:8:3)
end.

```

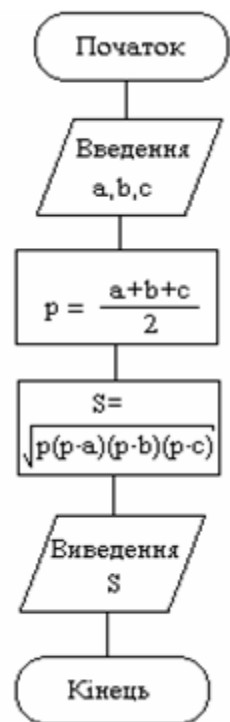


Рис. 5.7

5.11. Умовний оператор

Умовний оператор в мові Паскаль відповідає типовій алгоритмічній структурі "розгалуження" і має вигляд:

if B *then* S₁ *else* S₂;

де *if* (якщо), *then* (тоді), *else* (інакше) - службові слова; B - умова; S₁; S₂ - оператори. Виконується оператор дуже просто: якщо умова B справджується, то виконується оператор S₁, якщо ні - то оператор S₂.

Допускається скорочена форма умовного оператора:

if B then S.

В цьому випадку оператор *S* виконується тільки тоді, коли умова *B* справджується; інакше цей оператор не виконується, тобто скорочена форма умовного оператора відповідає типовій структурі "обхід".

За службовими словами *then* та *else* в умовному операторі може міститись лише один оператор. Якщо ж виникає необхідність виконання певної групи операторів, то слід скористатися так званим *складеним оператором*, який має таку структуру:

begin S₁; S₂;...; S_n end;

де *S₁; S₂;...; S_n* - будь-які оператори. Опрацьовується такий складений оператор точно так же, як і простий оператор.

Умовою може бути, наприклад, відношення, тобто два арифметичних вирази, розділені знаком відношення. Крім знаків *<*, *>*, *=* дозволяється використовувати комбінації знаків: *<=* (менше або дорівнює), *>=* (більше або дорівнює), *<>* (не дорівнює). Так, наприклад, оператор

if X<0 then Y:=sqr(X) else Y:=0.

задає обчислення виразу

$$y = \begin{cases} x^2, & x < 0 \\ 0, & x \geq 0 \end{cases}$$

В більш складних випадках умовою може бути будь-який *логічний вираз*, тобто сукупність констант, змінних та функцій, об'єднаних знаками арифметичних та логічних операцій, а також операцій відношення. Логічний вираз може приймати лише одне з двох значень: *true* (істинно) або *false* (хибно). Таким чином, умова *B* в логічному операторі справджується, якщо значенням відповідного логічного виразу буде *true*, і не справджується, якщо його значенням буде *false*.

З *логічних операцій* найчастіше (в усіх версіях мови) вживаються такі операції:

not - заперечення (логічне доповнення);
and - і (логічне множення);
or - або (логічне додавання);

Якщо *A* і *B* - логічні вирази, то *A and B* - це новий логічний вираз, який має значення *true* тільки тоді, коли обидва вирази *A* і *B* мають значення *true*. Якщо ж один або обидва вирази мають значення *false*, то і вираз *A and B* має таке ж значення.

Вираз *A or B* має значення *true* тільки тоді, коли один або обидва з виразів *A* і *B* мають таке значення. Якщо ж обидва вирази *A* і *B* хибні, то і вираз *A or B* має значення *false*.

Логічний вираз *not A* має значення *true* тоді, коли *A* - хибно; і значення *false* тоді, коли *A* - істинно.

