

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

**ЧЕРНІГІВСЬКИЙ ДЕРЖАВНИЙ ІНСТИТУТ
ЕКОНОМІКИ І УПРАВЛІННЯ**

Кафедра «Економічної кібернетики та інформатики»

ПРАКТИКУМ з дисципліни

«Алгоритмізація та програмування»

”Ухвалено”

на засіданні кафедри
економічної кібернетики
та інформатики
Протокол № 6
від «20» грудня 2011р.

Чернігів 2011

УДК. 004.42, 519.85

Рекомендовано до друку

Вченою радою Чернігівського державного інституту
економіки і управління

(протокол № 5 від « 27» грудня 2011р.)

Рецензенти: Казимир В.В., д.т.н., професор;

Сахно Є.Ю., д.т.н., професор.

Авторський колектив:

Зацерковний В.І., к.т.н., доцент;

Гур'єв В.І., к.т.н., доцент;

Сімакін Ю.С., викладач;

Практикум з дисципліни «Алгоритмізація та програмування» – Чернігів,: ЧДІЕУ, 2011. – 164 с.

Курс «Алгоритмізація та програмування» має на меті сформувані у студентів науково обґрунтовані погляди на сучасні технологічні процеси в інформаційній галузі. Лабораторний практикум допоможе ознайомитися з основними поняттями теорії алгоритмів, з найбільш поширеними алгоритмічними системами; засвоїти принципи організації алгоритмічних процесів та форми їхньої реалізації; ознайомитися з основними поняттями створення програм у середовищі Turbo Pascal; навчитися будувати базові алгоритми процесів пошуку, передавання, обробки інформації в різних інформаційних технологіях; навчитися здійснювати аналіз та контроль алгоритму на різних етапах життєвого циклу програмного продукту; особливостями програмування, принципами організації та реалізації програм; складати супровідну документацію.

Практикум призначений для студентів спеціальності «Економічна кібернетика», а також для інших напрямів підготовки, що вивчають основи програмування.

© ЧДІЕУ 2011р

ПЕРЕДМОВА

„У кожній дисципліні є стільки науки, скільки у ній математики ”
Д.І. Менделєєв

Стрімкий розвиток комп'ютерної техніки та її програмного забезпечення – одна з характерних рис сучасного розвитку суспільства. Інформаційні технології, основним елементом яких є комп'ютер, використовуються практично у всіх сферах людської діяльності. Комп'ютер потрібний людині для вирішення різноманітних практичних задач. Як відомо, персональні комп'ютери – це пристрої для збереження та обробки інформації. На відміну від телевізора або магнітофона, які здійснюють тільки задалегідь закладені у них функції, персональні комп'ютери можуть вирішувати будь-які задачі¹. Проте, для цього комп'ютеру необхідно попередньо скласти, на зрозумілій йому мові, точну та докладну послідовну інструкцію (програму², алгоритм), яка буде вказувати йому, як і що саме треба робити з інформацією.

Прикладами таких задач можуть бути: опис поведінки тіла, що рухається у середовищі з опором; опис наслідків ядерної війни; побудова оптимального варіанту транспортних перевезень; прогнозування результатів скиду промислових відходів у водоймище тощо. Незважаючи на значні розбіжності задач, у наведених прикладах проглядаються й загальні моменти у процесі їх вирішення:

☑ по-перше, потрібно виділити систему і побудувати її інформаційну модель – вона визначає набір даних і їх взаємозв'язки;

☑ по-друге, повинний бути встановлений певний порядок обробки даних.

Це ланки однієї послідовності рішення, тому представляється цілком виправданим розглянути їх спільно.

¹ Задача (problem, task) – проблема, що підлягає вирішенню.

² Комп'ютерна програма – запис алгоритму розв'язання задачі у вигляді послідовності команд або операторів мовою, яку розуміє комп'ютер.

У загальному випадку, обробка складається з перетворення первинних даних за певними правилами, у результаті чого з'являються нові дані. Безперечно, важливим виявляється й та обставина, що перетвореннями повинен займатись певний технічний пристрій, причому в автоматичному режимі (тобто без участі людини на кожному етапі перетворення). У зв'язку з цим, виникає ряд взаємопов'язаних задач, що потребують:

① визначення правил обробки інформації з урахуванням того, що вона представлена у дискретній формі;

② встановлення, яким вимогам повинен задовольняти пристрій, що здійснює обробку;

③ визначення того, яким чином дані та послідовність обробки може бути представлена для виконання пристроєм.

Будь-яка діяльність завжди має мету. Мета діяльності – це майбутній бажаний ідеальний результат діяльності. Людина що виконує певну роботу, зазвичай розуміє, для чого вона це робить і що хоче отримати в результаті цієї діяльності. Діяльність може мати одну мету (ціль), або деяку їх сукупність. Цілі можуть створювати ієрархію, у якій існують як проміжні, так і цілі більш високого рівня.

Алгоритмічність діяльності полягає у тому, що будь-яка діяльність складається з окремих кроків. Осмислене застосування інформаційних технологій має ґрунтуватись на розумінні принципів побудови сучасного програмного забезпечення, яке включає широкі можливості й розвинутий інтерфейс користувача. Тому необхідно оволодіти фундаментальними знаннями у цій галузі, починаючи з основ.

Наука „інформатика”, а відповідно і теорія алгоритмів, виникли недавно. Тому природно, що вони багато „запозичують” у старших наук-сусідів.

Фізика. Вплив фізики на інформатику проявився тільки останніми роками і викликав справжній бум нових досліджень, „тільки у фізиці сіл, все інше нуль”. Центральним інтегруючим напрямом цих наук, стало вивчення нестандартних моделей обчислень. Як довів Річард Фейнман, фізичні явища, такі як спин електрона, можуть бути використані для обчислень. Сучасні дослідження засвідчили, що квантові алгоритми не зводяться до звичайної (що використовує біти) моделі обчислень. Отже, простір ефективно вирішуваних завдань розширюється. Згадаємо тут також такі теми, як обчислення з дійсними числами (Real RAM), оптичний комп'ютер і навіть (!) більярдні обчислення.

Теорія ймовірностей. Одна з найбільш уживаних математичних теорій в інформатиці. Два ключові напрями: оцінка часу роботи алгоритму „в середньому” і ймовірнісні алгоритми. Дослідження в теорії складності показують, що детерміновані алгоритми не завжди дають якнайкращі рішення. Більш того, на практиці ймовірнісні алгоритми можуть працювати помітно швидше навіть за наявності альтернативного детермінованого алгоритму (наприклад, завдання перевірки на простоту).

Біологія. Для вирішення найважчих задач, які виникають у практичній діяльності людини, вона зазвичай звертається за допомогою до природи. Що робити, якщо ми бажаємо знайти розв'язок для таких складних завдань, які не формалізуються взагалі або формалізуються погано? Наприклад класифікація і розпізнавання образів? Вихід один – подивитися, як ці задачі вирішуються природою. Так виникла ідея симуляції й моделювання обчислювальних здібностей нейронів. Запропонована модель отримала назву нейронних мереж. Потім був зроблений наступний крок. Важливо не тільки як людина вирішує конкретну задачу (немовлята досить погано пристосовані до життя), важливо з якою феноменальною швидкістю людина навчається протягом свого життя. Так виникла теорія машинного навчання (machine learning).

Теорія графів і комбінаторика. Сучасні комп'ютери працюють з дискретними даними. Найбільш простими об'єктами такого роду є натуральні числа, послідовності, кінцеві множини і графи. Тому їх основні властивості, що вивчаються у відповідних розділах математики, використовуються в теорії алгоритмів дуже часто. Коли фахівці з алгоритмів доростуть до роботи з більш складними об'єктами, напевно й наступні рівні математики знайдуть своє застосування.

Математична логіка. Власне з неї й виросла теорія алгоритмів. Мрією математиків початку ХХ ст. було створення єдиного (обчислювального) методу вирішення математичних проблем. На жаль, як показали у своїх дослідженнях Черч і Тюрінг, існують обчислювально нерозв'язні завдання. Проте, логіка дала потужний апарат виразу різних властивостей математичних об'єктів і формальні правила роботи з цими властивостями. У сучасній теоретичній інформатиці логіка використовується для розробки нових мов програмування, завдань автоматичної верифікації програм і для вивчення складності обчислювальних завдань (proof complexity).

Функціональний аналіз. Виявляється, що й безперервна математика необхідна у розробці алгоритмів. Це стосується, головним чином, комп'ютерної обробки явищ, що мають безперервну природу. Перед усім, це цифрова обробка аудіо- і відеозаписів. Такі широко використовувані стандарти, як MPEG і JPG містять ідеї, узяті з властивостей перетворення Фур'є і активно використовують дискретний аналог операції згортання.

Першим прикладним напрямом, що виділив теорію обчислень в окремий напрям, стали чисельні розв'язання рівнянь з фізики, розрахунки у атомній галузі й управління космічними кораблями і супутниками.

Наступним джерелом величезної кількості обчислювальних завдань стали питання оптимізації задач в економіці. До основних досягнень варто віднести формулювання задач лінійного програмування (Канторович), симплекс-метод, алгоритм Кармаркара і алгоритм Хітчина.

Успіхи математичної статистики і розвиток вимірювальних приладів й рентгенів, породили необхідність в алгоритмізації автоматичної діагностики і обробки даних томографії. Зараз з величезною швидкістю проводиться впровадження комп'ютерної техніки у самих різних напрямках медицини.

Зі зростанням обсягів інформації виникла необхідність в ефективних механізмах її збереження й використання. Саме тому, алгоритми обробки запитів в базах даних належать до найбільш поширених.

Як відомо найбільшу частку інформації людина сприймає зором. Тому недивною є зацікавленість до алгоритмів обробки зображень, моделювання пейзажів і руху за уявною місцевістю (віртуальна реальність). Величезні зусилля витрачаються на розробку нових алгоритмів стиснення растрових зображень, аудіо і відео потоків (MPEG4, JPEG).

Головним напрямом розвитку інформаційних технологій останніх двох десятиліть став Інтернет і розподілені обчислення. Теорія алгоритмів тут знаходить своє застосування в завданнях маршрутизації пакетів (TCP/IP і DNS) й пошукових системах. Небувалий успіх системи Google став, мабуть, саме тим випадком, що надовго буде нагадувати, коли проста математична ідея (алгоритм PageRank) привів до феноменального комерційного успіху.

Особливе значення має вирішення задач, успіх у яких неможливо сформулювати строго – це так звані задачі штучного інтелекту, зокрема: автоматичне розпізнавання мови, відбитків пальців, осіб людей, системи розпізнавання „свій – чужий”, автоматична класифікація, автоматичний контроль якості тощо.

Врешті-решт, теорія алгоритмів прийшла до того, що об'єктами обробки стали самі алгоритми. Основними завданнями є автоматична верифікація і оптимізація програм й системи з розпаралелювання виконання програм на багатопроесорних обчислювальних системах. Наступним напрямом є лінгвістичні алгоритми: перевірка орфографії, автоматичний переклад, програми, що „розмовляють”, робота з граматиною.

Нарешті, комп'ютерам почали довіряти все більш важливі і складні задачі. Методи машинного навчання використовуються у розробці роботів (особливо принадною звучить створення футбольної команди роботів, здатної виграти у чемпіонів світу 2050р.). Природно чекати, що час поширення пристроїв, оснащених датчиками й здатних самостійно ухвалювати оптимальне рішення, наступить дуже скоро. Досить популярним прикладним напрямком в останній час стали дослідження у біоінформації: обчислення (відновлення) геномів і побудова найбільш вірогідного ланцюжка мутацій, який переводить один генотип у інший.

РОЗДІЛ I

„ПРОГРАМУВАННЯ В СЕРЕДОВИЩЕ – Turbo-Pascal”

Лабораторна робота № 1

1. Мета роботи

1. Знайомство з середовищем програмування Turbo Pascal.

2. Вивчення структури програми, стандартних функцій, оператора присвоювання і процедур введення-виведення.

2. Завдання роботи

Навчитися програмувати на мові Turbo Pascal з використанням стандартних функцій.

3. Порядок виконання роботи

3.1. Завантажити середовище програмування Turbo-Pascal. Ознайомитися з призначенням основних пунктів меню, змістом рядка стану і роботою у довідковій системі.

3.2. Скласти і налагодити програму, яка включає :

– повідомлення про введення двох чисел;

– введення значень двох дійсних чисел;

– виведення на друк у вигляді таблиці суми, різниці, добутку і середньоарифметичного значення 1-го і 2-го числа з різним числом знаків у дробовій частині.

3.3. Скласти і відредагувати програму обчислення функції $y=f(x)$ з використанням стандартних функцій мови Turbo Pascal у відповідності із заданим викладачем варіантом (див. додаток А).

3.4. Провести розрахунки для $x=0.1$, $x=0.2$, $x=0.3$, $x=0.4$ та $x=0.5$.

3.5. Роздрукувати створену програму й отримані результати у вигляді таблиці.

4. Вимоги до звіту

Звіт про виконану роботу повинен включати:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонитули з П.І.Б. виконавця і шифром групи;
- номер варіанта для виконання завдання і умови свого варіанта;
- блок-схеми розв'язання задачі;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п.7.

5. Теоретичні положення

5.1. Елементи мови

5.1.1. Алфавіт мови

При запису програм дозволяється використовувати:

- літери латинського алфавіту **A-Z** (у будь-якому регістрі), а також знак підкреслення **_**;
- літери українського або російського алфавітів **А-Я**;
- цифри **0-9**;
- спеціальні символи **> < = + - / * [] () { } . , ; ^ @ ' \$ #**
- пари символів (їх не можна розділяти пробілами) **< > <= >= := (**) (..)**
- пробіли (розглядаються як обмежники ідентифікаторів, констант, чисел, зарезервованих слів).

5.1.2. Ідентифікатори

Ідентифікатори – неподільні послідовності символів алфавіту, які використовуються для позначення констант, змінних, процедур, функцій тощо.

Ідентифікатор повинен починатися з букви або символу підкреслення, не повинен містити пробілів і спеціальних символів. Наприклад:

```
name  
WorkPhone  
_SUM1
```

5.1.3. Константи

Для позначення констант можуть використовуватися числа, логічні константи, символи і рядки символів.

Цілі числа записуються зі знаком або без нього за звичайними правилами і можуть мати значення:

від -2147483648 до +2147483647.

Дійсні числа записуються зі знаком або без нього з використанням десяткової крапки і/або експонентної частини. Експонентна частина позначається символом *e* або *E*, за яким можуть йти знаки «+» або «-» і десятковий порядок. Символ *e* (*E*) означає десятковий порядок і значення «помножити на 10 в степені». Наприклад, запис 3.14E5 означає $3,14 \cdot 10^5$, а запис -17e-2 – це $-17 \cdot 10^{-2}$ або 0,17.

Логічна константа – це або слово FALSE (хибність) або слово TRUE (істина).

Символьна константа – це будь-який символ, взятий в апострофи:

'z' – символ z;

'ф' – символ ф.

Строкова константа – послідовність символів, взятих в апострофи. Якщо в рядку потрібно вказати сам символ апострофа, він подвоюється, наприклад:

'Це – рядок символів ';

'That"s string'.

Рядок символів може бути порожнім – не мати жодних символів в апострофах, які б його обрамляли.

5.1.4. Вирази

Вираз задає порядок виконання дій над елементами даних і складається з операндів (констант, змінних, функцій, круглих дужок і знаків операцій).

Дії у виразі виконуються зліва направо з дотриманням підпорядкованості (в порядку спадання):

1) Not (логічне заперечення);

2) * (множення), / (ділення), div (цілочислове ділення), mod (цілочислове ділення з залишком по модулю), and (логічна операція “І”);

3) + (додавання), - (віднімання), or (логічна операція “АБО”);

4) операції відношень: = (дорівнює), <> (не дорівнює), < (менше), > (більше), <= (менше або дорівнює), >= (більше чи дорівнює).

Для зміни порядку виконання дій використовуються круглі дужки. Кількість дужок, які відкриваються, повинна дорівнюватись кількості дужок, що закриваються. Будь-який вираз в дужках має пріоритет, тобто обчислюється раніше, ніж виконується операція перед дужками.

Усі складові частини виразу записуються в один рядок, наприклад, вираз $\frac{a+bx}{c+d}$ записуємо у вигляді $(a+b*x)/(c+d)$.

У вираз можуть входити функції. Частіше уживані функції називають стандартними. Для роботи з ними не треба ні замовляти бібліотеку, ні описувати попередньо їх у програмі. Приклади стандартних математичних функцій:

$ABS(x)$ – модуль x ($|x|$);

$SQR(x)$ – квадрат числа x (x^2);

$SQRT(x)$ – квадратний корінь з x (\sqrt{x});

$LN(x)$ – натуральний логарифм від x ($\ln x$);

$EXP(x)$ – e в степені x (e^x);

$SIN(x)$ – синус x ($\sin x$);

$COS(x)$ – косинус x ($\cos x$);

$ARCTAN(x)$ – арктангенс x ($\arctg x$).

Аргумент цих функцій може бути як дійсним, так і цілим. Результат – завжди дійсний.

5.2. Типи даних

Будь-які дані – константи, змінні, значення функцій або вирази в Турбо-Паскалі характеризуються своїми типами.

Тип визначає множину припустимих значень, які може мати той чи інший об'єкт, а також множину припустимих операцій, які можуть до нього застосовуватись. Всі типи даних поділяються на дві групи – прості і складові.

До простих (скалярних) типів відносяться:

$INTEGER$ – дані цього типу можуть приймати тільки цілі значення (додатні, від'ємні, 0) в діапазоні від -32768 до $+32767$;

$REAL$ – величини цього типу можуть приймати тільки дійсні значення (числа з дробовою частиною, ціла частина від дробової відокремлюється крапкою);

$BOOLEAN$ – логічний тип, який приймає два значення $TRUE$ (істина) і $FALSE$ (хибно);

$CHAR$ – символ.

Прикладом складового (структурованого типу) може слугувати тип *STRING* – рядок символів. Цей тип широко використовується для обробки текстів.

5.3. Структура програми

Структура програми повинна бути наступною:
<Заголовок програми>
{Блок описів}
BEGIN
{Розділ виконуваних операторів}
END.

5.3.1. Заголовок програми

В заголовку вказується ім'я програми. Загальний вигляд заголовку:

program n;
де *n* – ім'я програми.

Заголовок програми не є обов'язковим, його можна опускати без будь-яких наслідків для програми.

5.3.2. Блок описів

В блоці описів оголошуються ідентифікатори типів, констант, змінних, а також мітки, процедури і функції. Блок описів може складатися з п'яти розділів, які повинні йти в строго визначеному порядку:

- 1) розділ міток (*label*);
- 2) розділ констант (*const*);
- 3) розділ типів (*type*);
- 4) розділ змінних (*var*);
- 5) розділ процедур і функцій.

5.3.2.1. Розділ міток (*label*)

Будь-який виконуваний оператор може бути позначений міткою – додатною константою, яка містить не більше 4-х цифр. Мітка відокремлюється від оператора двокрапкою. Всі мітки, які зустрічаються в програмі, повинні бути описані у розділі *label*.

Загальний вигляд:
label l1, l2, l3...;
тут *l1, l2, l3...* – мітки.