Як приклад використання логічних виразів наведемо три варіанти запису умовного оператора, який перевіряє, чи належить задане число X інтервалу $[0,1]$:

- 1) *if* ($X \geq 0$) *and* ($X \leq 1$) *then write* ('так') *else write* ('ні');
- 2) *if* ($X < 0$) *or* ($X > 1$) *then write* ('ні') *else write* ('так');
- 3) *if not* (($X \geq 0$) *and* ($X \leq 1$)) *then write* ('ні') *else write* ('так');
- 4) $Z := X \geq 0$; $V := X \leq 1$; $W := Z \text{ and } V$;
if W *then write* ('так') *else write* ('ні');

Очевидно, що в першому випадку умова справджується тоді і тільки тоді, коли виконується і відношення $X \geq 0$, і відношення $X \leq 1$. В другому випадку умова вважається справедливою, якщо виконується хоча б одне з відношень $X < 0$ або $X > 1$. В третьому випадку умова справджується, якщо невірним є твердження про те, що і $X \geq 0$, і $X \leq 1$. Четвертий випадок аналогічний третьому, але з використанням булевих змінних. Змінні Z, V, W в останньому випадку обов'язково повинні бути описаними в розділі опису змінних як змінні булевого типу (*var* Z, V, W : *boolean*;).

5.12. Приклад програми розгалуженої структури

Наведемо алгоритм та програму визначення дійсних коренів квадратного рівняння $ax^2 + bx + c = 0$.

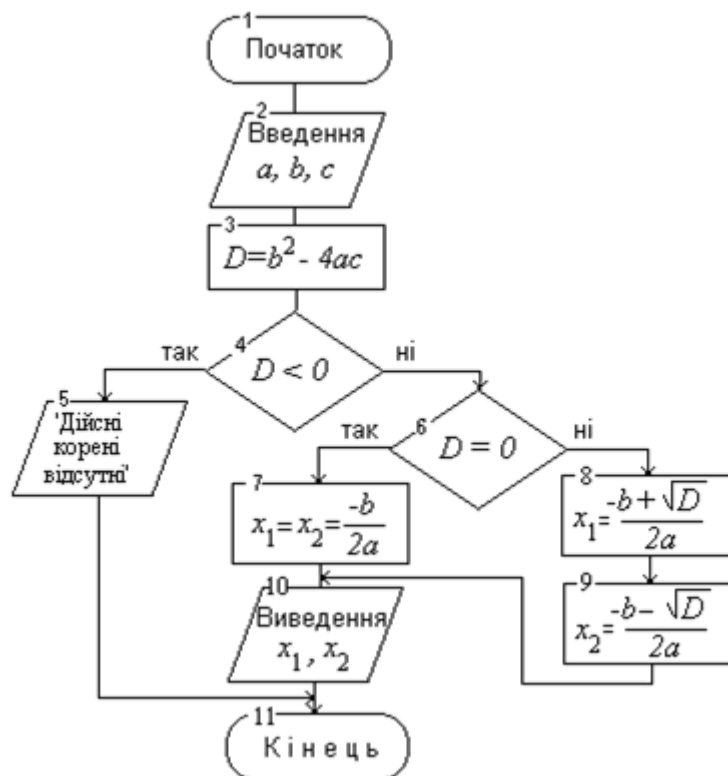


Рис.5.8. Алгоритм визначення дійсних коренів квадратного рівняння

Програма на алгоритмічній мові Паскаль має такий вигляд:

```

program Pr_2;
var
  A,B,C : real;    { коефіцієнти рівняння }
  D : real;        { дискримінант      }
  X1,X2 : real;    { корені рівняння   }
begin
  writeln ('Введіть коефіцієнти рівняння A,B,C'); readln(A,B,C);
  D:= sqrt(B)- 4*A*C;
  if D<0 then
    writeln('Дійсні корені відсутні')
  else
    begin
      if D=0 then
        begin X1:=-B/(2*A); X2:=X1 end
      else
        begin
          X1:=(-B + sqrt(D))/(2*A); X2:=(-B - sqrt(D))/(2*A);
        end;
      writeln('X1=',X1,'X2=',X2)
    end
  end.