Приклад.

label 20;

Нехай оператор $a:=b$; має мітку 20. Тоді цей оператор виглядає так:

20: $a:=b$;

5.3.2.2. Розділ констант (const)

Якщо в програмі використовуються константи, які мають достатньо громіздкий запис (наприклад, число π з 8-ма знаками), або змінні константи (наприклад, для завдання варіанта програми), то такі константи зазвичай позначаються іменами і описуються в розділі *const*. Це робить програму більш наочною і зручною при налагодженні і внесенні змін.

Загальний вид:

const a1 = c1; a2 = c2; ...

Тут *a1, a2, ...* – ім'я константи, *c1, c2, ...* – значення константи.

Приклад.

const pi=3.14; c=2.7531;

5.3.2.3 Розділ типів (type)

Якщо в програмі вводиться тип, відмінний від стандартного, то цей тип описується в розділі *type*:

type t1=<вид туну>;

t2=<вид туну>;

.....

де *t1* і *t2* – ідентифікатори типів, які будуть вводиться.

Приклад.

Type color=(red, yellow, green, blue);

Тут описаний тип *color*, який задається переліком кольорів.

5.3.2.4. Розділ змінних (var)

В розділі *var* вводиться ім'я кожної змінної і вказується до якого типу ця змінна належить:

var v11, v12, ...: type1;

v21, v22, ...: type2; ...

Тут *v11, v12, ...* – імена змінних; *type1* – тип змінних *v11, v12, ...*; *type2* – тип змінних *v21, v22, ...*

Приклад

var k,i,j:integer; a,b:real;

5.3.2.5. Розділ процедур і функцій

Ті алгоритми, які оформлюються як підпрограми (процедури і функції), розміщуються в головній програмі після розділу *var* і перед операндом *begin* програми.

5.3.3. Розділ дій (операторів)

Ця частина програми починається з ключового слова *begin* і закінчується словом *end*, після якого повинна стояти крапка (*end.*). Розділ дій – це виконувана частина програми, яка складається з операторів.

5.4. Коментарі

Коментар – це довільна послідовність будь-яких символів, які пояснюють текст програми. Коментар дозволяється вставляти у будь-яке місце програми, де за змістом може стояти пробіл. Обмеженням коментаря слугують фігурні дужки «{» і «}», а також пари символів: «(*)» – ліворуч від коментаря і «*)» – праворуч від нього:

{ Це коментар }

(* Це також коментар *)

5.5. Оператор присвоювання

Під операторами в мові Паскаль розуміють опис дій. Оператори відокремлюються один від одного крапкою з комою. Якщо оператор стоїть перед *end*, *until* або *else*, то в цьому випадку крапка з комою не ставиться.

Загальний вид оператора присвоювання:

$v:=a;$

тут v – змінна, a – вираз, $:=$ – операція присвоювання. Вираз a може містити константи, змінні, назви функцій, знаки операцій і дужки. В операторі $v:=a$ зміна v і вираз a повинні мати один і той же тип.

Приклади.

$f:=3*c+2*\sin(x);$

$x:=x+1;$

Зауваження. Дозволяється присвоювати змінній типу *real* вираз типу *integer*. Але не можна присвоювати змінній типу *integer* вираз типу *real*.

5.6. Процедура введення інформації

Загальний вид:

Read (v1, v2, ...,vn);

або

Readln (v1, v2, ...,vn);

тут **v1, v2, ...,vn** – ідентифікатори змінних.

Значення змінних вводяться з клавіатури і повинні відповідати типам змінних. У випадку використання процедури *readln*, після введення відбувається перехід на наступний рядок.

5.7. Процедура виведення інформації на друк

Загальний вид оператора:

write(p1, p2, ..., pn);

або

writeln(p1, p2, ..., pn);

Тут *p1, p2, ..., pn* – список виразів, значення яких виводяться на друк.

Оператор *write* залишає курсор в кінці виведеного рядка тексту.

У випадку використання процедури *writeln*, після друку відбувається перехід на наступний рядок.

Крім значень виразів, на друк можна виводити і довільний набір символів, укладений в апострофи, наприклад:

writeln('p=',p);

Цей оператор виконується так: спочатку виводяться символи в апострофах. Потім виводиться значення змінної *p*, наприклад 13.5. На екрані в результаті роботи оператора з'явиться:

p=13.5

Приклад 1. Обчислити довжину кола радіуса 5,785.

program t10;

(Програма обчислення довжини кола*)*

const r=5.785;

var l:real;

begin

*l:=2*3.1416*r;*

writeln(' l=',l);

end.

```

або
program t11;
var r:real;
begin
readln(r);
writeln(' l=',2*3.1416*r);
end.

```

Є можливість задати ширину поля (число позицій) M для виведеної величини P:

```
Write (P1:M1, P2:M2, ...PN:MN);
```

Для дійсних чисел можна задавати поля M і N, де M – загальне число позицій, відведених під все число, N – число позицій під його дробову частину.

Наприклад,
Write (P:10:2);

Тут під P відводиться 10 позицій, 2 з них під дробову частину.

Приклад 2. Призначення наступної програми – ввести з клавіатури два цілих числа, знайти результат ділення першого числа на друге, вивести числа і отриманий результат на екран у вигляді таблиці.

```
Program Input_Output;
```

{Програма вводить два цілих числа і виводить дробове від ділення 1-го на 2-е}

```
Var
```

```
n1,n2:Integer; {n1 і n2 – цілі, що вводяться }
```

```
x:real; {x – результат}
```

```
Begin
```

```
Write(n1=); {Повідомлення про введення n1}
```

```
ReadLn(n1); {Введення n1}
```

```
Write(n2=); {Повідомлення про введення n2}
```

```
ReadLn(n2); {Введення n2}
```

```
X:=n1/n2; {Обчислення результату}
```

```
WriteLn(-----); {Друк таблиці}
```

```
WriteLn(| n1 | n2 | Часткове |);
```

```
WriteLn(-----);
```

```
WriteLn(n1:8,n2:8,x:8:4);{Виведення n1, n2 і x}
```

```
WriteLn(-----);
```

```
End.
```


6. Методичні рекомендації

6.1. Запуск середовища програмування Turbo Pascal виконується будь-яким відомим зі стандартних способів запуску, передбачених в ОС Windows (наприклад, за допомогою ярлика на робочому столі).

Для переходу до вибору команд головного меню використовується клавіша <F10>. Для повернення в режим редагування потрібно натиснути клавішу <Esc>.

Для отримання довідки використовуються клавіші:

<F1> – отримання контекстно-залежної довідки;

<SHIFT+F1> – вибір довідки зі списку доступних довідкових повідомлень;

<CTRL+F1> – отримання довідки про потрібну стандартну процедуру, функцію, про стандартну константу або змінну.

6.2. При розв'язуванні задач можна скористатися прикладами програм з п.5.7. Текст програми набирається у текстовому редакторі середовища Turbo Pascal.

Після заповнення чергового рядка потрібно натиснути клавішу <ENTER>, для переведення курсору на наступний рядок.

Найбільш часто використовувані команди редактора Turbo Pascal:

– зміщення курсору:

<Page Up> – на сторінку вгору;

<Page Down> – на сторінку вниз;

<Home> – на початок поточного рядка;

<End> – в кінець поточного рядка;

<Ctrl+ Page Up> – на початок тексту;

<Ctrl+ Page Down> – в кінець тексту.

– команди редагування:

<Backspace> – витирає символ ліворуч від курсору;

<Delete> – витирає символ, на який вказує курсор;

<Ctrl+Y> – витирає рядок, на якому розташований курсор;

<Enter> – вставляє новий рядок, розрізає старий;

<Ctrl+Q L> – відновлює змінений рядок (діє, якщо курсор не залишив рядок після його зміни).

- робота з блоками:

<Ctrl+K B> – починає виділення блоку;

<Ctrl+K K> – закінчує виділення блоку (крім того, блок можна виділити з допомогою миші);

<Ctrl+K Y> – знищує виділений блок;

<Ctrl+K C> – копіює блок;

<Ctrl+K V> – переміщує блок на нове місце;

<Ctrl+K W> – записує блок в файл;

<Ctrl+K R> – зчитує блок з файлу;

<Ctrl+K P> – друкує блок.

Набраний текст програми запишіть у файл. Клавішею <F2> викликається вікно діалогу, в якому потрібно задати ім'я файлу.

Після підготовки тексту програми потрібно спробувати виконати її. Для цього потрібно відкомпілювати програму, зв'язати її (якщо це необхідно) з бібліотекою стандартних програм і функцій, завантажити в оперативну пам'ять і передати їй управління. Ця послідовність дій – прогін програми – здійснюється після натискання клавіш <CTRL+F9>.

Якщо в програмі немає синтаксичних помилок, то всі дії виконуються послідовно одна за одною, при цьому у невеликому вікні з'являється інформація про кількість відкомпільованих рядків і об'єму доступної пам'яті. Перед передачею управління програмі, яка завантажується, середовище Turbo Pascal виводить на екран вікно прогону програми, а після завершення роботи відновлює на екрані вікно редактора.

Якщо на якомусь етапі середовище Turbo Pascal знайде помилку, воно припиняє подальші дії, відновлює вікно редактора і поміщає курсор на той рядок програми, при компіляції чи виконанні якого виявлена помилка. При цьому, у верхньому рядку редактора з'являється діагностичне повідомлення про причину помилки. Все це дозволяє налагодити програму – усунути в ній синтаксичні помилки і переконатися у правильності її роботи. Деякі повідомлення про помилки, їх переклад і пояснення наведені в додатку Б.

Оператором *read* (або *readln*) викликається вбудована процедура введення даних і програма зупиняється в очікуванні введення. Необхідно набрати на клавіатурі потрібні дані і натиснути клавішу <ENTER>.

За допомогою клавіш <ALT+F5> у будь-який момент можна переглянути дані, видані на екран в результаті прогону програми.

7. Контрольні запитання

- 7.1. Як в програмному середовищі Turbo-Pascal описуються змінні?
- 7.2. Які бувають типи змінних?
- 7.3. Який вид має оператор присвоєння?
- 7.4. Яким символом відокремлюються один від одного оператори у програмі?
- 7.5. В яких випадках після оператора не ставляться крапка з комою?
- 7.6. Яка процедура служить для виведення інформації на друк?
- 7.7. Яка процедура служить для введення значень з клавіатури?
- 7.8. Які функції служать для обчислення квадрата, квадратного кореня, модуля, експоненти числа або числового виразу?
- 7.9. Які стандартні тригонометричні функції існують в середовищі Turbo Pascal?
- 7.10. Як в середовищі Turbo Pascal запустити програму на виконання?

8. Література

1. Інформатика. Базовий курс / Симонович С.В. і ін.– СПб: Питер, 2000. – 640 с.
2. Семашко Г.А., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація та програмування”, Чернігів, ЧДЕІУ – 2011.
5. Войтюшенко Н.М. Інформатика і комп'ютерна техніка: Навч. пос. з баз.підготовки для студ.екон. і техн.спеціальностей ден. і заочн. форм навчання. - К.: Центр навчальної літератури, 2006. – 568 с.
6. Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч.посіб. Вид. 2-ге, перероб., доп. - К.: Академвидав, 2007.- 416 с.(Альма-матер).
7. Інформатика і комп'ютерна техніка: Навчальний посібник / За ред. М.Є. Рогози. - К.: Видавничий центр «Академія», 2006. – 368 с.

ДОДАТОК А

Номер варіанта	Функція	Номер варіанта	Функція
1	$y = \sqrt{x^2 + 5x + 8}$	16	$y = \sqrt{x^3 - 7x^2 + 8x + 6}$
2	$y = \frac{3x^2 - 4}{x + 1}$	17	$y = \frac{9x^2 - 4x + 1}{x + 7}$
3	$y = x - \sqrt{x + 2}$	18	$y = x - \sqrt{x + 3}$
4	$y = \frac{7x + 4}{3x + 5} + \cos x$	19	$y = \frac{7x^3 + 2}{3x^2 - 1} + \operatorname{tg} x$
5	$y = x^3 + 4x^2 + x - 2$	20	$y = 8x^5 + 7x^4 + 3x^2 - 2$
6	$y = x + \sin 3x$	21	$y = 5x + \cos 3x$
7	$y = \cos x + \sqrt{x}$	22	$y = \sin x + \sqrt{x^3}$
8	$y = \frac{\sin x}{3} + \frac{\operatorname{tg} x}{6}$	23	$y = \frac{\cos x}{6} + \frac{\operatorname{arctg} x}{3}$
9	$y = \sqrt{x + 2} - \sqrt{x + 3}$	24	$y = \sqrt{x^3 - 22} - \sqrt{x^2 + 4}$
10	$y = 2x + \sin 2x$	25	$y = 3x + \sin 3x$
11	$y = 5 - \frac{\cos 3x}{3}$	26	$y = 7 + \frac{\sin 3x}{6}$
12	$y = \sqrt{x} - \sqrt{x + 1}$	27	$y = \sqrt{x^2} - \sqrt{x^3 + 1}$
13	$y = x - 5 + x^2$	28	$y = x^3 - 5 + x^5$
14	$y = e^x + 1$	29	$y = e^{3x} + 54$
15	$y = \sqrt{ \sin x }$	30	$y = \sqrt{ \cos x }$

ДОДАТОК Б

Повідомлення про помилки

2 **Identifier expected** (Не вказаний ідентифікатор).

В цьому місці повинен знаходитися ідентифікатор.

3 **Unknown identifier** (Невідомий ідентифікатор)

Цей ідентифікатор не був описаний.

4 **Duplicate identifier** (Подвійний ідентифікатор).

Спроба двічі описати один і той же ідентифікатор.

5 **Syntax error** (Синтаксична помилка).

В початковому тексті знайдений неприпустимий символ.

10 **Unexpected end of file** (Не знайдений кінець файлу).

Причини цього повідомлення можуть бути наступні:

- початковий файл закінчився перед останнім **END** основного розділу операторів; ймовірно в програмі неоднакова кількість операторів **BEGIN** і **END**;

- не закінчений коментар.

11 **Line too long** (Занадто довгий рядок).

Максимальна довжина рядка, оброблювана компілятором, дорівнює 126 символам.

12 **Type identifier expected** (Тут потрібний ідентифікатор типу).

Не зазначений тип ідентифікатора.

16 **Disk full** (Диск заповнений).

Потрібно видалити деякі файли або скористатися новим диском.

20 **Variable identifier expected** (Відсутній ідентифікатор змінної).

На цьому місці повинний бути ідентифікатор змінної.

21 **Error in type** (Помилка в об'яві типу).

Об'ява типу не може починатися з цього символу.

26 **Type mismatch** (Невідповідність типу).

Це повідомлення може бути викликане наступними причинами:

- несумісні типи змінної і виразу в операторі присвоювання;
- тип виразу не сумісний з типом індексу при об'яві масиву;
- несумісні типи операндів у виразі.

33 **Label identifier expected** (Потрібний ідентифікатор мітки)

Мітка не позначена за допомогою ідентифікатора, як це потрібно з контексту програми.

36 **BEGIN expected** (Потрібний BEGIN)

37 **END expected** (Потрібний END)

38 **Integer expression expected** (Потрібний вираз типу Integer).

41 **Operand types do not match operator** (Типи операндів не відповідають операції). Дана операція не може бути застосована до зазначених операндів.

42 **Error in expression** (Помилка у виразі)

Даний символ не може брати участь у виразі зазначеним чином. Можливо, не вказана операція між двома операндами.

50 **DO expected** (Потрібний оператор DO)

57 **THEN expected** (Потрібно THEN)

58 **TO or DOWNTO expected** (Потрібно TO чи DOWNTO)

62 **Division by zero** (Ділення на нуль)

Попередня операція намагається виконати ділення на нуль.

64 **Cannot Read or Write variables of this type** (Немає можливості зчитати або записати змінні даного типу).

Порушені наступні обмеження:

- процедури **READ** і **READLN** можуть зчитувати змінні символьного, цілого, дійсного і строкового типів;

- процедури **WRITE** і **WRITELN** можуть виводити змінні символьного, цілого, дійсного, логічного і строкового типів.

76 **Constant out of range** (Константа порушує границі).

Можливі причини повідомлення:

- спроба вказати індекс масиву, який виходить за його границі;

- спроба присвоїти змінній значення, яке виходить за границі, пропустити для типу цієї змінної.

79 **Integer or real expression expected** (Потрібний вираз дійсного чи цілого типу).

81 **Label already defined** (Мітка вже визначена).

Дана мітка уже позначає оператор.

85 **«;» expected** (Потрібно вказати «;»).

97 **Invalid FOR control variable** (Неправильний параметр циклу оператора FOR).

98 **Integer variable expected** (Потрібна змінна цілого типу).

Попередня змінна повинна мати цілий тип.

103 **Integer or real variable expected** (Потрібна змінна типу INTEGER or REAL).

113 **Error in statement** (Помилка в операторі).

Даний символ не може бути першим символом в операторі.

207 **Invalid floating point operation** (Неприпустима операція з комою що плаває) .

Можливі причини повідомлення:

- негативний аргумент функції SQRT;

- аргумент функції LN дорівнює нулю чи має негативне значення.

Лабораторна робота № 2

АЛГОРИТМИ РОЗГАЛУЖЕННЯ В МОВІ Turbo-Pascal

1. Мета роботи

Вивчення випадків розгалуження програм.

2. Задачі роботи

Навчитися створювати програми з використанням оператора розгалуження.

3. Порядок виконання роботи

3.1. Створіть і налагодьте програму для розв'язування наступної задачі. Є дійсні числа a , b , c . Знайдіть найбільшу або найменшу з комбінацій даних чисел (за вказівкою викладача, згідно додатку).

3.2. Проведіть чисельні розрахунки для $a=1$, $b=2$, $c=3$;
 $a=-5$, $b=0$, $c=10$.

3.3. Доповніть програму наступним фрагментом. Визначте, чи належить обчислене значення максимуму (мінімуму) інтервалу $[0.5; 2.0]$ і виведіть на екран відповідне повідомлення. Налагодьте і виконайте програму з довільними значеннями вхідних даних.

4. Вимоги до звіту

Звіт про виконану роботу повинен включати:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи.
- номер варіанта для виконання завдання і умови свого варіанта;
- блок-схеми рішення задач;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п.7

5. Теоретичні положення

Умовний оператор або оператор розгалуження використовується, коли в алгоритмі розв'язування задачі передбачені альтернативні шляхи розв'язання, – з двох альтернатив вибирається одна, залежно від умови (умовою є логічний вираз або декілька логічних виразів рис.2.1).

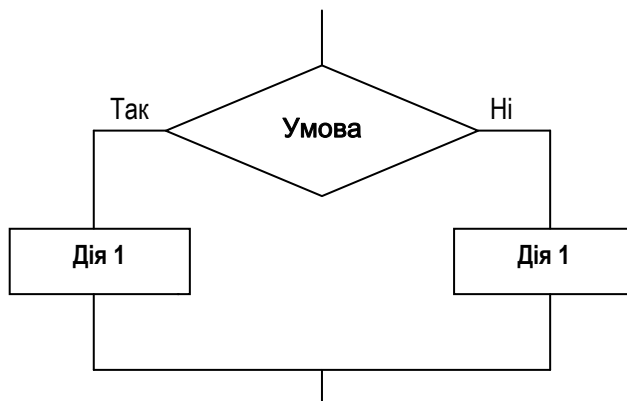


Рис. 2.1. Блок-схема алгоритму розгалуження

5.1. Логічні вирази

Логічні вирази можуть приймати одне з двох значень *TRUE* (істина) або *FALSE* (хибно). Найпростішими логічними виразами є вирази відношення: *A1 OB A2*

Тут *A1* і *A2* – вирази, а *OB* – операція відношення. Операції відношень в Turbo Pascal позначаються так: = (дорівнює), <> (не дорівнює), < (менше), > (більше), < = (менше або дорівнює), > = (більше або дорівнює).