```

5.13. Оператори циклу

Однією з найбільш поширених циклічних структур є цикл з параметром (або цикл "Для"). Керування обчисленнями в такому циклі здійснюється шляхом зміни параметра циклу x від x_n до x_k за законом арифметичної прогресії, тобто з певним кроком Δx . Кількість повторень в такому циклі визначається однозначно: вона дорівнює цілому значенню, що обчислюється

за формулою
$$n = \frac{x_k - x_n}{\Delta x} + 1.$$

В алгоритмічній мові Паскаль *цикл з параметром реалізується за допомогою оператора*, що має вигляд:

for I:= Io to Ik do S,

де *for* (для), *to* (до), *do* (виконувати) - службові слова; *I* - управляюча змінна (параметр циклу); *Io* - її початкове, а *Ik* - кінцеве значення; *S* - будь-який оператор (він може бути і складеним!).

Виконується оператор for так: управляюча змінна *I* набуває по черзі всіх значень від початкового *Io* до кінцевого *Ik* так, що кожне наступне значення більше попереднього на одиницю. Для кожного значення *I* виконується оператор *S*. Якщо ж початкове значення *I* в операторі *for* більше кінцевого ($Io > Ik$), то оператор *S* не виконується ні разу. Змінна *I* обов'язково

повинна бути цілого типу; на місці I_0 та I_k можуть бути як константи, так і змінні або вирази типу *integer*. Цикл закінчується після того, як параметр циклу набуде значення, що більше від кінцевого значення (оператор *S* при цьому вже не виконується!).

Арифметична прогресія, що зумовлює зміну параметра циклу, в наведеному прикладі є зростаючою з кроком $\Delta I = 1$. Крім такого кроку допускається ще $\Delta I = -1$, тобто арифметична прогресія може бути і спадаючою (в цьому випадку в операторі *for* замість службового слова *to* використовується службове слово *downto*). Зрозуміло, що в останньому випадку початкове значення параметру циклу повинно бути більшим, ніж його кінцеве значення.

Оператор *while* реалізує цикл "Поки" (див. рис. 5.3,в). Його загальний вигляд:

while P do S,

де *while* (поки), *do* (виконувати) - службові слова; *S* - оператор; *P* - умова (вираз логічного типу).

Оператор *S* в операторі *while* виконується повторно весь час, поки логічний вираз *P* має значення *true*, тобто поки умова *P* справджується. Як тільки вираз *P* набуде значення *false*, то виконання оператора *S* припиняється, керування передається наступному (за *while*) оператору. Якщо ж вираз *P* з самого початку має значення *false*, то оператор *S* взагалі не виконується (ні разу!). Оператор *S* в операторі *while* може бути як простим, так і складеним.

Оператор *repeat* реалізує цикл "До" (див. рис. 5.4,в). Він має вигляд:

repeat S₁; S₂;...; S_n until P,

де *repeat* (повторювати), *until* (поки не) - службові слова;

S₁; S₂;...; S_n - оператори, що виконуються повторно; *P* - логічний вираз.

На відміну від оператора *while* в операторі *repeat* управляючий вираз міститься після повторюваних операторів. Оператори *S₁; S₂;...; S_n* виконуються повторно до тих пір, поки управляючий вираз *P* не набуде значення *true* (істинно). Оскільки значення умовного виразу визначається в кінці циклу, то на відміну від оператора *while*, в операторі *repeat* тіло циклу виконується принаймні один раз.