Приклади логічних виразів: 3<5; 18>2; A=B.

В одному виразі може знадобитися перевірка декількох подібних умов. Умови можуть бути пов'язані за допомогою логічних операцій, з них найбільш використовувані – *AND* (І) і *OR* (АБО).

Вираз *A AND B* дає значення *TRUE* (істина), тільки в тому випадку, якщо *A* і *B* мають значення *TRUE*. В усіх інших випадках значення виразу *A AND B* – *FALSE* (хибно).

<TRUE> AND <TRUE> = <TRUE>

<TRUE> AND <FALSE> = <FALSE>

<FALSE> AND <FALSE> = <FALSE>

Наприклад, визначити чи потрапляє значення змінної X у інтервал від 0 до 10, можна за допомогою умови:

$(X \Rightarrow 0) \text{ and } (X \leq 10)$

Вираз $A \text{ OR } B$ дає значення *FALSE* (хибно), тільки в тому випадку, якщо A і B мають значення *FALSE*. В усіх інших випадках результат – *TRUE* (істина).

$\langle \text{TRUE} \rangle \text{ OR } \langle \text{TRUE} \rangle = \langle \text{TRUE} \rangle$

$\langle \text{TRUE} \rangle \text{ OR } \langle \text{FALSE} \rangle = \langle \text{TRUE} \rangle$

$\langle \text{FALSE} \rangle \text{ OR } \langle \text{FALSE} \rangle = \langle \text{FALSE} \rangle$

5.2. Складений оператор

Якщо при деякій умові треба виконати певну послідовність операторів, то їх поєднують в один складений оператор. Складений оператор починається ключовим словом *BEGIN* і закінчується словом *END*. Між цими словами знаходяться складові оператори, які виконуються у порядку їх проходження. Після *END* ставиться крапка з комою.

Приклад.

begin

i:=2;

k:=i/5;

end;

Слова *BEGIN* і *END* відіграють роль операторних дужок. Тіло самої програми також має вид складеного оператора. Після останнього *END* програми ставиться крапка.

5.3. Оператор IF

Загальний вид оператора *IF*:

IF A THEN ST1 ELSE ST2;

Тут *IF*, *THEN*, *ELSE* – зарезервовані слова (*якщо, то, інакше*); A – логічний вираз, $ST1$, $ST2$ – оператори (прості або складені).

Умовний оператор працює за наступним алгоритмом. Спочатку обчислюється умовний вираз A . Якщо результат *TRUE* (істина), то виконується оператор $ST1$, а оператор $ST2$ пропускається; якщо результат *FALSE* (хибно) – навпаки: оператор $ST1$ пропускається, а виконується оператор $ST2$. Потім в обох випадках управління передається до наступного оператора.

Наприклад:
var
x, y, z: integer;
begin

if x>z then

y:=z
else
y:=x;

При виконанні цього фрагмента змінна *Y* отримає значення змінної *X*, якщо це значення не перевищує *Z*. В протилежному випадку *Y* дорівнює *Z*.

Частина *ELSE ST2* може бути пропущена. Тоді при значенні *TRUE* умовного виразу виконується оператор *ST1*, в протилежному випадку цей оператор пропускається:

var
x, y, max: integer;
begin

if x>max then
 max:=x;
y:=x;

В цьому прикладі змінна *Y* завжди буде мати значення невідомої *X*, а в *MAX* запам'ятовується максимальне значення *X*. Таку форму оператора *IF* називають скороченою.

Приклад 1. Змінні *a*, *b* і *c* – цілі числа. Визначити найбільше з них.

Розглянемо три способи розв'язування цієї задачі.

1 спосіб.

.....
if a>b then
 if a>c then max:=a
 else max:=c
 else
 if b>c then max:=b
 else max:=c;

Якщо $a > b$, то max шукається серед a і c , в протилежному випадку - серед b і c (рис.2.2).

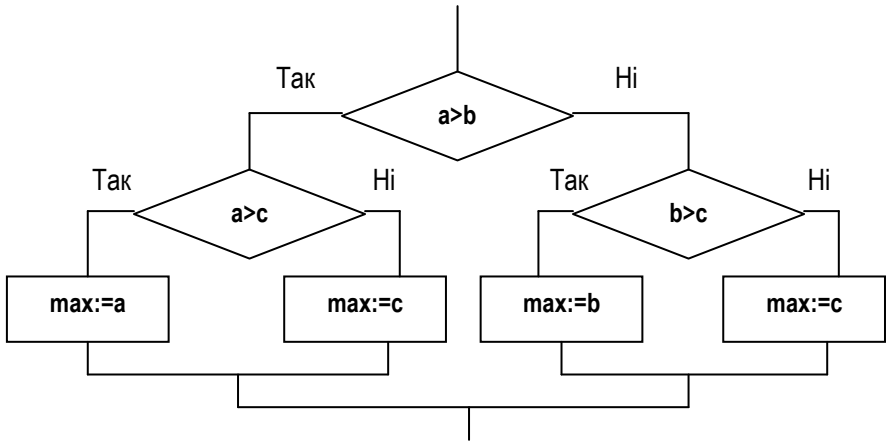


Рис. 2.2. Блок-схема алгоритму розв'язування задачі 1-м способом

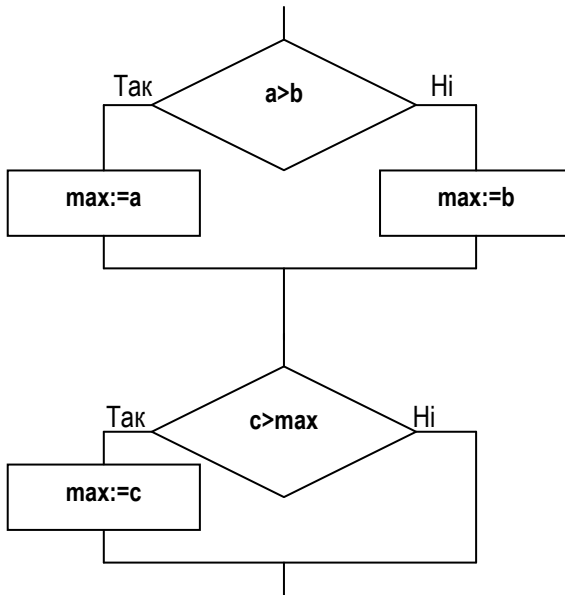


Рис. 2.3. Блок-схема алгоритму розв'язування задачі 2-м способом

2 спосіб.

```
.....  
if a>b then max:=a else max:=b;  
    then max:=c;  
.....
```

Попередньо за *max* приймається більше з *a* і *b*, але якщо виявиться, що *c* перевищує прийняту величину *max*, то остання замінюється на *c* (рис.2.3).

3 спосіб.

```
.....  
max:=a;  
if b>max then max:=b;  
if c>max then max:=c;  
.....
```

Можна за початкове максимальне значення *max* прийняти спочатку будь-яку з початкових величин (наприклад, першу). Потім, якщо серед інших величин будуть зустрічатися величини, які перевищують поточні значення *max*, замінити поточне значення *max* новим.

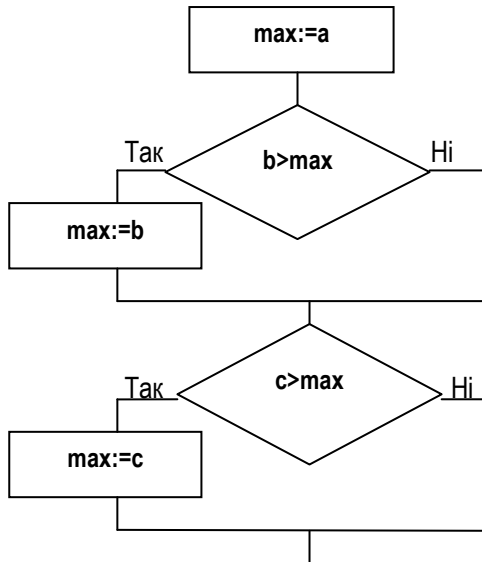


Рис. 2.4. Блок-схема алгоритму розв'язування задачі 3-м способом.

6. Методичні рекомендації

При складанні програми належить використовувати оператор *READLN* для введення значень змінних *a*, *b*, *c*; потрібно ввести й описати змінну для позначення шуканого найбільшого (чи найменшого) значення (наприклад, *max* чи *min*). При розв'язуванні задачі можна скористатися одним зі способів, описаних у прикладі 1 п. 5.3. При виконанні завдання п. 3.3 для перевірки умови використовувати оператор *IF* з логічним виразом, у якому використовується операція **AND**.

7. Контрольні запитання

- 7.1 Що розуміють під алгоритмом розгалуження?
- 7.2 Наведіть приклади випадків розгалуження.
- 7.3 Як позначається розгалуження в блок-схемах?
- 7.4 Які оператори розгалуження існують в мові Turbo Pascal?
- 7.5 Який формат має оператор *IF*?
- 7.6 Яка різниця між повною і скороченою формою оператора *IF*?

8. Література

1. Інформатика. Базовий курс / Симонович С.В. і ін.– СПб: Питер, 2000. – 640 с.
2. Семашко Г.Л., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація і програмування”, Чернігів, ЧДЕІУ – 2011.
5. Войтюшенко Н.М. Інформатика і комп'ютерна техніка: Навч. пос. з баз.підготовки для студ.екон. і техн.спеціальностей ден. і заочн. форм навчання. - К.: Центр навчальної літератури, 2006. – 568 с.
6. Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч.посіб. Вид. 2-ге, перероб., доп. - К.: Академвидав, 2007.- 416 с.(Альма-матер).
7. Інформатика і комп'ютерна техніка: Навчальний посібник / За ред. М.Є. Рогози. - К.: Видавничий центр «Академія», 2006. – 368 с.

Додаток

№ варіанту	Значення для пошуку	№ варіанту	Значення для пошуку
1	$\max (a+b+c, abc, a-b)$	16	$\min (a^2+b^2+c^2, a^2b^2c^2, a^2-b^2)$
2	$\min (a+2b, b+3c, c)$	17	$\max (a^2+2b^2, b^2+3c^2, c^2)$
3	$\min (2a-b, c^2, b+c)$	18	$\min (2a^2-b^2, c^2, b^2+c^2)$
4	$\max (a/c, b+c, 5c)$	19	$\min (a^2/c^2, b^2+c^2, 5c^2)$
5	$\min (3a, a+b+7c, 8c)$	20	$\max (3a^2, a^2+b^2+7c^2, 8c^2)$
6	$\max (5a+7b, b-c, 3c)$	21	$\min (5a^2+7b^2, b^2-c^2, 3c^2)$
7	$\min (a^2, a+2b-c, c)$	22	$\max (a^2, a^2+2b^2-c^2, c^2)$
8	$\max (a+b+c, 3a+8, a+7c)$	23	$\min (a^2+b^2+c^2, 3a^2+8, a^2+7c^2)$
9	$\min (ab, ac, bc)$	24	$\max (a^2b, ac^2, bc^2)$
10	$\max (a, a^2-b, ac)$	25	$\min (a^2, a^2-b^2, ac^2)$
11	$\min (a+7, b-4, 3c)$	26	$\max (a^2+7, b^3-4, 3c^2)$
12	$\max (ab, b+5, ac)$	27	$\min (ab^5, b^5+5, ac^3)$
13	$\min (a+2b, b-c, a+c)$	28	$\max (a^2+2b^2, b^2-c^2, a^2+c^2)$
14	$\max (4a, a^2+c, b+c)$	29	$\min (4a^3, a^2+c^2, b^2+c^2)$
15	$\min (a-b, a-c, b-c)$	30	$\max (a^2-b^2, a^2-c^2, b^2-c^2)$

Лабораторна робота № 3

ОПЕРАТОРИ МОВИ TURBO PASCAL

1. Мета роботи

Вивчити оператори циклу мови PASCAL.

2. Завдання роботи

Навчитися працювати з алгоритмами циклічної структури на мові PASCAL.

3. Порядок виконання роботи

3.1. З використанням оператора циклу *For* скласти і налагодити програму для розв'язання наступної задачі.

Дано дійсне число x . Обчислити суму залежного від x ряду у відповідності з зазначеним викладачем варіантом додатка А.

3.2. Провести обчислення для $x=1$, $x=1.5$, $x=2$.

3.3. Розв'язати поставлену у пункті 3.1 задачу з використанням циклу *While* (для непарних номерів варіантів) або з використанням оператора циклу *Repeat* (для парних номерів варіантів). Провести чисельні обчислення і порівняти отримані результати з результатами пункту 3.2.

4. Вимоги до звіту

Звіт про виконану роботу повинен містити:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи;
- номер варіанта для виконання завдання і умови свого варіанта;
- блок-схеми рішення задач;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п.7

5. Загальні положення

5.1. Алгоритми циклічної структури

Відповідно до алгоритмічної мови циклом можна назвати повторення послідовності дій. При цьому дані повторюваних дій складають так зване тіло циклу. Умова припинення повторень називається умовою кінця циклу.

З точки зору програмування, цикл – повторювані фрагменти програм. В мові Pascal є три різних оператора, які слугують для програмування циклів:

- цикл з параметром *For...to...Do*;
- цикл з попередньою перевіркою умови припинення повторень *While...Do*;
- цикл з постумовою припинення повторень *Repeat...Until*.

5.2. Оператор циклу з параметром

Блок-схема, що відповідає циклу з параметром, представлена на рис. 3.1.

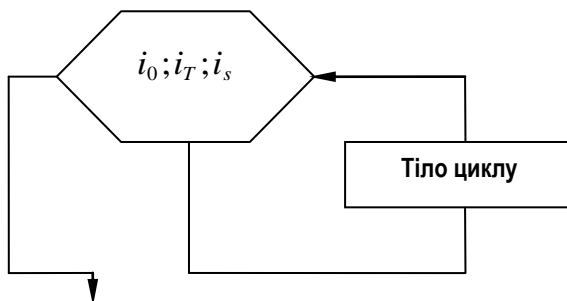


Рис. 3.1. Блок-схема циклу з параметром

Тут i – параметр циклу; i_0 – початкове значення параметра; i_T – кінцеве значення параметра; i_s – крок приросту параметру (у випадку мови Pascal $+1$ або -1).

Циклічна гілка блок-схеми виконується для усіх i , починаючи з i_0 і закінчуючи i_T з кроком i_s (тобто для $i_0, i_0+i_s, i_0+2i_s, i_0+3i_s, \dots$). Як тільки i вийде за межі інтервалу $[i_0, i_T]$, повторення перериваються і програма виходить з циклу.

Оператор циклу з параметром на мові Pascal має наступний формат:

For i:=i₀ to i_T do <оператор>

Тут *For*, *to*, *do* – зарезервовані слова («для», «до», «виконати»);

i (параметр циклу) – змінна типу *INTEGER* (точніше, в загальному випадку, будь-якого порядкового типу);

i_0 (початкове значення) і i_T (кінцеве значення параметра) – константи або вирази того ж типу;

<оператор> – довільний оператор мови Pascal (тіло циклу). В якості оператора може виступати розширений оператор з операторними дужками *begin...end*.

Крок приросту параметру при цьому постійний і дорівнює +1.

При виконанні оператора *For* спочатку обчислюється вираз i_0 і виконується присвоювання $i:=i_0$. Після цього циклічно повторюється:

– перевірка умови $i \leq i_T$; якщо умова не виконана, то оператор *For* припиняє свою роботу;

– виконання тіла циклу *<оператор>*;

– нарощування змінної циклу на одиницю.

Якщо умова не виконується на початку роботи оператора *For*, то виконуючий оператор не буде виконаний жодного разу.

Існує й інша форма запису оператора *For*, при якій крок збільшення параметра приймає значення -1. У цьому випадку оператор *For* буде мати наступний вид:

For i:=i₀ downto i_T do <оператор>

Приклад. Обчислити функцію $f=x+e^{-x}$ при $x=0, 1, 2, \dots, 10$ і роздрукувати отримані результати в табличному вигляді.

Можливий текст програми:

```
program calc_f;
var f: real;
    x: integer;
begin
    for x:=0 to 10 do
        begin
            f:=x+exp(-x);
            writeln('x=', x, ' f=', f);
        end;
end.
```

5.3. Цикл з передумовою

Оператор циклу з передумовою *While...Do* схематично представлений у вигляді блок-схеми на рис.3.2 і має наступний формат:

While <умова> *Do* <оператор>

де: *While*, *Do* – зарезервовані слова («Поки [виконується умова]», «робити»);

<умова> – вираз логічного типу;

<оператор> – довільний оператор, який складає тіло циклу (включаючи випадок розширеного оператора).

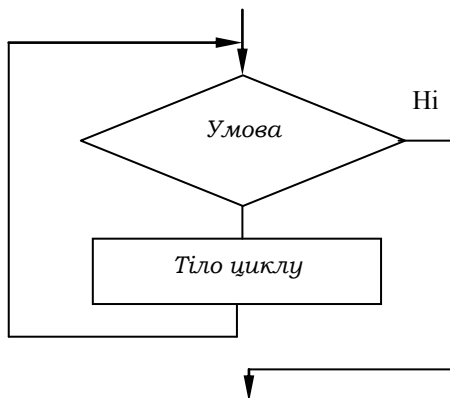


Рис.3.2. Блок-схема циклу з передумовою

Якщо умова має значення „Істина” (*True*), то виконується <оператор>, після чого перевірка умови повторюється. Якщо умова має значення „Хибно” (*False*), то оператор *While* припиняє свою роботу. Даний оператор може бути використаний у тих випадках, коли заздалегідь невідомо майбутнє число повторень.

Приклад. Є нескінченний ряд виду

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$$

Знайти суму даного нескінченного ряду з заданою точністю $e=0.01$, тобто продовжувати процес підсумовування до тих пір, поки модуль різниці між сумами для k -го и $(k+1)$ -го кроку не стане менше e .

```
program Summa;  
var k: integer;  
    s,ss,eps,real;  
begin  
    eps:=0.01;
```

```

s:=1;      { значення суми на 1 кроці циклу}
ss:=0;    { початкове значення допоміжної змінної для
зберігання проміжних сум}
k:=1;     { номер кроку циклу }
while abs(s-ss)>=eps do
  begin
    ss:=s; {зберігання значення суми на поточному
кроці циклу}
    k:=k+1; {перехід до наступного кроку циклу}
    s:=s+1/k; {обрахування суми на наступному кроці}
  end;
writeln;  { 'Сума ряду зі загаданюю точністю=', s }
end.

```

5.4. Цикл з постумовою

Оператор циклу з постумовою має наступний формат:

Repeat <тіло циклу> *Until* <умова>

Тут *Repeat*, *Until* – зарезервовані слова («Повторювати», «доки не [виконається умова]»);

<тіло циклу> – довільна послідовність операторів;

<умова> – вираження логічного типу.

Оператори тіла циклу завжди виконуються хоча б один раз, після чого обчислюється значення <умова>: якщо його значення хибно (*FALSE*), оператори тіла циклу повторюються, в протилежному випадку оператор *Repeat* завершує свою роботу.