Як приклади використання операторів циклу наведемо фрагменти програми визначення таблиці значень функції $y = \sin x$ в проміжку $[0;2]$ з кроком $\Delta x = 0,2$:

<pre>for I=1 to 10 do begin X:=(I-1)*0.2; Y:=sin(X); writeln(X,Y); end;</pre>	<pre>X:=0; while X<=2 do begin Y:=sin(X); writeln(X,Y); X:=X+0.2 end;</pre>	<pre>X:=0; repeat Y:=sin(X); writeln(X,Y); X:=X+0.2 until X>2;</pre>
---	--	---

Рис. 5.9. Фрагменти програми табулювання функції

*Питання теми 5, які винесені на самостійну проробку
і не включені в конспект*

1. Типові алгоритми обробки одновимірних масивів
2. Алгоритмічні мови програмування, компілятори і інтерпретатори, інтегровані середовища програмування
3. Програмування процедур обробки одновимірних масивів та матриць

Контрольні запитання до теми 5

1. Що таке алгоритм? Які властивості характерні для алгоритму?
2. Які способи опису алгоритмів ви знаєте?
3. Що являє собою блок-схема алгоритму ?
4. Назвіть основні символи блок-схем. Які функції алгоритму вони відображають?
5. В чому полягає суть метода покрокової деталізації ?
6. Назвіть основні принципи структурного підходу до розробки алгоритмів.
7. Які алгоритмічні структури відносяться до базових? В чому їх суть ?
8. Поясніть призначення та дію додаткових алгоритмічних структур.
9. Поясніть алгоритм обчислення суми скінченного числа доданків.
10. Чим відрізняються алгоритми обчислення сум і добутків ?
11. Поясніть алгоритм табулювання функції однієї змінної.
12. Чим відрізняється алгоритм табулювання функції багатьох змінних від алгоритму табулювання функції однієї змінної ?
13. Поясніть поняття "масив", "індекс елемента масиву", "розмірність масиву".
14. Чим зумовлене широке використання масивів в задачах обробки інформації? Поясніть алгоритм обчислення суми всіх елементів масиву.
15. Чим відрізняється алгоритм обчислення суми тих елементів, які відповідають певній умові, від алгоритму обчислення суми всіх елементів масиву ?
16. Складіть блок-схему алгоритму визначення суми додатних та кількості від'ємних елементів масиву.
17. Побудуйте алгоритм обчислення суми тих елементів масиву, які належать інтервалу [4; 12].
18. Поясніть алгоритм визначення найбільшого з елементів масиву.
19. Складіть алгоритм визначення порядкових номерів найбільшого та найменшого елементів заданого масиву.
20. В чому полягає задача сортування масивів ? Які переваги має упорядкований масив перед неупорядкованим ?
21. Поясніть алгоритм сортування масивів методом простого вибору.
22. Поясніть алгоритм сортування масивів методом "бульбашки". В яких випадках цей алгоритм найбільш ефективний ?
23. Побудуйте блок-схему алгоритму сортування масивів на основі пошуку мінімального елемента .
24. Перерахуйте основні елементи мови програмування. Поясніть поняття "оператор", "алфавіт мови", "структура програми".
25. Які дані можуть використовуватися в програмах ? Чим вони можуть відрізнятися в різних мовах програмування?
26. Що таке "стандартні функції"? Чим вони можуть відрізнятися в різних мовах програмування?

27. Поясніть поняття “арифметичний вираз”, “логічний вираз”. Чим вони можуть відрізнятися в різних мовах програмування?
28. Які символи входять до алфавіту мови Паскаль ?
29. Які імена можна надавати змінним в Паскалі ?
30. Які стандартні типи даних ви знаєте ? В чому полягає різниця між ними ?
31. Як записуються в Паскалі константи різних типів ?
32. Що таке стандартна функція ? Як вона використовується в програмі на Паскалі ?
33. Які стандартні арифметичні функції ви знаєте? Якими можуть бути аргументи та значення стандартних функцій ?
34. В якому порядку виконуються дії при обчисленні значень арифметичних виразів?
35. Сформулюйте правила запису арифметичних виразів.
36. Опишіть структуру програми в Паскалі.
37. Чому в програмах на Паскалі обов'язково повинен бути розділ опису змінних ? Як саме описуються змінні ?
38. Для чого в програмах на Паскалі використовуються іменовані константи ? Як вони визначаються ?
39. Опишіть структуру розділу операторів.