Блок-схема даного оператора представлена на рис. 3.3.

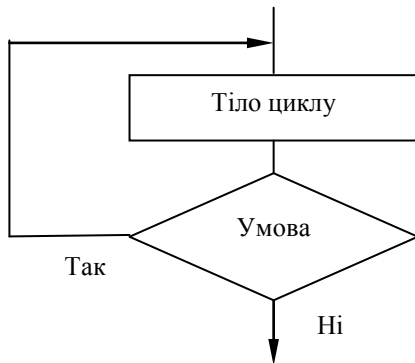


Рис. 3.3. Блок-схема циклу з постумовою

Приклад. Є натуральне число $N > 10$. Знайти першу цифру даного числа.

```
program first;
  var n: longint;
      begin
        writeln ('введіть  $N > 10$ , n);
        readln (n);
        repeat
          n:= n div 10; {відсікається остання цифра числа n,
іншими словами n пере присвоюється значення цілої частини від
ділення n на 10 }
        until n<10;{ повторювати доки n, не стане менше
10, тобто в ньому не залишиться тільки одна цифра }
        writeln (n);
      end.
```

6. Методичні вказівки до виконання роботи

При розв'язанні стандартної задачі на знаходження суми ряду передусім потрібно виявити закономірність побудови ряду, тобто залежність виду доданка від його порядкового номера.

Наприклад, у випадку суми ряду вигляду $\cos(x)+\cos(2x)+\cos(3x)+\dots+\cos(7x)$ загальний вид доданка з номером k буде мати вид $\cos(k*x)$, де k змінюється від 1 до 7.

Накопичення суми при цьому повинне проводитися за кроками, на кожному кроці циклу до наявного вже значення суми повинний додаватися черговий доданок (спочатку передбачається, що значення суми дорівнює 0).

Фрагмент програми знаходження зазначеної вище суми ряду за допомогою оператора *FOR* може мати, наприклад, наступний вигляд:

```
...
s:=0;
for k:=1 to 7 do s:=s+cos(k*x);
writeln('сума =', s);
end.
```

За допомогою оператора *While*:

```
...
s:=0;
k:=1;
while k<=7 do
begin
s:=s+cos(k*x);
```

```
k:=k+1;  
end;  
writeln('сумма =', s);  
end.
```

За допомогою оператора *Repeat*:

```
s:=0;  
k:=1;  
repeat  
s:=s+cos(k*x);  
k:=k+1;  
until k>7;  
writeln('сума =', s);  
end.
```

7. Контрольні запитання

- 7.1 Що називається циклом, тілом циклу?
- 7.2 Які оператори циклу мови *Pascal* Ви знаєте?
- 7.3 Наведіть відомі Вам формати оператора *For*.
- 7.4 З яким кроком може змінюватися параметр оператора *For*?
- 7.5 Як можна розрахувати число кроків в операторі *For*?
- 7.6 Поясніть відмінності в операторах циклу з передумовою і постумовою.
- 7.7 Який формат має оператор *While*?
- 7.8 Наведіть приклад, коли тіло циклу в операторі *While* не виконується жодного разу.
- 7.9 Приведіть формат оператора *Repeat*.
- 7.10 Поясніть, в якому випадку припиняються повторення в операторі *Repeat*.

8. Література

1. Інформатика. Базовий курс / Симонович С.В. і ін.– СПб: Питер, 2000. – 640 с.
2. Семашко Г.Л., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація і програмування”, Чернігів, ЧДЕІУ – 2011.

Додаток А

Варіанти завдань для виконання

$$1) S = x + \frac{x}{4} + \frac{x}{9} + \frac{x}{16} + \frac{x}{25} + \dots + \frac{x}{100}$$

$$2) S = \sin x + \sin 2x + \sin 3x + \dots + \sin 7x$$

$$3) S = \cos x + 2 \cos 2x + 3 \cos 3x + \dots + 6 \cos 6x$$

$$4) S = \frac{e^{-x}}{2} + \frac{e^{-2x}}{3} + \frac{e^{-3x}}{4} + \dots + \frac{e^{-8x}}{9}$$

$$5) S = \sin 3x + \sin 4x + \sin 5x + \dots + \sin 11x$$

$$6) S = \cos 2x + \cos 3x + \cos 4x + \dots + \cos 10x$$

$$7) S = 2x + \frac{3x}{2} + \frac{4x}{3} + \frac{5x}{4} + \dots + \frac{11x}{10}$$

$$8) S = \frac{\sin x}{2} + \frac{\sin 2x}{3} + \frac{\sin 3x}{4} + \dots + \frac{\sin 9x}{10}$$

$$9) S = \cos x + \cos 2x + \cos 3x + \dots + \cos 13x$$

$$10) S = \sin x + \frac{\sin 2x}{4} + \frac{\sin 3x}{9} + \frac{\sin 4x}{16} + \dots + \frac{\sin 9x}{81}$$

$$11) S = \frac{\cos x}{2} + \frac{\cos 2x}{4} + \frac{\cos 2x}{6} + \dots + \frac{\cos 8x}{16}$$

$$12) S = 2 \cos x + 3 \cos 2x + 4 \cos 3x + \dots + 8 \cos 7x$$

$$13) S = \frac{\sin x}{3} + \frac{\sin 2x}{6} + \frac{\sin 3x}{9} + \dots + \frac{\sin 7x}{21}$$

$$14) S = \frac{\sin x}{3} + \frac{\sin 2x}{4} + \frac{\sin 3x}{5} + \dots + \frac{\sin 9x}{11}$$

$$15) S = 2 \cos 2x + 3 \cos 3x + \dots + 15 \cos 15x$$

$$16) S = 3 \sin x + 4 \sin 2x + 5 \sin 3x + \dots + 9 \sin 7x$$

$$17) S = \frac{\sin x}{2} + \frac{\sin 2x}{4} + \frac{\sin 3x}{6} + \dots + \frac{\sin 7x}{14}$$

$$18) S = x \sin x + 2x \sin 2x + 3x \sin 3x + \dots + 15x \sin 15x$$

$$19) S = 1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \dots + \frac{1}{x^5}$$

$$20) S = \frac{1}{x} + \frac{2}{x^2} + \frac{3}{x^3} + \dots + \frac{6}{x^6}$$

$$21) S = x \cos x + 2x \cos 2x + 3x \cos 3x + \dots + 15x \cos 15x$$

$$22) S = x \operatorname{tg} x + 2x \operatorname{tg} 2x + 3x \operatorname{tg} 3x + \dots + 15x \operatorname{tg} 15x$$

$$23) S = x \operatorname{arctg} x + 2x \operatorname{arctg} 2x + 3x \operatorname{arctg} 3x + \dots + 15x \operatorname{arctg} 15x$$

$$24) S = x \ln x + 2x \ln 2x + 3x \ln 3x + \dots + 15x \ln 15x$$

$$25) S = 2 \sin 2x + 3 \sin 3x + \dots + 15 \sin 15x$$

$$26) S = \sin 3x + \cos 4x + \sin 5x + \dots + \cos 11x$$

$$27) S = \log_2 x + \log_2 2x + \log_2 3x + \dots + \log_2 7x$$

$$28) S = \lg x + \lg 2x + \lg 3x + \dots + \lg 7x$$

$$29) S = \operatorname{tg} x + \operatorname{tg} 2x + \operatorname{tg} 3x + \dots + \operatorname{tg} 7x$$

$$30) S = \operatorname{arctg} x + \operatorname{arctg} 2x + \operatorname{arctg} 3x + \dots + \operatorname{arctg} 7x$$

Лабораторна робота № 4

АЛГОРИТМИ ЦИКЛІЧНОЇ СТРУКТУРИ

1. Мета заняття

Вивчення основ побудови алгоритмів і програм циклічної структури.

2. Задачі заняття

Освоєння основних прийомів і набуття навичок у складанні алгоритмів і програм циклічної структури.

3. Зміст заняття

3.1. Ознайомитися з основними елементами блок-схем алгоритмів і програм циклічної структури.

3.2. Оволодіти прийоми побудови блок-схем алгоритмів циклічної структури.

3.3. Розібрати наведені в методичних вказівках приклади.

3.4. Виконати індивідуальні завдання.

4. Вимоги до звіту

Звіт про виконану роботу повинен містити:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи;
- номер варіанта для виконання завдання і умови свого варіанта;
- блок-схеми розв'язування задач;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п. 6.

5. Основні положення

Командою повторення або циклом називається така форма організації дій, при якій одна й та ж послідовність дій повторюється до тих пір, поки зберігається значення певного логічного виразу. При зміні значення логічного виразу на протилежне, повторення припиняються (цикл завершується).

Для організації циклу необхідно виконати наступні дії:

- перед початком циклу задати початкове значення параметра;
- усередині циклу змінювати параметр циклу з допомогою оператора присвоювання;
- перевіряти умову повторення або закінчення циклу;
- керувати циклом, тобто переходити до його початку, якщо він не закінчений, або виходити з циклу в протилежному випадку.

Розрізняють цикли з відомим числом повторень (*цикли з параметром*) та *ітераційні* (з перед- і постумовою).

У циклі з відомим числом повторень параметр змінюється у заданому діапазоні.

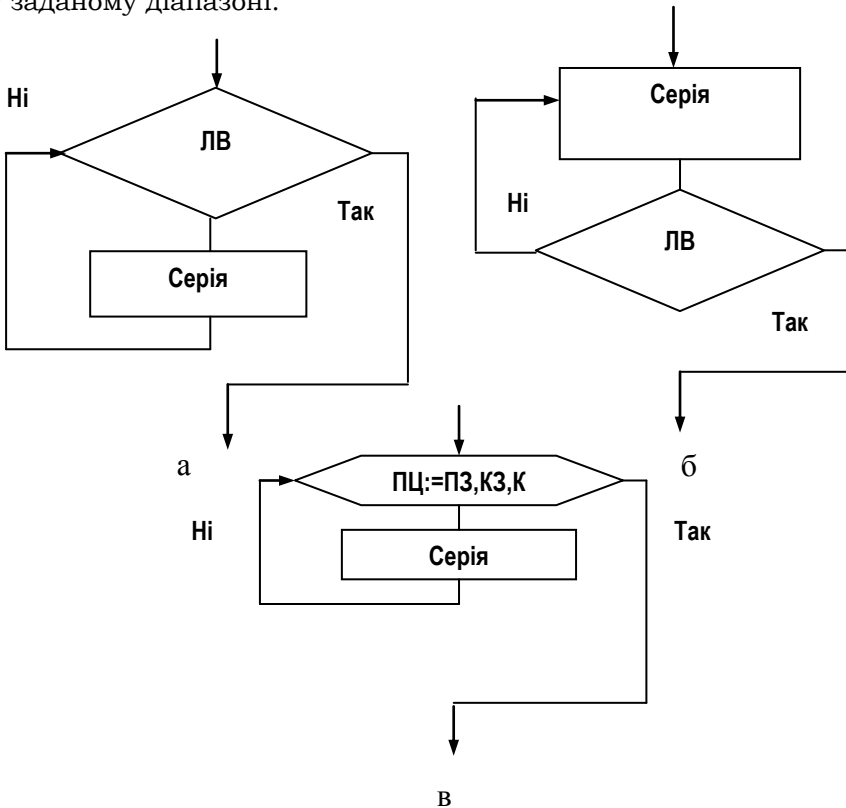


Рис. 4.1. Цикли:

- а) – цикл з передумовою (цикл *ПОКИ*), б)– цикл з постумовою (цикл *ДО*), в) – цикл з параметром

Якщо у циклі змінюється проста змінна, то вона є параметром циклу; якщо у циклі змінюється змінна з індексом, то індекс цієї змінної є параметром циклу.

Графічні позначення (позначення на блок-схемах) алгоритмів циклічних структур наведені на рис. 4.1.

На представлених блок-схемах (рис. 4.1), блок *СЕРІЯ* позначає один або декілька будь-яких операторів; *УМОВА* є логічним виразом (*ЛВ*) (якщо його значення *ІСТИНА* – перехід відбувається по гілці *ТАК*, інакше – за гілкою *Ні*). На схемі циклу з параметром використані позначення: *ПЦ* – параметр циклу, *ПЗ* – початкове значення параметра циклу, *КЗ* – кінцеве значення параметра циклу, *К* – крок зміни параметра циклу.

Початок і кінець алгоритму на блок-схемах позначають овалом, змінні, які вводяться і виводяться записуються в паралелограмі.

В подальшому будемо використовувати запис алгоритмів за допомогою блок-схем і словесного опису.

Приклад 1. Підрахувати кількість непарних цифр в записі натурального числа n .

Розв'язання. З заданого числа вибирати з молодшого розряду цифру за цифрою до тих пір, поки вони не вичерпаються, тобто стане дорівнювати нулю. Кожну непарну цифру враховувати.

1. Ввести число n

2. $K := 0$ {підготовка лічильника}

3. Якщо $n = 0$ – перехід до п. 7

4. Якщо $n \bmod 10 \bmod 2 = 1$, то $K := K + 1$

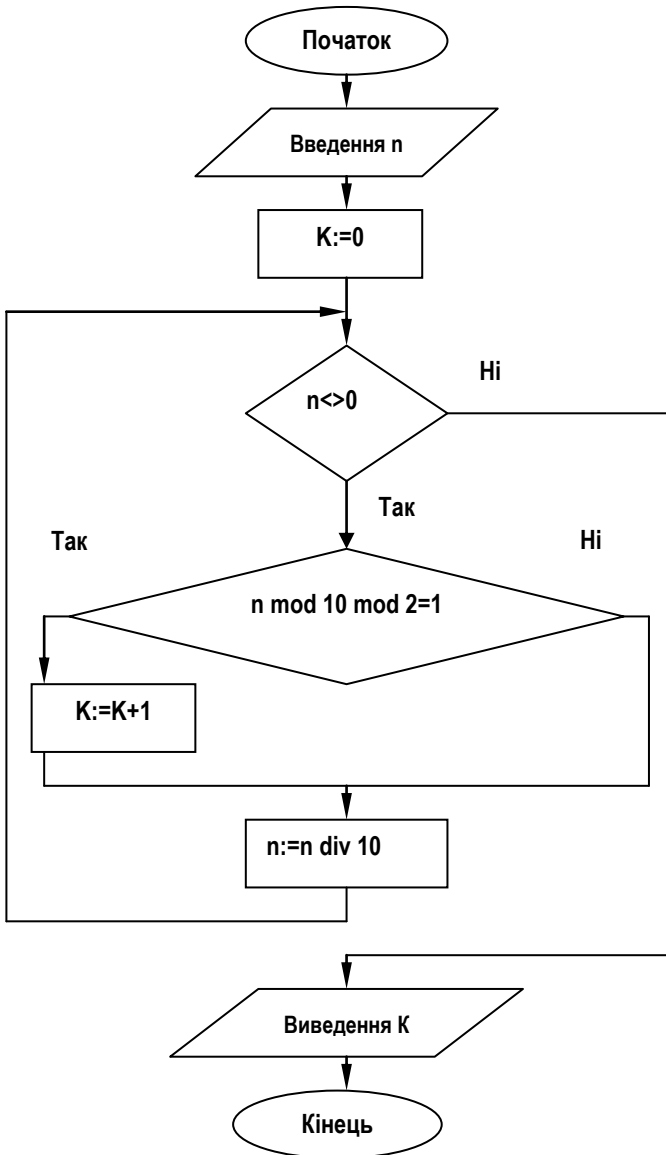
5. $n := n \operatorname{div} 10$

6. Перехід до п 3

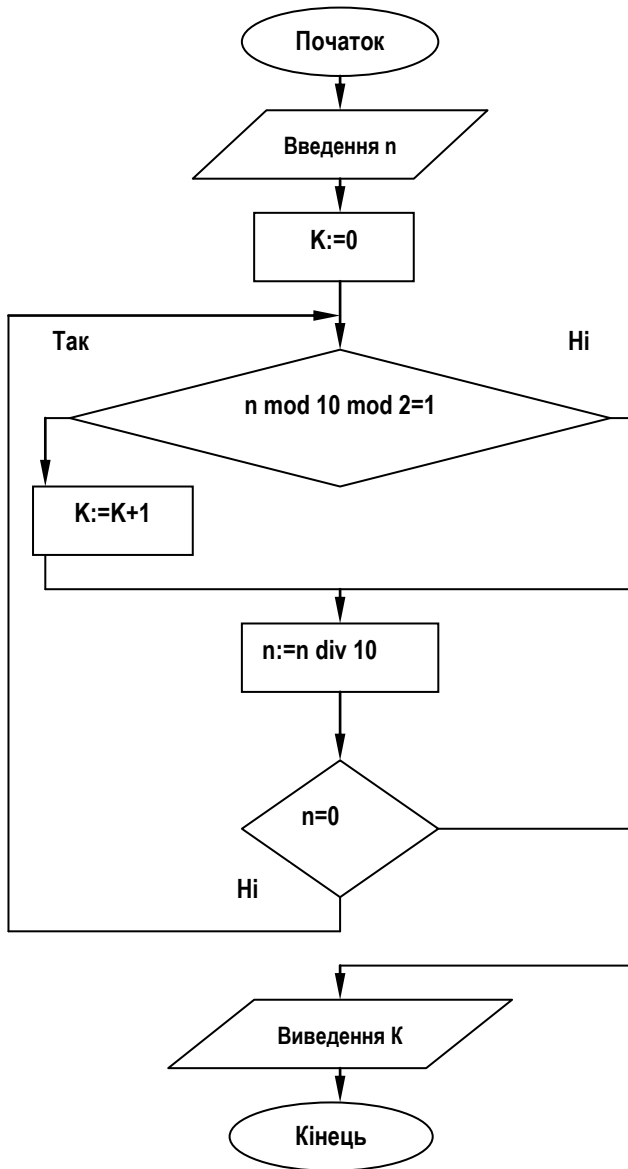
7. Виведення K

8. Кінець

Задача розв'язана двома способами:



Перший спосіб розв'язання виконаний з використанням циклу з передумовою



Другий спосіб розв'язання виконаний з постумовою.

Приклад 2. Дана послідовність, загальний член якої

визначається формулою: $a = \frac{n-1}{n^2}$

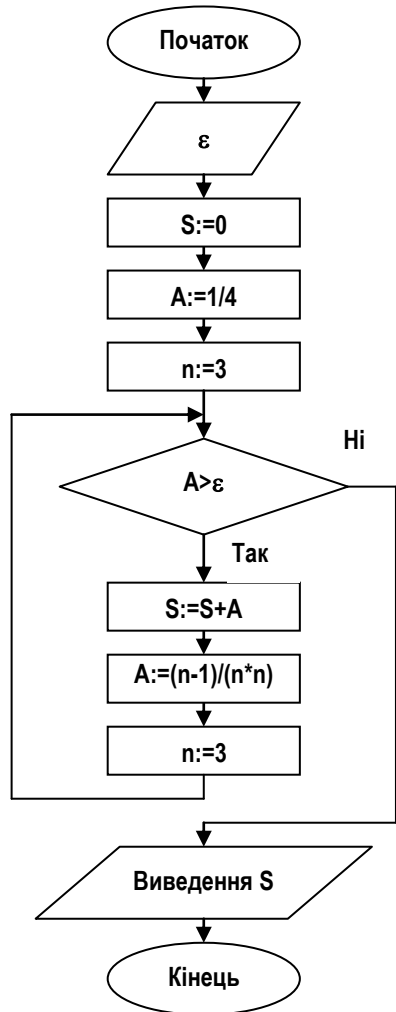
Обчислити при $n > 2$ суму тих її членів, які більше заданого числа ϵ .

При розв'язанні задачі знаходиться послідовно черговий член i , якщо він більше ϵ , додається до суми.

В розглянутих вище прикладах кількість повторень заздалегідь невідома.

В першому прикладі вона залежить від кількості цифр у запису натурального числа, у другому - від числа ϵ .

У тих же випадках, коли кількість кроків відома з умови задачі, краще використовувати цикл з параметром.

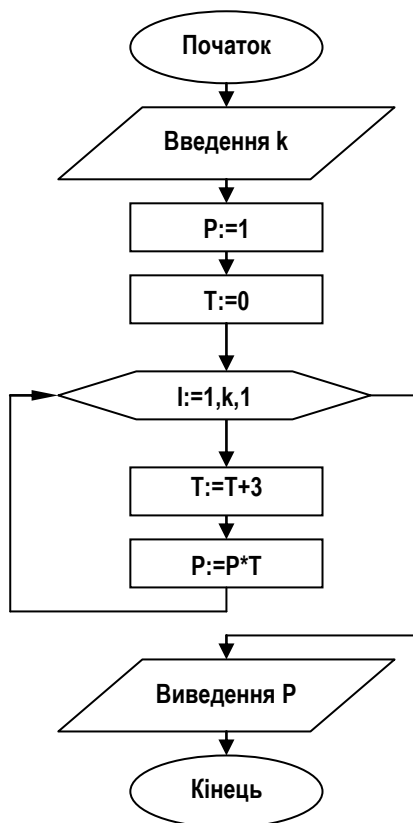


Приклад 3. Знайти добуток перших k натуральних чисел, кратних 3.

При складанні алгоритму врахуємо, що перше натуральне число, кратне 3, є трійка, а усі наступні більше попереднього на 3.

Для перевірки працездатності алгоритму необхідно задати значення вхідних змінних, обчислити кінцевий результат за алгоритмом і порівняти з результатом ручного розрахунку.

1. Введення k
2. $P := 1$ {тут накопичуємо добуток}
3. $T := 0$ {тут будуть числа, кратні 3}
4. $I := 1$
5. Якщо $I > k$, перехід до п. 10
6. $T := T + 3$
7. $P := P * T$
8. $I := I + 1$
9. Перейти до п. 5
10. Вивід P
11. Кінець



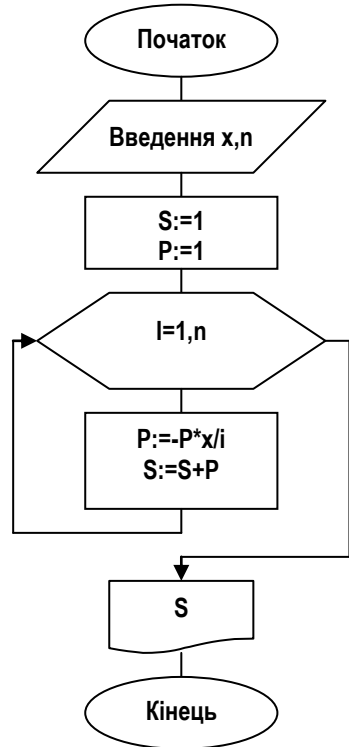
Приклад 4. Для заданого x і n обчислити

$$S = 1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^n}{n!}$$

Тут $n! = 1 \cdot 2 \cdot 3 \dots n$ (читається як "n-факторіал").

1. Ввести ε
2. $S := 0$
3. $A := 1/4$
4. $n := 3$
5. Порівняти A з ε . Якщо $A \geq \varepsilon$,
перехід до п. 10
6. $S := S + A$
7. $A := (n-1)/(n*n)$
8. $n := n + 1$
9. Перехід до п. 5
10. Виведення S
11. Кінець

Дані		Результат
$X=1$	$n=3$	$S = 1 - \frac{1}{1} + \frac{1^2}{1 \cdot 2} - \frac{1^3}{1 \cdot 2 \cdot 3} = 0.33$



Turbo Pascal

Program SumUp;

Uses Crt;

Var x, S, P : Real;

{P – черговий доданок}

i, n : Integer;

BEGIN ClrScr;

Write('Введіть n = '); ReadLn(n);

Write(' Введіть x = '); ReadLn(x); WriteLn;

S := 1; P := 1;

For i := 1 to n do

begin

*P := - P*x / i; {одержання чергового доданка}*

S := S + P

end;

WriteLn(Відповідь : S = ', S : 7 : 3); ReadLn

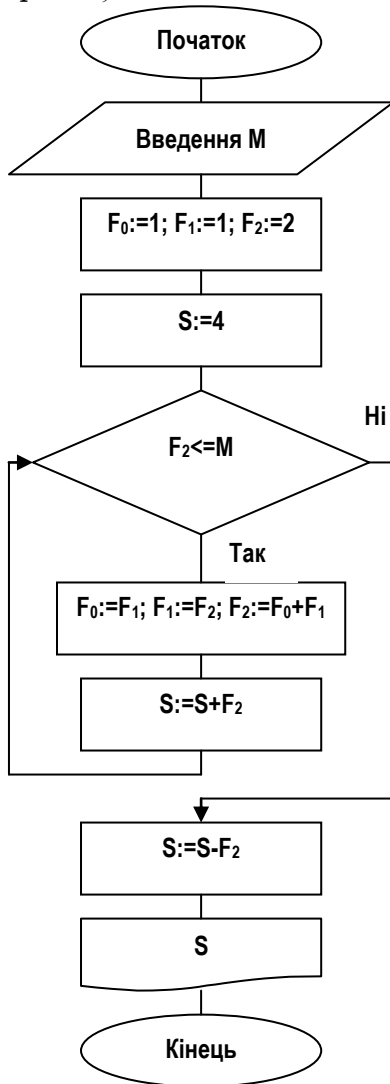
END.

Приклад 5. Фрагмент програми простого парольного захисту:
 For I:=1 to 3 do begin {дається три спроби введення паролю}
 Write ('Введіть ПАРОЛЬ');
 ReadLn(S); {змінні S і Parol – одного типу}
 IF S=Parol
 THEN BREAK {переривання циклу захисту}
 ELSE IF I<3 THEN CONTINUE; {даємо ще спроби}
 WriteLn('Не знаєте Пароль!'); {спрацює при I=3}
 ReadLn;
 HALT {переривання програми}
 End;
 { Ділянка програми що захищається }

Приклад 6. Числа Фібоначчі (F_i) обчислюються за формулами $F_0 = F_1 = 1$; $F_i = F_{i-1} + F_{i-2}$ при $i = 2, 3, \dots$ (кожне наступне число дорівнює сумі двох попередніх). Обчислити суму усіх чисел Фібоначчі, які не перевершують задане натуральне число M .

Тест		
Номер тесту	Дані	Результат
1	M=10	S=1+1+2+3+5+8=20
2	M=1	S=1+1=2

Виконання алгоритму				
F_0	F_1	F_2	S	$F_2 < M$
1	1	2	4	+
1	2	3	4+3=7	+
2	3	5	7+5=12	+
3	5	8	12+8=20	+
5	8	13	20+13=33	-(кц)
			33-13=20	




```

Turbo Pascal
Program SummaFib;
Uses Crt;
Var M,           {задане число }
    F0, F1, F2,  {три послідовних числа Фібоначчі}
    S : Integer; {сума чисел Фібоначчі}
BEGIN
  ClrScr;
  Write(увведіть натуральне M : ');
  ReadLn(M);
  F0:=1; F1:=1; F2:=2;
  S:=4;          {4 – сума перших трьох чисел Фібоначчі}
  Write('Числа Фібоначчі, що не перевершують ', M, ' : ', F0:4, F1:4);
  While F2<=M do
    begin
      F0:=F1; F1:=F2; Write(F1 : 4);
      F2:=F0+F1; S:=S+F2;
    end;
  S:=S-F2; {вирахування з суми останнього числа, яке перевершує
M}
  WriteLn; WriteLn;
  WriteLn(Відповідь : Сума цих чисел дорівнює ', S); ReadLn
END.

```

Результати роботи Pascal-програми

Введіть натуральне M>0 : 10 <Enter>
Числа Фібоначчі, що не перевищують 10 : 1 1
2 3 5 8
Відповідь : Сума цих чисел дорівнює 20

Приклад 7. Визначимо суму ряду $S = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \dots + \frac{1}{(2N-1)^2}$, поки

черговий член ряду не стане менше 10^{-8} .

```
Var S,a,eps : real;
```

```
N : word;
```

```
Begin
```

```
  Eps:=1e-8; {точність обчислень}
```

```
  S:=0; N:=0; {початкові значення параметрів}
```

```
REPEAT
```

```
  a:=1/(2*N+1)/(2*N+1); {поточний член ряду}
```

```
  inc(N); {лічильник членів ряду}
```

```
  S:=S+a; {часткова сума ряду}
```

```

UNTIL a<eps;
WriteLn('S=',S:10:8);
ReadLn;
END.

```

Приклад 8. Запобігання невірному введенню даних. Модернізуємо програму визначення площі трикутника за трьома сторонами так, щоб при невірному введенні даних відбувалося не просто переривання програми, а виникав запит на їх корегування (з повтором до виконання потрібних умов).

```

Var a,b,c,p,S: real;
Begin
REPEAT {перевірка коректності введення даних;}
  REPEAT
    Write('Введіть сторону a='); ReadLn(a);
    IF a<=0 THEN WriteLn('повинно бути a>0!') Until a>0;
    REPEAT
      Write('Введіть сторону b='); ReadLn(b);
      IF b<=0 THEN WriteLn('повинно бути b>0!') Until b>0;
    REPEAT
      Write('Введіть сторону c='); ReadLn(c);
      IF c<=0 THEN WriteLn('повинно бути c>0!') Until c>0;
    IF (a+b<=c) and (a+c<=b) and (c+b<=a)
    THEN WriteLn('сума двох сторін повинна бути більше третьої! ')
    UNTIL (a+b>c) and (a+c>b) and (c+b>a);
    {Позрахунок по перевіреним даним;}
    p:=(a+b+c)/2;
    S:=sqrt(p*(p-a)*(p-b)*(p-c));
    WriteLn('Площа:', S:15:5);
  ReadLn
END.

```

Цикл з передумовою у подібних випадках використовувати менш зручно, тому що він може не виконатися жодного разу, якщо змінні, яким задаються значення з клавіатури, вже мають значення, що задовольняють умовам „захисту”.

Приклад 9. Розрахувати період (в місяцях) росту банківського внеску в m разів при відомій щомісячній відсотковій ставці pr (1% відповідає $pr=1$).

```
Var pr,m,s,s0: double;  
    N: word;  
Begin  
Repeat  
Write ('Введіть відсотки по внеску:'); redLn(pr);  
Write ('Введіть коефіцієнт зросту внеску:'); redLn(m)  
Until (m>1) and (pr>0);  
N:=0; {лічильник місяців}  
s0:=1000; {початкова сума внеску – довільно}  
s:=s0; {поточна сума внеску}  
    Repeat {перевір по місяцям}  
    s:=s*(1+pr/100);  
    inc(N)  
    Until s/s0>=m;  
WriteLn ('кількість місяців внеску:',N);  
ReadLn  
End.
```

6. Задачі для самостійного розв'язання

6.1. Є дійсні числа x , a , натуральне число $n=50$.

Обчислити $((\dots ((x+a)^2+\dots a)^2+a)^2+a$ n дужок.

6.2. Є дійсне число $x=100$. Обчислити: $\frac{(x-2)(x-4)(x-8)\dots(x-64)}{(x-1)(x-3)(x-7)\dots(x-63)}$

6.3. Є натуральне $n=150$, дійсне x . Обчислити: $x^{n^2} / 2^n$

6.4. Є натуральне $n=125$, дійсне x . Обчислити: $x^{n^3} / 3^n$

6.5. Є натуральне $n=300$, дійсне x . Обчислити: $\sum_{i=1}^n \frac{x^i}{i!}$

6.6. Обчислити: $\sum_{i=1}^{100} \frac{1}{\sqrt[2]{i}}$

6.7. Обчислити: $\sum_{i=1}^{100} \frac{1}{i!}$

6.8. Обчислити: $\sum_{i=1}^{50} \frac{1}{i^3}$

6.9. Обчислити: $\sum_{i=1}^{200} \frac{1}{(2i)^2}$

6.10. Є натуральне $n=100$. Обчислити: $\sum_{k=1}^n \frac{1}{k}$

6.11. Є натуральне $n=100$. Обчислити: $\sum_{k=1}^n \frac{1}{k^5}$

6.12. Є натуральне n . Обчислити: $\sum_{k=1}^n \frac{1}{(2k+1)^2}$

6.13. Дано натуральне n . Обчислити: $\sum_{k=1}^n \frac{(-1)^k}{(2k+1)k}$

6.14. Є натуральне n . Обчислити: $\sum_{k=1}^n \frac{(-1)^k (k+1)}{k!}$

6.15. Є натуральне n . Обчислити: $\sum_{k=0}^n \frac{(-1)^k (k-1)}{k!}$

6.16. Є натуральне n . Обчислити: $\sum_{k=1}^n \frac{k!}{\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{k+1}}$

6.17. Є натуральне n , дійсне число x . Обчислити: $\sum (\frac{1}{n!} + \sqrt{|x|})$

6.18. Є натуральне n , дійсне число x . Обчислити: $\sum_{i=0}^n \frac{x + \cos(ix)}{2^i}$

6.19. Є натуральне n , дійсне число x . Обчислити: $\sum_{i=1}^n \frac{x}{2^i}$

6.20. Є натуральне n , дійсне число x . Обчислити: $\sin x + \sin x^2 + \dots + \sin x^n$

6.21. Є натуральне n , дійсне число x .
Обчислити: $\cos x + \cos x^2 + \dots + \cos x^n$

6.22. Обчислити суму $Z = 1 + 2 + 3 + \dots$. Обчислення припинити, коли значення Z перевищать задане значення A .

6.23. Відомий початковий вклад клієнта в банк і відсоток річного доходу. Визначити, через скільки років внесок перевищить заданий розмір і який при цьому буде розмір внеску.

6.24. Торгова фірма в перший день роботи реалізувала товарів на P тис. грн., а потім щодня збільшувала виторг на 3%. Яким буде виторг фірми в той день, коли вона вперше перевищить задане значення Q ? Скільки днів прийдеться торгувати фірмі для досягнення цього результату?

6.25. Мале підприємство в перший день роботи випустило P одиниць товарної продукції. Кожний наступний день воно випускало продукції на Q одиниць більше, чим в попередній. Скільки днів буде потрібно підприємству, щоб загальна кількість випущеної продукції за весь час роботи вперше перевищило запланований обсяг?

6.26. Розрахувати суму нескінченного ряду. Підсумовування проводити, поки черговий член ряду по модулю не стане менше заданої точності ε . Результат порівняти з точним значенням S_T , а похибку зіставити з величиною ε .

Таблиця 4.1.

№	Вид суми	N	Вид ряду	S _г	ε
6.26.1	$\frac{\pi^2}{6}$	2	$1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$	$\frac{\pi^2}{6}$	10^{-4}
6.26.2		4	$1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots$	$\frac{\pi^4}{90}$	10^{-6}
6.26.3	$\sum_{i=1}^{\infty} (-1)^{i-1} i^{-N}$	2	$1 - \frac{1}{2^2} + \frac{1}{3^2} - \dots$	$\frac{\pi^2}{12}$	10^{-5}
6.26.4		4	$1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots$	$\frac{7\pi^4}{720}$	10^{-7}
6.26.5	$\sum_{i=1}^{\infty} (2i)^{-N}$	2	$1 + \frac{1}{3^2} + \frac{1}{5^2} + \dots$	$\frac{\pi^2}{8}$	10^{-4}
6.26.6		4	$1 + \frac{1}{3^4} + \frac{1}{5^4} + \dots$	$\frac{\pi^4}{96}$	10^{-5}
6.26.7	$\sum_{i=1}^{\infty} \frac{(-1)^i}{(2i+1)^N}$	2	$1 - \frac{1}{3} + \frac{1}{5} - \dots$	$\frac{\pi}{4}$	10^{-4}
6.26.8		4	$1 - \frac{1}{3^2} + \frac{1}{5^2} - \dots$	$\frac{\pi^3}{32}$	10^{-5}

7. Литература

1. Інформатика. Базовий курс / Симонович С.В. і ін.- СПб: Питер, 2000. – 640 с.
2. Семашко Г.Л., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація і програмування”, Чернігів, ЧДЕІУ – 2011.

Лабораторна робота № 5

ВИКОРИСТАННЯ ОДНОМІРНИХ І ДВОМІРНИХ МАСИВІВ

1. Мета роботи

Вивчити принципи роботи з одно - і двовимірними масивами на мові Turbo Pascal.

2. Задачі роботи

Навчитися основним алгоритмам розв'язання задач з використанням масивів на мові Turbo Pascal.

3. Порядок виконання роботи

3.1. У середовище програмування Turbo Pascal скласти і налагодити програму розв'язання наступної задачі на тему "Одномірні масиви": *Заданий масив з 8 дійсних чисел виду $a_1, a_2, a_3, \dots, a_8$. Вивести на екран в один рядок усі елементи масиву і виконати завдання у відповідності з зазначеним варіантом додатка А.*

3.2. Отримати чисельні результати і провести їх аналіз.

3.3. Скласти і налагодити програму розв'язання наступної задачі на тему «Двовимірні масиви»:

Заданий цілочисловий двовимірний масив розміру 3×4 . Вивести на екран масив у вигляді матриці, що складається з трьох рядків і чотирьох стовпців, і виконати завдання у відповідності з варіантом додатка Б.

3.4. Проаналізувати отримані результати.

3.5. Скласти звіт про виконану роботу.

4. Вимоги до звіту

Звіт про виконану роботу повинен містити:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи;
- номер варіанта для виконання завдання і умови варіанта;
- блок-схеми рішення задач;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п.7

5. Теоретичні положення

5.1. Загальні відомості про масиви

Масивом називається сукупність елементів однакового типу. Кількість елементів у масиві називається розмірністю масиву; кожний елемент масиву задається своїм порядковим номером у масиві – так званим індексом.

Прикладом масиву може бути список прізвищ студентів однієї групи, де кожний студент однозначно визначається своїм порядковим номером у списку (індексом у масиві).

Для створення типу масивів у мові Pascal використовуються службові слова *TYPE* і *ARRAY*, котрі дозволяють створювати користувальницький тип даних «Масив». Кожному типу масивів присвоюється власне ім'я; елементи, що відносяться до даного масиву, описуються у операторі *VAR* як змінні його типу.

Введення і виведення елементів масиву здійснюється у циклах. Індеси елементів масивів записуються у квадратних дужках.

5.2. Одномірні масиви

Масиви, елементи яких однозначно визначаються одним індексом, називаються одномірними. У вигляді одномірного масиву можна представити, наприклад, врожайності різних полів одного господарства, де першому полю відповідає перше значення врожайності, другому полю – друге тощо.; показники успішності різних груп одного курсу за певним предметом, де кожній з груп відповідає своє значення середнього бала за результатами іспиту.