40. Як записується та як виконується оператор присвоєння ?
41. Які значення будуть мати змінні A і B після виконання послідовності операторів
42. $A:=4$; $B:=2$; $A:=A+2$; $B:=B-A$?
43. За допомогою яких операторів організовується введення даних ?
44. Опишіть можливості операторів write і writeln.
45. Як можна організувати введення даних з попереднім запитом про те, які саме дані необхідно вводити ?
46. Опишіть структуру умовного оператора. Як він виконується ?
47. В яких випадках доцільно використовувати скорочену форму умовного оператора?
48. Який оператор називається складеним ? В яких випадках він використовується ?
49. Що являє собою логічний вираз? Коли він використовується і які значення може приймати ?
50. Які операції можна використовувати в логічних виразах ? В якому порядку виконуються дії при визначенні значень логічних виразів?
51. Як записується та як виконується оператор case? Що може виступати як селектор в операторі case ?
52. Як записується та як виконується оператор for ?
53. Які зміни необхідно внести в форму оператора for , щоб параметр циклу змінювався з кроком -1 ?
54. Як можна використовувати оператора for для організації циклів, в яких параметр циклу змінюється з нецілим кроком ?
55. Який результат з'явиться на екрані дисплею після виконання операторів
56. `for I:=1 to 10 do X:=(I-1)*0.2; writeln(I,X);` ?
57. Як записується та як виконується оператор while ?
58. Чим відрізняється оператор while від оператора for ?
59. Як записується та як виконується оператор repeat ?
60. Чим відрізняється оператор repeat від оператора while ?
61. Порівняйте фрагменти програм, які табулюють функцію $y=\sin(x)$ за допомогою операторів for, while і repeat (рис. 5.8). Який з вказаних способів найбільше доцільний ?
62. Як в програмах на Паскалі описуються масиви ?
63. Як виконуються введення-виведення та інші операції по обробці масивів ?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Карімов І.К. Комп'ютерні методи та засоби розв'язання інженерних задач: Навч. посібник/ І.К.Карімов – Кам'янське: ДДТУ, 2017. - 283 с.
2. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів/ За редакцією О.І.Пушкаря. Вид. 2-ге , перероб., доп. - К.: Видавничий центр “Академія”, 2002.-704 с.
3. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / В.А.Баженов, П.С.Венгерський, В.М.Горлач та інші. - К.: Каравела, 2003,2004. - 464с.
4. Карімов І.К. Інформатика та програмування: Навч. посіб./ І.К.Карімов, О.І.Литвин, С.А.Нужна та інші. – Дніпродзержинськ:ДДТУ, 2014.-387 с.
5. Огурцов А.П., Карімов І.К., Мамаєв Л.М. Основи алгоритмізації та програмування: Навч.посіб. - К.:ІЗМН, 1997.- 153 с.
6. Долженков В.А. Microsoft Excel 2003 / В.А. Долженков, Ю.В. Колесников. - СПб. : БХВ- Петербург, 2004. - 1023 с.
7. Михеева В.Д. Microsoft Access 2003 / В.Д. Михеева, И.А. Харитонов. - БХВ-Петербург,2004. -1069 с.
8. Кирьянов Д.В. Mathcad 14. - СПб.: БХВ-Петербург, 2007.- 686 с.

НАВЧАЛЬНЕ ВИДАННЯ

Конспект лекцій з дисципліни “Комп’ютерні методи та засоби розв’язання інженерних задач” для здобувачів вищої освіти першого (бакалаврського) рівня зі спеціальностей 101 “Екологія”, 161 “Хімічні технології та інженерія”, 162 “Біотехнології та біоінженерія” (частина 1)

Укладач: Карімов Іван Кабіденович

Підписано до друку 19.12. 2019 р. Формат А5
Обсяг 3,6 д.а. Тираж 50 прим. Замовлення № 281
Безкоштовно. 51918, м. Кам’янське, Дніпробудівська, 2.