Мова Turbo Pascal допускає опис у програмах одномірних масивів у наступному вигляді:

TYPE <Ім'я_типу_масиву>=*ARRAY* [<Діапазон_індексів>] *OF* <Тип> ,

де, *TYPE* – службове слово, яке використовується для створення користувальницького типу даних;

ARRAY ... OF – службові слова для опису масивів (“Масив...з”);

<Ім'я_типу_масиву> – ім'я типу масиву, що задається користувачем (вимоги до імен типів збігаються з вимогами до імен змінних);

<Діапазон_індексів> – діапазон зміни індексів масиву. У вигляді даного діапазону може виступити будь-який порядковий тип, крім *LONGINT*. Частіше усього використовують тип-діапазон, у якому задають границі зміни індексів;

<Тип> – тип елементів масиву, наприклад, *REAL*, *INTEGER*, *CHAR* тощо.

Приклади опису типу масивів:

```
TYPE M=ARRAY [1..10] OF INTEGER;
```

```
TYPE AR=ARRAY [0..15] OF REAL;
```

```
TYPE VT=ARRAY [-5..5] OF SINGLE;
```

```
TYPE GRUP=ARRAY [1..27] OF STRING[10].
```

Змінні, що відносяться до того чи іншого типу масивів, повинні бути відповідно оголошені у розділі опису змінних.

Наприклад:

```
VAR REZ:M;
```

```
VAR P:AR;
```

```
VAR A:VT;
```

```
VAR STUD:GRUP.
```

Можна також описати змінну як масив, що безпосередньо має тип, наприклад:

```
VAR REZ:ARRAY [1..10] OF INTEGER.
```

При звертанні до елементів масиву у тексті програми обов'язково потрібно вказувати їх порядковий номер у масиві, наприклад, *REZ[1]*, *POLE[0]*, *A[5]*, *STUD[i]* (де *i* належить діапазону зміни індексів масиву).

Одномірні масиви, діапазон індексів яких починається з 1, іноді називають векторами.

Наприклад, якщо заданий вектор *V* з 5 чисел (3, 0, 5, -7, 8), то другий елемент вектора дорівнює 0 (тобто $V[2]=0$), п'ятий елемент вектора дорівнює 8 (тобто $V[5]=8$) тощо.

5.3. Приклади основних типів задач з використанням одномірних масивів

До стандартних типів задач на використання одномірних масивів відносяться такі:

- знаходження суми (добутку) елементів масиву;
- визначення найбільшого (найменшого) елемента в масиві;
- знаходження кількості елементів масиву, що задовольняють певним умовам;
- упорядкування масиву у порядку зростання (спадання) елементів.

Розглянемо перераховані типи задач на наступних прикладах.

5.3.1. Знаходження суми елементів масиву

Нехай відомо, що у автопарку, що має 18 машин марки КАМАЗ, кожний з КАМАЗів перевіз за день певний об'єм вантажу. Визначити сумарний об'єм перевезень вантажів за день.

При розв'язанні задачі будемо використовувати тип масиву КАМАЗ для опису усіх КАМАЗів автопарку; змінну $P[i]$ для опису об'єму вантажу, що перевезений i -ою машиною за день (i змінюється від 1 до 18).

Текст програми може при цьому може виглядати наступним чином:

```
program pr1;
type KAMAZ =array[1..18] of real;
var i:integer;
    P:KAMAZ;
    S:real;
begin
s:=0;
for i:=1 to 18 do
begin
writeln ('Введіть об'єм перевезень ', i, '-ї машини, m');
readln (p[i]);
s:=s+p[i];
end;
writeln ('Сумарний об'єм перевезень s=',s:8:2, ' m');
end.
```

Накопичення суми у даному прикладі буде проводитися за кроками, при введенні для чергової машини значення обсягу перевезень сума буде збільшуватися на дану величину.

Аналогічно реалізується і алгоритм знаходження добутку елементів масиву (з заміною початкового значення суми $S:=0$ на початкове значення добутку $S:=1$, і з заміною операції додавання елементів масиву «+» на операцію множення «*»).

5.3.2. Знаходження найбільшого елемента у масиві

Відома середньомісячна зарплатня усіх 16 співробітників одного відділу. Знайти величину найбільшої середньомісячної зарплатні відділі.

Для опису списку зарплат співробітників скористаємося типом масиву *ZARPL*, для завдання зарплатні кожного із співробітників – змінною *SOTR*.

Текст програми для наведеного прикладу:

```
Program pr2;
Type ZARPL=array[1..16] of real;
Var i:integer;
    sotr:ZARPL;
    Max: real;
Begin
For i:=1 to 16 do
Begin
Writeln ('Введіть зарплату ', i, ' -го співробітника, грн. ');
Readln(sotr[i]);
End;
Max:=sotr[1];
For i:=2 to 16 do if max<sotr[i] then max:=sotr[i];
Writeln('Найбільша зарплата =',max:10:2, ' грн. ');
End.
```

У даній програмі реалізований наступний алгоритм. В початковий момент передбачається, що найбільшою є зарплата першого із співробітників. Далі величина цієї зарплати порівнюється з зарплатами інших співробітників, і, якщо знайдеться зарплата, більша, чим передбачуваний максимум, то змінній *Max* присвоюється це значення.

Аналогічно може бути знайдений і найменший елемент у масиві. Для цього достатньо замінити у викладеній програмі умову *max<sotr[i]* на умову *min>sotr[i]*, де *i* змінюється від 1 до 16, *min* – змінна типу *REAL*.

5.3.3. Знаходження кількості елементів масиву, що задовольняють будь-яку умову

Відомі результати іспиту 22 студентів однієї групи з інформатики. Визначити, скільки студентів склали іспит на 4 і 5.

Один з варіантів розв'язання поставленої задачі має наступний вигляд:

```
Program pr3;
Label 1;
Type INF=array[1..22] of integer;
Var stud:INF;
    i,p:integer;
begin
p:=0;
for i:=1 to 22 do
begin
    1: writeln('Введіть оцінку ',i,'-го студента');
    readln(stud[i]);
    if (stud[i]<1) or (stud[i]>5) then goto 1;
    if stud[i]>3 then p:=p+1;
end;
writeln('На 4 і 5 здали іспит ',p:2,' студентів);
end.
```

В даній програмі для позначення списку оцінок з інформатики використовувався тип масиву *INF*, для позначення оцінок конкретних студентів – змінна *stud*. Програма передбачає перевірку коректності даних що вводяться: при спробі введення неіснуючої за п'ятибальною системою оцінки, програма повторює її введення. Для цього використовується оператор переходу *GOTO*, де ім'я мітки, до якої здійснюється перехід (у даному випадку 1), описується у розділі опису міток *LABEL*.

5.3.4. Сортування масиву за зростанням

Відомі результати змагання 9 учасників зі стрільби. Розташувати дані результати за зростанням набраних при стрільбі балів.

Алгоритм розв'язку даної задачі є найбільш складним з наведених вище прикладів і вимагає використання вкладених циклів.

Один зі способів сортування масивів полягає у наступному. Спочатку перший елемент масиву в циклі почергово порівнюється з усіма елементами, які залишилися. Якщо черговий елемент масиву менший за величиною, ніж перший, то ці елементи переставляються місцями. Порівняння продовжується далі вже для оновленого першого елемента. В результаті закінчення даного циклу буде знайдений і встановлений на перше місце найменший елемент масиву. Далі продовжується аналогічний процес уже для елементів масиву, які залишилися, тобто другий елемент порівнюється з усіма іншими і при необхідності, переставляється з ними місцями. Після визначення і встановлення другого елемента масиву, даний процес продовжується для третього елемента, четвертого елемента тощо. Алгоритм завершується, коли відбудеться порівняння і впорядкування передостаннього і останнього з елементів масиву, які залишилися.

Програма реалізації викладеного алгоритму може мати наступний вигляд:

```

Program pr4;
Type STREL=array[1..9]of integer;
Var rez:strel;
    i,j,s:integer;
Begin
For i:=1 to 9 do
begin
writeln('Введіть результати ',i,'-го учасника);
readln(rez[i]);
end;
for i:=1 to 8 do
for j:=i+1 to 9 do
if rez[i]>rez[j] then
begin
s:=rez[j];
rez[j]:=rez[i];
rez[i]:=s;
end;
writeln('Відсортовані результати по зростанню :');
for i:=1 to 9 do write (rez[i]:5,' ');
end.

```

Тут *STREL* – тип масиву результатів пострілів учасників, *rez[i]* – змінна для опису результатів *i*-го учасника (*i* змінюється від 1 до 9). Допоміжна змінна *s* використовується при перестановці місцями елементів масиву.

5.4. Двовимірні масиви

Масив, для завдання елементів якого потрібно використовувати два індекси, називається двовимірним.

Приклади двовимірних масивів: місця в залі театру, де кожне місце характеризується номером ряду і номером крісла у ряді; таблиця результатів спортивного чемпіонату, де всі команди характеризуються результатом гри з кожним із суперників тощо.

Двовимірні масиви, у яких діапазони індексів починаються з 1, також називаються іноді матрицями. Розмірність кожної матриці визначається як $M \times N$, де M – число рядків у матриці, N – число стовпців.

Наприклад, задана матриця A розмірністю 3×3 має наступний вигляд:

$$A = \begin{pmatrix} 3 & 7 & 9 \\ 5 & 2 & 8 \\ 7 & 5 & 8 \end{pmatrix}$$

Тут елемент $A[2,1]=5$, $A[1,2]=7$, $A[3,2]=5$, $A[3,3]=8$. Перший індекс елемента двовимірного масиву задає номер рядка матриці, в якій розташований елемент, другий індекс – номер стовпця.

Якщо кількість рядків матриці дорівнює кількості стовпців, то матриці даного типу називаються квадратними. Елементи квадратної матриці виду $B[1,1]$, $B[2,2]$, $B[3,3]$... складають головну діагональ матриці. Іноді вводять поняття побічної діагоналі квадратної матриці для елементів виду $B[1,N]$, $B[2,N-1]$, $B[3,N-2]$... $B[N,1]$, де N – кількість рядків (стовпців) матриці.

Опис типів двовимірних масивів у мові Turbo Pascal здійснюється аналогічно до опису типів одномірних масивів з додаванням діапазону зміни другого індексу.

Приклади:

```
TYPE MATR=[1..4,1..5] OF INTEGER;
```

```
TYPE B=[2..9,0..6] OF REAL;
```

```
TYPE C=ARRAY[1..4,-1..4] OF CHAR.
```

Також допускається вказівка імені другого типу масиву як типу елементів масиву, наприклад:

```
TYPE VEC=ARRAY[1..4] OF REAL;
```

```
MAS=ARRAY[1..5] OF VEC.
```

В результаті наведеного вище опису, тип масиву *MAS* буде оголошений як тип двовимірного масиву, перший індекс якого буде змінюватися від 1 до 5, а другий індекс – від 1 до 4, тобто розмірність масиву складе 5x4 елементів.

При введенні і виведенні елементів двовимірних масивів використовуються вкладені цикли, у яких зовнішній оператор циклу, як правило, задає зміну рядків масиву, внутрішній оператор циклу – зміну стовпців.

5.5. Приклади задач з використанням двовимірних масивів

При розв'язанні задач на використання двовимірних масивів застосовуються, як правило, ті ж алгоритми, що й при роботі з одномірними масивами. Проілюструємо це на конкретних прикладах.

5.5.1. Знаходження найбільшого елементу у заданому рядку матриці

Нехай задана матриця *A* з дійсних чисел розміру 3x4. Знайти найбільший елемент у другому рядку даної матриці.

Наведемо програму розв'язання задачі у вигляді:

```
Program max_st;  
  Type Matr=array[1..3,1..4] of real;  
  Var max:real;  
      a:Matr;  
      i,j:integer;  
  
begin  
  for i:=1 to 3 do  
    for j:=1 to 4 do  
begin  
  writeln('Введіть елемент a[';i;',';j;']');  
  readln(a[i,j]);  
end;  
    max:=a[2,1];  
    for j:=2 to 4 do  
      if max<a[2,j] then max:=a[2,j];  
      writeln('Найбільший елемент другого  
рядка=';max:8:2);  
end.
```

Дана програма представляє собою реалізацію алгоритму знаходження найбільшого елементу вектора, отриманого шляхом фіксування одного з індексів двовимірного масиву.

5.5.2. Знаходження елементів масиву, що задовольняють певну умову

Відомі результати 5 студентів за підсумками іспитів з історії та інформатики. Знайти прізвища студентів, які здали обидва іспити на відмінно.

Для розв'язання поставленої задачі може бути використана наступна програма:

```
program Sessia;
  type PR=array [1..5,1..2]of integer;
        Fam=array[1..5]of string[10];
  var r:pr;
      st:fam;
      i,j:integer;
  begin
    for i:=1 to 5 do
      begin
        writeln('Введіть прізвище ',i,'-го студента ');
        readln(st[i]);
        writeln(' Введіть оцінку даного студента по
історії (ом 2 до 5)');
        readln(r[i,1]);
        writeln('Введіть оцінку даного студента
по інформатиці (ом 2 до 5)');
        readln(r[i,2]);
      end;
    for i:=1 to 5 do
      if (r[i,1]=5) and (r[i,2]=5) then writeln('Студент-
відмінник - ',st[i]);
    end.
```

В даній програмі для збереження прізвищ студентів використовується одномірний строковий масив *st* типу *Fam*, для збереження оцінок студентів – двовимірний цілочисловий масив *r* типу *PR*, причому перший стовпець матриці *r* використовується для збереження результатів іспиту з історії, другий стовпець – іспиту з інформатики. Якщо у певного студента оцінки за обидва іспити склали 5 балів, то його прізвище буде виведено на екран з повідомленням «Студент-відмінник».

5.5.3. Знаходження сум елементів рядків матриці

Розглянемо задачу знаходження сум елементів рядків матриці на прикладі задачі підрахунку підсумків футбольного чемпіонату.

Нехай задана таблиця результатів ігор 5 команд футбольного чемпіонату розміру 5x5. На діагоналі таблиці стоять значення 0, інші елементи таблиці дорівнюють 0, 1 або 2, де 0 балів відповідає програшу команди у грі, 1 бал – нічий, 2 бала – виграшу. Визначити суму балів кожної команди за результатами чемпіонату.

Легко помітити, що для побудови матриці R результатів ігор достатньо ввести лише половину матриці, яка стоїть вище (або нижче) головної діагоналі, тому що результати інших ігор можуть бути розраховані з відомого співвідношення: якщо, наприклад, перша команда обіграла другу, то елемент $R[1,2]=2$, а елемент $R[2,1]=2-R[1,2]=0$; аналогічно, якщо друга команда зіграла в нічию з третьою, то $R[2,3]=1$, $R[3,2]=2-R[2,3]=1$. Таким чином, неважко отримати вигляд взаємозв'язку елементів матриці: $R[i,j]+R[j,i]=2$, де i і j змінюються від 1 до 5 (крім елементів головної діагоналі). На головній діагоналі матриці R за умовою задачі завжди стоять числа 0.

Перейдемо до написання програми рішення поставленої задачі.

```
Program foot;
```

```
  Type tab=array[1..5,1..5] of integer;
```

```
  Var r:tab;
```

```
      i,j,s:integer;
```

```
begin
```

```
  {введення елементів матриці, що стоять вище діагоналі }
```

```
    for i:=1 to 4 do
```

```
      for j:=i+1 to 5 do
```

```
        begin
```

```
          writeln ('Введіть результат гри ',i,'-ї команди з ',j,'-ю:  
0, 1 або 2 бала);
```

```
          readln(r[i,j]);
```

```
        end;
```

```
  {заповнення елементів, що стоять на діагоналі нулями}
```

```
    for i:=1 to 5 do r[i,i]:=0;
```

```

{обчислення елементів матриці, що стоять нижче діагоналі }
    for i:=2 to 5 do
        for j:=1 to i-1 do r[i,j]:=2-r[j,i];

        {виведення на екран матриці результатів ігор}
        writeln("Таблиця чемпіонату);
        for i:=1 to 5 do
            begin
                for j:=1 to 5 do write(r[i,j]:4);
                writeln;
            end;

            {обчислення сум елементів рядків матриці}
            for i:=1 to 5 do
                begin
                    s:=0;
                    for j:=1 to 5 do s:=s+r[i,j];
                    writeln(i,'-а команда набрала ',s:3,' балів);
                end;
            end.

```

6. Методичні рекомендації до виконання роботи

6.1. При виконанні завдання скористатися алгоритмами розв'язання стандартних задач по темі «Одномірні масиви», викладеними в п.5.3.1-5.3.4.

6.2. Провести чисельні розрахунки для вектора (4, 7, -9, 5, 1, -5, 10, 7). Проаналізувати правильність отриманих результатів.

6.3. При розв'язанні задачі скористатися прикладами програм в п.5.5.1-5.5.3 і алгоритмами п.5.3.1-5.3.4.

Приклад виведення на екран двовимірного масиву у вигляді матриці розмірності 3x4 наводиться у наступному фрагменту програми:

```

...
for i:=1 to 3 do
    begin
        for j:=1 to 4 do write(a[i,j]:5);
        writeln;
    end;

```

...
Проаналізувати отримані результати.

7. Контрольні запитання

- 7.1 Що розуміють під масивом даних?
- 7.2 Що називають розмірністю масиву?
- 7.3 Що розуміють під індексом елемента масиву?
- 7.4 Який масив називається одномірним?
- 7.5 Наведіть приклади одномірних масивів.
- 7.6 Як описуються одномірні масиви на мові PASCAL?
- 7.7 Як задається діапазон зміни індексів масиву?
- 7.8 Як позначаються індекси масивів на мові PASCAL?
- 7.9 Які стандартні алгоритми по роботі з одномірними масивами Ви знаєте?
- 7.10 Поясніть поняття двовимірного масиву, матриці.
- 7.11 Що позначають індекси матриці?
- 7.12 Скільки елементів у матриці з 7 рядків і 9 стовпців?
- 7.13 Дайте означення квадратної матриці, діагоналей квадратної матриці.
- 7.14 Наведіть приклад опису двовимірних масивів на мові PASCAL.
- 7.15 Поясніть порядок використання вкладених циклів при введенні елементів двовимірного масиву.

8. Література

1. Інформатика. Базовий курс / Симонович С.В. і ін.– СПб: Питер, 2000. – 640 с.
2. Семашко Г.А., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація і програмування”, Чернігів, ЧДЕІУ – 2011.
5. Войтюшенко Н.М. Інформатика і комп'ютерна техніка: Навч. пос. з баз.підготовки для студ.екон. і техн.спеціальностей ден. і заочн. форм навчання. - К.: Центр навчальної літератури, 2006. – 568 с.
6. Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч.посіб. Вид. 2-ге, перероб., доп. - К.: Академвидав, 2007.- 416 с.(Альма-матер).
7. Інформатика і комп'ютерна техніка: Навчальний посібник / За ред. М.Є. Рогози. - К.: Видавничий центр «Академія», 2006. – 368 с.

Додаток А

Варіанти завдань на тему «Одномірні масиви»

- 1) Знайти середнє арифметичне всіх елементів масиву.
- 2) Знайти найменший елемент у масиві.
- 3) Знайти кількість позитивних елементів масиву.
- 4) Знайти кількість негативних елементів масиву.
- 5) Визначити, скільки раз зустрічається число 7 серед елементів масиву.
- 6) Визначити, скільки елементів масиву менше за число 6.
- 7) Визначити, скільки елементів масиву більше за число 3.
- 8) Знайти суму усіх позитивних елементів масиву.
- 9) Знайти найменше з чисел $a_1, 2a_2, 3a_3, \dots, 8a_8$.
- 10) Знайти найбільше з чисел $2a_1, 3a_2, 4a_3, \dots, 9a_8$.
- 11) Знайти суму $a_1+2a_2+3a_3+\dots+8a_8$.
- 12) Знайти найменший по модулю елемент масиву.
- 13) Знайти різниця між найбільшим і найменшим елементами масиву.
- 14) Відсортувати масив у порядку спадання елементів.
- 15) Знайти добуток негативних елементів масиву.
- 16) Знайти суму усіх елементів масиву, що мають парні індекси.
- 17) Знайти найбільший з елементів масиву, що мають непарні індекси.
- 18) Знайти середнє арифметичне усіх позитивних елементів масиву.
- 19) Знайти середнє арифметичне усіх негативних елементів масиву.
- 20) Знайти суму елементів масиву, що перевищують число 5.

Додаток Б
Варіанти завдань на тему «Двовимірні масиви»

- 1) Знайти найбільший елемент у третьому стовпці матриці.
- 2) Знайти суму усіх елементів матриці.
- 3) Знайти суму усіх позитивних елементів матриці.
- 4) Знайти найменший елемент у першому рядку матриці.
- 5) Знайти середнє арифметичне значення елементів другого стовпця матриці.
- 6) Знайти добуток елементів третього стовпця матриці.
- 7) Знайти суму елементів першого і третього рядків матриці.
- 8) Визначити, скільки разів зустрічається число 5 серед елементів усієї матриці.
- 9) Визначити, скільки разів зустрічається число 1 серед елементів третього рядка матриці.
- 10) Знайти кількість від'ємних елементів у матриці.
- 11) Знайти кількість від'ємних елементів у першому рядку матриці.
- 12) Знайти кількість від'ємних елементів у другому стовпці матриці.
- 13) Знайти суму елементів другого рядка матриці.
- 14) Знайти найбільший елемент у матриці.
- 15) Знайти найменший елемент у матриці.
- 16) Знайти найменший за модулем елемент у першому рядку матриці.
- 17) Знайти середнє арифметичне значення усіх негативних елементів матриці.
- 18) Визначити, скільки елементів матриці перевищує число 2.
- 19) Розташувати усі елементи матриці в рядок в порядку зростання.
- 20) Розташувати усі елементи матриці у рядок в порядку спадання.

Лабораторна робота № 6

ОБРОБКА ТЕКСТОВИХ ДАНИХ В СЕРЕДОВИЩІ ПРОГРАМУВАННЯ TURBO PASCAL

1. Мета роботи

Навчитися принципам обробки даних у середовищі програмування *Turbo Pascal*.

2. Задачі роботи

Вивчити прийоми розв'язання задач обробки текстових даних.

3. Порядок виконання роботи

3.1. У середовищі програмування *Turbo Pascal* скласти і налагодити програму розв'язку наступної задачі:

Заданий список студентів групи. Роздрукувати даний список у стовпець і виконати завдання у відповідності з зазначеним викладачем варіантом додатка А.

3.2. Провести аналіз результатів роботи програми.

3.3. Скласти звіт про виконану роботу.

4. Вимоги до звіту

Звіт про виконану роботу повинен містити:

- назву і мету роботи, хід виконання роботи, відомості про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи;
- номер варіанта для виконання завдання і умови свого варіанта;
- блок-схеми рішення задач;
- тексти програм;
- отримані при розрахунках чисельні результати;
- письмові відповіді на контрольні запитання п.6

5. Теоретичні положення

5.1. Опис змінних строкового типу

Для обробки текстів у середовищі програмування *Turbo Pascal* використовується тип даних *STRING* (рядок). Кількість символів у рядку-змінній задається об'явою типу *STRING[N]*, де значення *N* може бути будь-якою константою порядкового типу, не більшою за 255.

Об'ява типу *STRING[N]* вказує, що кількість символів у рядку-змінній може змінюватися у діапазоні від 1 до *N*. Якщо при опису типа кількість символів у рядку не зазначена, то за умовчання довжина рядка приймається максимально можливою, тобто 255 символів.

Рядок у *Turbo Pascal* трактується як ланцюжок символів. До любого символу у рядку можна звернутися так само, як і до елементу символного масиву типу *ARRAY [1..N] OF CHAR*. Над символами рядка можна здійснювати необхідні дії і, тим самим, змінювати зміст рядка, його довжину тощо.

Приклад:

```
program pr1;
  var st:string[6];
      i:integer;
  begin
    st:='primer';
    for i:=1 to 6 do
      write(st[i], ' ');
  end.
```

В результаті роботи даної програми на екрані будуть виведені наступні значення:

p r i m e r

Таким чином, першим символом *st[1]* у змінній *st* є буква *p*, другим символом *st[2]* є *r*, третім – *i* тощо.

До рядків можна застосовувати операцію зчеплення, наприклад:

```
program pr2;
  var st:string[3];
  begin
    st:='a'+ 'b'+ 'c';
  writeln(st);
  end.
```

В результаті виконання запропонованої вище програми змінній *st* буде присвоєно значення 'abc'.

Допускається порівняння рядкових змінних. Операції відношень =, <>, >, <, >=, <= виконуються над двома рядками позначково, зліва направо, з урахуванням внутрішнього кодування символів. Приклади порівняння рядкових змінних з результатом «Істина»:

```
'A' < 'B'  
'Turbo Pascal' > 'Turbo'  
'2' > '1'
```

5.2. Процедури і функції по роботі з рядками

Дії над рядками реалізуються у Turbo Pascal за допомогою наступних вбудованих процедур і функцій.

5.2.1. Функція *Concat(S1,S2,...,SN)*: *string* – зчіплює (об'єднує) рядки *S1, S2,..., SN*. Параметри, зазначені у квадратних дужках, не є обов'язковими.

Приклад використання функції:

```
program pr3;  
  var  
    S: string;  
begin  
  S := Concat('ABC', 'D');  
  writeln(S);           {ABCD}  
end.
```

Тут і далі (у прикладах використання рядкових операторів) у фігурних дужках наводиться виведений на екран результат роботи програми.

5.2.2. Функція *Copy(S: string; Index, Count: integer): string* – копіює з рядка *S* кількість *Count* символів, починаючи із символу з номером *Index*.

Приклад:

```
program pr4;  
  var S: string;  
begin  
  S := 'ABCDEF';  
  S := Copy(S, 2, 3);  
  writeln(S);           {BCD}  
end.
```


5.2.3. Функція **Length(S: string):byte**

– визначає довжину рядка S.

Приклад використання функції:

```
program pr5;
  var
    S: string;
begin
  s:='ABCD';
  Writeln(Length(S));    {4}
end.
```

5.2.4. Функція **Pos(SubS, S: string):byte**

– знаходить у рядку S перше входження підрядка SubS і визначає номер позиції, з якої вона починається. Якщо підрядок не знайдений, то видається 0.

Наприклад:

```
program pr6;
  var S: string;
begin
  S := ' 123.5';
  {Заміна всіх пробілів нулями}
  while Pos(' ', S) > 0 do
    S[Pos(' ', S)] := '0';
  writeln(S);    {000123.5}
end.
```

5.2.5. Функція UpCase(C: char):char – перетворює рядкову (строкову) латинську букву у прописну. Будь-які інші символи залишаються без перетворення.

Приклад:

```
program pr7;
  var
    s : string;
begin
  s:='red';
  s[1] := UpCase(s[1]);
  writeln(s);    {Red}
end.
```

5.2.6. Процедура **Delete**(*var S: string; Index, Count: integer*)

– видаляє кількість *Count* символів з рядка *S*, починаючи з символу з номером *Index*.

Приклад використання процедури:

```
program pr8;
  var
    s: string;
begin
  s := 'Шльончик Павло ';
  Delete(s, 10, 5);
  writeln(s);    {Шльончик}
end.
```

5.2.7. Процедура **Insert**(*SubS: string; var S: string; Index: integer*) – вставляє підрядок *SubS* у рядок *S*, починаючи з символу з номером *Index*.

Приклад:

```
program pr9;
  var
    S: string;
begin
  S := 'Петро Іванов';
  Insert('Петрович ', S, 7);
  writeln(S);    {Петро Петрович Іванов}
end.
```

5.2.8. Процедура **Str**(*X [:Width [:Decimals]]; Var S: string*)

– перетворює число *X* у рядок символів *S*. Параметри *Width* і *Decimals* задають формат перетворення (загальну ширину поля і кількість символів у дробовій частині відповідно).

Приклад:

```
program pr10;
  var
    S: string[12];
begin
  Str(5461, S);
  writeln ('Результат: '+S+' бал');
{Результат: 5461 бал}
end.
```

5.2.9. Процедура *Val(S: String; var X; var Code: integer)* –

перетворює рядкове значення *S* у його чисельне представлення *X* типу *Real* або *Integer*. Параметр *Code* містить ознаку помилки перетворення (0 – немає помилки), наприклад:

```
program pr11;
  var I, Code: Integer;
      S:string;
  begin
    S:='7908';
    Val(S, I, Code);
    writeln(I, ' ',Code);      {7908 0}
  end.
```

5.3. Алгоритми розв'язування задач із використанням рядкових змінних

Розглянемо алгоритми роботи з рядковими змінними на прикладі деяких задач.

Задача 1. Заданий список з 6 слів. Обчислити, скільки слів списку починається на букву «п».

Один із можливих розв'язків даної задачі має наступний вигляд:

```
program z1;
  var
    s: string[20];
    i,k:integer;
  begin
    k:=0;
    for i:=1 to 6 do
      begin
        writeln('Введи слово');
        readln(s);
        if s[1]='п' then k:=k+1;
      end;
    writeln(k);
  end.
```

У вищенаведеній програмі обробка слів відбувається без їх наступного збереження, тобто на кожному кроці циклу змінної *s* присвоюється значення чергового слова і відбувається порівняння першої літери *s[1]* даного слова *s* зі зразком 'п'.

Примітка. Аналогічно можна встановити й кількість слів, які закінчуються на ту чи іншу букву. У цьому випадку достатньо ввести якусь проміжну цілочислову змінну *len*, яка слугує для позначення довжини слова *s* і яка обчислюється як *length(s)*, після чого проводити порівняння останньої букви *s[len]* слова *s* зі зразком.

Задача 2. Заданий список з 8 слів. Знайти найкоротше слово у списку. Якщо таких слів декілька, то роздрукувати їх в один стовпець.

Розв'язок поставленої задачі зводиться до декількох етапів: ввести список слів у вигляді масиву рядкових змінних; підрахувати довжину кожного рядка; встановити найменшу з довжин; роздрукувати ті рядки масиву, довжина яких збігається з найменшою.

Приклад можливого розв'язання поставленої задачі:

```
program z2;
  type t=array[1..8] of string[20];
  tt=array[1..8] of integer;
  var s: t;
      n: tt;
      i, min: integer;
begin
  for i:=1 to 8 do
  begin
    writeln('Введіть слово');
    readln(s[i]);
    n[i]:=length(s[i]);
  end;
  min:=n[1];
  for i:=2 to 8 do
    if min>n[i] then min:=n[i];
  for i:=1 to 8 do
    if n[i]=min then writeln(s[i]);
end.
```

У даній програмі для позначення слів використовується масив рядкових змінних *s*, для позначень відповідних їм довжин слів – цілочисловий масив *n*. Найменший елемент масиву *n* зберігається у вигляді змінної *min*.

Задача 3. Заданий рядок з двох слів, розділених пробілом.

Змінити місцями у даному рядку слова.

Приклад програми розв'язання задачі:

```
program z3;
  var s: string[40];
      s1,s2: string[20];
      i,n: integer;
begin
  writeln('Введіть рядок');
  readln(s);
  n:=length(s);
  i:=pos(' ', s);
  s1:=copy(s, 1, i);
  s2:=copy(s, i+1, n-i);
  s:=s2 + ' ' + s1;

  writeln(s);
end.
```

У даній програмі змінна *s* служить для позначення рядка з двох слів, змінні *s1* і *s2* – для позначення відповідно першого і другого слова рядка. Алгоритм розв'язання полягає у тому, що спочатку за допомогою оператора *Pos* з'ясовується, яким символом за рахунком у рядку *s* знаходиться пробіл. Далі у рядку виділяються два підрядка: один включає всі символи ліворуч пробілу (перше слово), інший – праворуч пробілу (друге слово). В кінці програми слова зчіплюються у зворотному порядку з використанням розділяючого їх пробілу.

Задача 4. Нехай задана певна довільна цифра. Створити програму, яка дозволяє вивести на екран назву даної цифри у прописному вигляді, наприклад, «5 – п'ять», «8 - вісім» тощо

Можливий розв'язок задачі:

```
program z4;
  var n:integer;
      s:string[30];
begin
  writeln('Введіть цифру');
  readln(n);
  case n of
    0: s:='нуль';
    1: s:='один';
    2: s:='два';
```

```

3: s:='три';
4: s:='чотири';
5: s:='п'ять';
6: s:='шість';
7: s:='сім';
8: s:='вісім';
9: s:='дев'ять';
else
s:=' не є цифрою';
end;
writeln(n,' - ', s);
end.

```

У даній програмі використовується оператор множинного вибору CASE, який дозволяє відповідно зіставити кожній введений цифрі її назву. При спробі введення чисел, які не є цифрами, на екран видається повідомлення «не є цифрою».

Задача 5. Заданий двовимірний масив, що складається з 3 рядків і 4 стовпців. З'ясувати, скільки разів зустрічається латинська буква «a» у першому і останньому стовпцях масиву і змінити місцями дані стовпці.

Приклад розв'язання задачі:

```

program z5;
type mas=array[1..3,1..4] of string[10];
var s: mas;
i, j, k, n_1, n_4: integer;
c: string[10];
begin
for i:=1 to 3 do
for j:=1 to 4 do
begin
writeln('Введіть s[', i:2, j:2, ']');
readln(s[i,j]);
end;
writeln('початковий масив');
for i:=1 to 3 do
begin
for j:=1 to 4 do write(s[i,j]:10, ' ');
writeln;
end;
n_1:=0;

```

```

n_4:=0;
for i:=1 to 3 do
  for k:=1 to 10 do
    begin
      if s[i,1][k]='a' then n_1:=n_1+1;
      if s[i,4][k]='a' then n_4:=n_4+1;
    end;
  for i:=1 to 3 do
    begin
      c:=s[i,1];
      s[i,1]:=s[i,4];
      s[i,4]:=c;
    end;
  writeln('В 1-му стовпці ',n_1, ' а');
  writeln('В 4-му стовпці ',n_4, ' а');
  writeln('Отриманий масив');
  for i:=1 to 3 do
    begin
      for j:=1 to 4 do write(s[i,j]:10,' ');
    end;
  writeln;
end.

```

Тут допоміжна змінна *c* використовується при перестановці місцями елементів першого і четвертого стовпців; змінні *n_1* і *n_4* містять кількість літер «а» у першому і четвертому стовпцях матриці відповідно.

6. Контрольні запитання

- 6.1 Для чого слугує тип *STRING*?
- 6.2 Як задати кількість символів у рядковій змінній?
- 6.3 Скільки символів можуть містити строкові змінні *a* і *b*, якщо вони були описані як *var a: STRING; b: STRING[15]*?
- 6.4 Як можна звернутися у програмі до першого символу рядкової змінної *s*?
- 6.5 Як відбувається порівняння строкових змінних?
- 6.6 Якими способами можна зчепити рядкові змінні?
- 6.7 Поясніть призначення і формат оператора *Length*.
- 6.8 Чому дорівнює значення функції *Length('program')*?
- 6.9 Як додати декілька символів у рядкову змінну?
- 6.10 Поясніть призначення операторів *Str* і *Val*.
- 6.11 Для чого служить оператор *UpCase*?

7. Література

1. Інформатика. Базовий курс / Симонович С.В. і ін.- СПб: Питер, 2000. – 640 с.
2. Семашко Г.А., Салтыков А.И. Програмування на мові Паскаль – М.: Наука, 1998. – 128 с.
3. Фаронов В.В. Турбо Паскаль 7.0 – М.: Нолидж, 2000. – 576 с.
4. Зацерковний В.І., Гур'єв В.І., Фірсова І.В. Конспект лекцій з дисципліни „Алгоритмізація і програмування”, Чернігів, ЧДЕІУ – 2011.
5. Войтюшенко Н.М. Інформатика і комп'ютерна техніка: Навч. пос. з баз.підготовки для студ.екон. і техн.спеціальностей ден. і заочн. форм навчання. - К.: Центр навчальної літератури, 2006. – 568 с.
6. Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч.посіб. Вид. 2-ге, перероб., доп. - К.: Академвидав, 2007.- 416 с.(Альма-матер).
7. Інформатика і комп'ютерна техніка: Навчальний посібник / За ред. М.Є. Рогози. - К.: Видавничий центр «Академія», 2006. – 368 с.

Додаток А

Варіанти завдань для виконання

1. Встановити, скільки прізвищ у списку групи починаються з літери «А».
2. Встановити, скільки прізвищ у списку групи мають закінчення «ко».
3. Встановити, скільки прізвищ у списку групи закінчуються на букву «к».
4. Встановити, скільки букв у найдовшому прізвищі списку групи.
5. Встановити, скільки букв у найкоротшому прізвищі списку групи.
6. Знайти найдовше прізвище у списку. Якщо таких прізвищ декілька, то роздрукувати їх у один рядок.
7. Знайти у списку усі прізвища, що починаються з букв «В» або «Г».
8. Знайти у списку усі прізвища, що мають закінчення «а».
9. Знайти у списку усі прізвища, що починаються з «М».
10. Встановити, скільки прізвищ у списку групи складаються менше ніж з 6 букв.
11. Встановити, скільки прізвищ у списку складаються більш ніж з 8 букв.
12. Встановити, скільки прізвищ у списку складаються з 8 або 9 букв.
13. Знайти у списку усі прізвища, що складаються з 6 або 7 букв.
14. З'ясувати, чи є у списку прізвище «Леоненко». Якщо є, то виправити її на «Левоненко».
15. З'ясувати, скільки разів у списку зустрічається прізвище «Іванов».
16. Поміняти місцями перше і останнє прізвища у списку.
17. Замінити друге прізвище у списку на «Гриценко» і видалити із списку третє прізвище.
18. З'ясувати, чи є у списку прізвищ «Шаріков». Якщо є, то видалити його зі списку.
19. Впорядкувати список у алфавітному порядку.
20. Встановити, чи є у списку однофамільці.

РОЗДІЛ II

„ПРОГРАМУВАННЯ В СЕРЕДОВИЩЕ – MATHCAD 2001i”

Практична робота № 1

1. Мета роботи

Знайомство з системою комп'ютерної математики MathCAD 2001i

2. Задачі роботи

Оволодіти практичними навичками роботи з системою MathCAD 2001i

3. Зміст роботи

3.1 Завантажити додаток MathCAD 2001i.

3.2 Ознайомитися з областями екрана, засвоїти роботу з панелями інструментів. По черзі включити декілька пунктів панелі “Математика”, ознайомитися з принципом їх розташування та виклику їхніх опцій.

3.3 Виконати розрахунки, у відповідності із варіантом. Варіант вибрати згідно списку у журналі.

4. Вимоги до звіту

Звіт повинний включати:

- титульний лист з назвою роботи та відомостями про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи.
- постановку задачі дослідження, послідовність її виконання;
- результати обчислень згідно варіанту (на дискеті і в роздрукованому вигляді);
- відповіді на контрольні запитання.

5. Рекомендована література

5.1 Дьяконов В.П. MathCAD 2001i і MathCAD 20011.-М.: СОЛОН Пресс, 2004.-832.

5.2 Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Загальні положення

MathCad – це система комп'ютерної математики для автоматизації розв'язання інженерних задач в різних галузях науки, техніки й освіти. Назва системи походить від двох слів MATNematica (математика) і (CAD (Computer Aided Design – система автоматичного проектування або САПР). Можна сказати, що MathCAD – це САПР у математиці.

Перші версії системи, розроблені фірмою MathSoft (США), набули широкої популярності ще в середині 80-х років ХХ ст. І по цей день вони залишаються єдиними математичними системами, в яких опис розв'язку математичних задач здійснюється за допомогою звичних математичних формул і знаків. При цьому важливо відзначити, що MathCAD – не тільки засіб для розв'язання математичних задач. Це, по суті, потужна математична система, яка дозволяє готувати на найвищому поліграфічному рівні будь-які матеріали: документацію, наукові звіти, книги й статті, дисертації, дипломні й курсові проекти тощо. При цьому в них одночасно можуть бути присутні тексти складного виду, будь-які математичні формули, графіки функцій та різноманітні ілюстративні матеріали. Дозволяє MathCAD 2001i готувати й високоякісні електронні книги із гіпертекстовими посиланнями.

Серед інших систем комп'ютерної математики, таких як Maple, Matematica і MATLAB, система MathCAD як і раніше виділяється своїм дружнім інтерфейсом, а також зручною і надзвичайно простою у використанні математично і візуально орієнтованою мовою спілкування з користувачем.

6.1 Склад системи

Інтегровані системи MathCAD2001i включають наступні основні компоненти:

- редактор документів – редактор з можливістю вставки математичних виразів, шаблонів графіків і текстових коментарів;
- центр ресурсів – інтегратор ресурсів системи;
- електронні книги – електронні книги з описом типових розрахунків в різних областях науки і техніки;
- довідкова система – система для отримання довідкових даних з тематичного і індексного каталогів, а також для пошуку необхідних даних за ключовим словом або фразою;

- „швидкі шпаргалки” – короткі приклади з мінімальними коментарями, які описують застосування всіх вбудованих операторів і функцій системи;
- браузер Інтернету – власний засіб виходу в Інтернет.

6.2 Документи і робочі аркуші MathCAD 2001i

Характерною рисою MathCAD є так звані документи, які поєднують опис математичного алгоритму розв'язання задачі (або ряду задач) з текстовими коментарями і результатами обчислень, заданих у вигляді символів, чисел, таблиць або графіків. Фактично документи MathCAD – програма на візуально-орієнтованій мові програмування, тобто програма не у вигляді малозрозумілих кодів, а у вигляді візуально зрозумілих об'єктів.

Документ програми MathCAD називається робочим аркушем. Він містить формули і текстові блоки. В процесі розрахунків формули обробляються послідовно, зліва направо і зверху вниз, а текстові блоки ігноруються.

Введення інформації здійснюється в місці розташування курсору. Програма MathCAD використовує три види курсорів. Якщо жоден з об'єктів не обраний, використовується хрестоподібний курсор, який визначає місце створення наступного об'єкта.

При введенні формул використовується кутовий курсор, який вказує поточний елемент виразу. При введенні даних в текстовий блок використовується текстовий курсор у вигляді вертикальної риски.

Мова програмування MathCAD орієнтована на математичні обчислення і тому практично не відрізняється від звичайної мови математичних статей, звітів, книг, тобто математичні алгоритми описуються в природній математичній формі із застосуванням загальноприйнятої символіки для математичних знаків, таких як квадратний корінь, знак ділення у вигляді горизонтальної риски, знак інтегралу тощо. Це робить документ, зображений на екрані монітора, надзвичайно схожим на сторінки математичних книг.

Можна сказати, що в MathCAD ідея розв'язання задач без їх програмування доведена до досконалості: значна кількість інженерних задач потребує тільки коректного формального опису розв'язання і не потребує програмування в загальноприйнятому розумінні.

6.3 Завантаження MathCAD 2001i

Пакет MathCAD завантажується або через кнопку „Пуск”
➤ „Програми” ➤ „MathCAD Professional 2001i”, або
подвійним клацанням на ярлику на робочому столі
Windows (рис. 1.1).



Рис. 1.1. Завантаження MathCAD Professional 2001i

Після запуску MathCAD через певний час (залежить від швидкодії ПК) з'явиться заставка (рис. 1.2), яка швидко змінюється вікном системи (рис. 1.3)

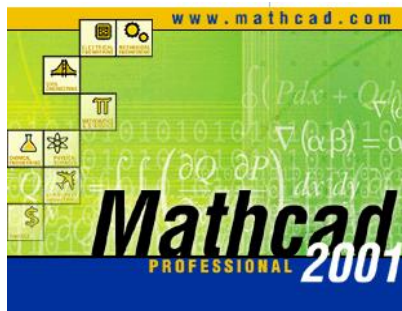


Рис. 1.2. Заставка MathCAD Professional 2001i

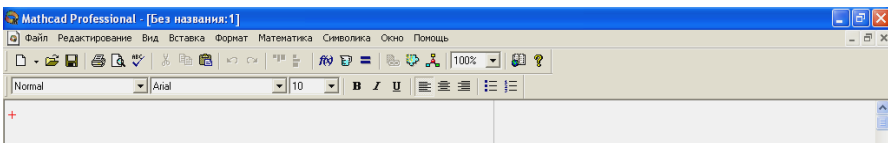


Рис. 1.3. Робоче вікно MathCAD Professional 2001i

6.4 Робота з інтерфейсом користувача

Фактично, інтегрована прикладна система MathCad інтегрує в собі три редактори: *формульний, текстовий і графічний*. Для запуску формульного редактора достатньо встановити покажчик миші у будь-якому вільному місці вікна редагування і клацнути лівою кнопкою. У цьому місці буде знаходитись курсор введення, який має вигляд маленького червоного хрестика. Його можна переміщати клавішами переміщення курсору. Не треба плутати курсор введення з курсором миші, який має вигляд об'ємної стрілки ↖.

Відразу після запуску система готова до створення документа з необхідними користувачеві обчисленнями. Перша ж кнопка панелі інструментів (з зображенням чистого аркуша паперу) „**Новий**” дозволяє почати підготовку нового документа. Відповідне йому вікно редагування, отримує назву „**Без названня: N**”, де **N** – порядковий номер документа, який починається з цифри 1. На початку роботи вікно редагування очищене.

Користувальницький інтерфейс системи створений таким чином, що користувач, який має елементарні навички роботи з Windows-додатками, може відразу почати роботу з MathCAD. Інтерфейс системи зовні дуже нагадує інтерфейс широко відомого текстового процесора MS Word.

Характерними елементами інтерфейсу вікна (рис. 1.4) є:

- **рядок заголовка** – рядок з ім'ям системи і поточного документа, а також з кнопками управління вікном системи;
- **рядок меню** – рядок, який відкриває доступ до пунктів меню з різними командами;
- **панель інструментів** – панель з кнопками (значками), які забезпечують швидке виконання найбільш важливих команд при роботі з системою;
- **панель форматування** – панель з кнопками (значками), які забезпечують швидке форматування текстових та формульних блоків в документах;
- **панель виводу палітр математичних знаків** – панель з кнопками (значками), які виводять палітри спеціальних математичних знаків і грецьких букв;
- **рядок стану** – рядок контекстної довідки для роботи з системою, який відображає поточний стан останньої;
- **курсор введення** – курсор (маркер) у вигляді червоного хрестика, який позначає місце введення блоку документу;

- **смуги прокручування** – елементи управління, які розташовані вздовж нижньої і правої границь поточного вікна і призначені для прокрутки зображення на екрані по горизонталі і/або вертикалі;
- **лінія поділу сторінок** – вертикальна лінія, яка відокремлює поточну сторінку від сусідньої праворуч;
- **спливаюча підказка** – невелике текстове жовте поле, яке з'являється при наведенні покажчика миші на більшість елементів інтерфейсу і блоки у вікні редагування (на малюнку спливаюча підказка відноситься до кнопки виводу палітри символівних операцій);
- **координатна лінійка** – лінійка з діленнями, котра дозволяє (якщо це потрібно) точно розташовувати блоки по горизонталі.

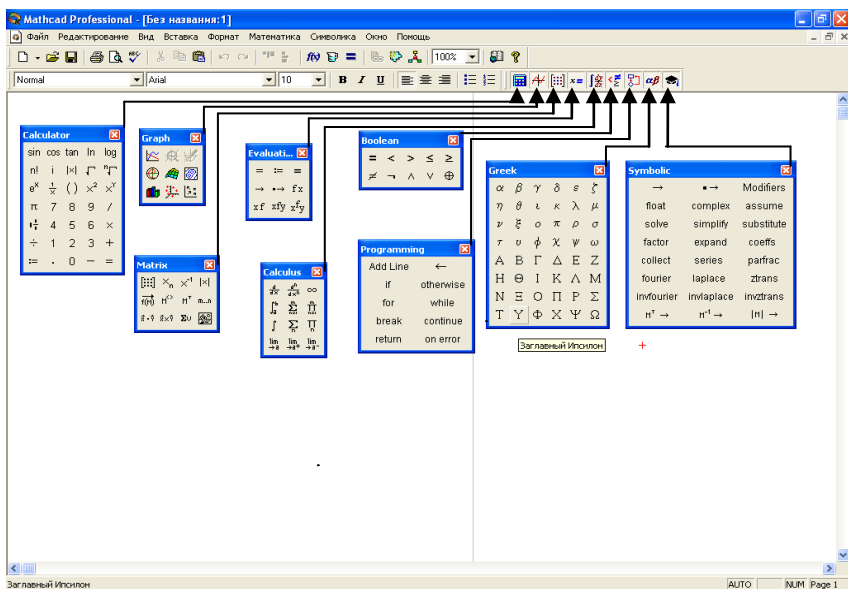


Рис. 1.4. Елементи інтерфейсу вікна MathCAD Professional 2001i

Примітка. Окремі деталі інтерфейсу можна приховувати або виводити на екран за допомогою команд меню „Вид”.

Курсор введення і лінія розподілу сторінок. Уважно придивіться до вікна редагування (рис. 1.3). Навіть коли вікно очищене, воно виявляється не зовсім порожнім. Зазвичай, на ньому можна бачити два об'єкти: курсор (маркер) у вигляді червоного хрестика і вертикальну лінію, яка відокремлює поточну сторінку від сусідньої праворуч. Положення лінії визначається заданими за замовчанням параметрами сторінки документа. А ось курсор введення встановлюється мишею. Зазвичай він знаходиться у лівому верхньому куті вікна документа. Для його вставки в інше місце, необхідно підвести туди покажчик миші і клацнути лівою кнопкою миші.

Примітка В MathCad 200і можна вилучити вертикальну лінію розподілу сторінок, якщо у вікні настроювання параметрів сторінки задати альбомну орієнтацію сторінок, встановивши прапорець „Альбомная”.

Рядок заголовка. Рядок заголовка знаходиться у верхній частині вікна Mathcad. Він відображає назву завантаженого документа або документа, який вводиться з клавіатури. У лівій частині рядка є стандартна кнопка управління вікном, а в правій частині – три маленькі кнопки для згортання вікна, розгортання його у весь екран і закриття.

Рядок головного меню. Другий рядок вікна системи – головне меню. В рядку меню представлені наступні пункти (рис.1.5):

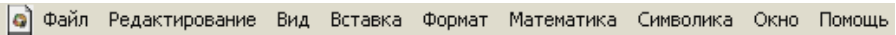


Рис. 1.5. Рядок головного меню Mathcad

„**Файл**” – робота з файлами, Інтернетом та електронною поштою;

„**Редактирование**” – редагування документів;

„**Вид**” – зміна способів представлення документів і приховування/відображення елементів інтерфейсу;

„**Вставка**” – вставка об'єктів і їх графіки;

„**Формат**” – зміна формату об'єкта;

„**Математика**” – управління процесом обчислень;

„**Символика**” – вибір опцій символічного процесора;

„**Окно**” – управління вікнами системи;

„**Помощь**” – робота з довідковою системою.

Меню MathCad 2001i – контекстні. Це означає, що число позицій в них і їх призначення, залежать від стану системи.

Робота з документами MathCAD не вимагає обов'язкового використання можливостей головного меню, оскільки основні з них дублюються кнопками швидкого управління.

Рядок панелі інструментів знаходяться під рядком головного меню. Панель включає декілька груп кнопок управління, кожна з яких дублює найбільш важливі команди меню. Дивлячись на ці кнопки, можна легко зрозуміти їх функції. При підведенні покажчика миші на будь-яку з них, з'являється спливаюча підказка з ім'ям цієї кнопки. На початку панелі інструментів є характерна вішка переміщення у вигляді опуклої вертикальної риски. Зачепившись за неї покажчиком миші і утримуючи ліву кнопку, можна перетягти панель у будь-яке місце вікна редагування або „приліпити” її до будь-якої сторони вікна.

Панелі інструментів можна виводити на екран або вилучати з нього за допомогою відповідних опцій позиції „**Вид**” головного меню. Зазвичай є дві такі панелі (рис. 1.6): панель інструментів (дублююча ряд найбільш поширених команд і операцій) і панель форматування для вибору типу і розміру шрифтів і способів вирівнювання текстових коментарів.

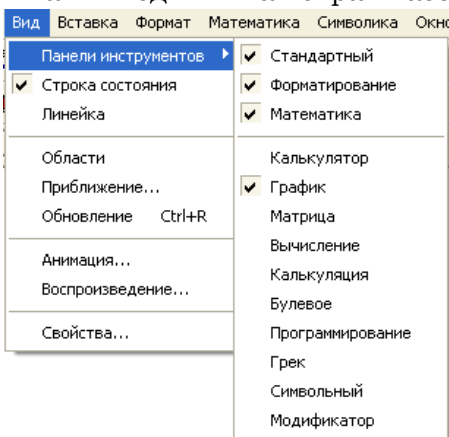


Рис. 1.6. Меню „Вид»”

Для зручності також можна вивести панель математичних символів і операторів („**Математика**”). Вона слугує для виводу заготовок – шаблонів математичних знаків (цифр, знаків арифметичних операцій, матриць, знаків інтегралів, похідних тощо) (рис. 1.7).

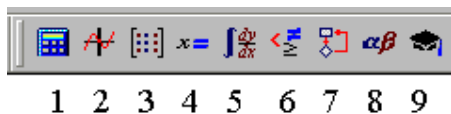


Рис. 1.7. Панель математичних символів

- 1 – арифметичні інструменти;
- 2 – інструменти графіків;
- 3 – векторні і матричні операції;
- 4 – інструменти деяких знаків;
- 5 – оператори математичного аналізу;
- 6 – панель логіки;
- 7 – інструменти програмування;
- 8 – символи грецького алфавіту;
- 9 – символічні оператори

Третій рядок вікна системи займає панель інструментів (рис. 1.8), яка містить декілька груп кнопок управління з піктограмами, кожна з яких дублює одну з найважливіших операцій головного меню. Дивлячись на ці піктограми, можна легко зрозуміти їх функції. Потрібно тільки зупинити курсор миші на будь-якій з цих піктограм, як у вікні з'явиться текст, який пояснює функції піктограми.

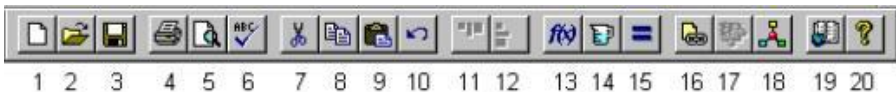


Рис. 1.8. Панель інструментів

Кнопки операцій з файлами. Документи системи MathCAD є файлами, тобто іменами блоків збереження інформації на магнітних дисках. Файли можна створювати, завантажувати (відкривати), записувати і роздруковувати на принтері. Можливі операції з файлами представлені на панелі інструментів першою групою з трьох кнопок:

- 1 – **(Новый)** – створення нового документа з очисткою вікна редагування;
- 2 – **(Открыть)** – завантаження раніше створеного документа з діалогового вікна;
- 3 – **(Сохранить)** – запис поточного документа з його ім'ям.

Друк і контроль документів. Ця група представлена трьома кнопками:

- 4 – **(Печать)** – друкування документа на принтері;
- 5 – **(Просмотр)** – попередній перегляд документа;
- 6 – **(Проверка)** – перевірка орфографії документа.

Кнопки операцій редагування. Наступні чотири кнопки призначються для виконання операцій редагування документів:

7 – (**Вирезать**) – перенесення виділеної частини документа в буфер обміну з очисткою цієї частини документа;

8 – (**Копировать**) – копіювання виділеної частини документа в буфер обміну зі збереженням виділеної частини документа;

9 – (**Вставить**) – перенесення вмісту буфера обміну у вікно редагування на місце, вказане курсором миші;

10 – (**Отменить**) – скасування попередньої операції редагування.

Три останні операції пов'язані із використанням буфера обміну. Його можливості і призначення добре відомі користувачам Windows. Він призначений для тимчасового збереження даних і їх переносу з однієї частини документа в іншу або для організації обміну даними між різними додатками.

Кнопки розміщення блоків. Документи складаються з різних блоків: текстових, формульних, графічних тощо. Блоки проглядаються системою, інтерпретуються і виконуються. Перегляд йде справа вліво і знизу вгору. Дві операції розміщення блоків представлені кнопками наступної групи:

11 – (**Выровнять по горизонтали**) – блоки вирівнюються за горизонталлю;

12 – (**Выровнять вниз**) – блоки вирівнюються за вертикаллю, розташовуючись зверху вниз. Піктограми цих кнопок зображують блоки і вказані варіанти їх розташування. Певне розміщення блоків в деяких випадках має значення. Наприклад, нерідко результат операції бажано розмістити в одному рядку із завданням на здійснення операції. Розміщення блоків за вертикаллю характерно для великих документів.

Кнопки операцій з виразами. Формульні блоки часто є обчислювальними виразами, або виразами, що входять в склад заданих користувачем нових функцій. Для роботи із виразами служать піктограми:

13 – (**Вставить функцию**) – вставка функції зі списку, що з'являється у діалоговому вікні;

14 – (**Вставить единицы измерений**) – вставка одиниць виміру;

15 – (**Вычислить**) – обчислення виділеного виразу.

MathCAD має багато вбудованих функцій: від елементарних до складних статистичних і спеціальних математичних. Синтаксис їх запису часом легко забувається. Тому можливість вставки функції є надзвичайно корисною. Інша кнопка, **Вставить единицы измерений**, дозволяє вставити потрібну одиницю виміру.

Якщо документи великі, то при їх змінах не завжди зручно запускати обчислення із самого початку. Операція **Вычислить** дозволяє запускати обчислення для виділених блоків, що може зменшити час обчислень.

16 – **(Вставка гіперссылки)** – забезпечує створення гіперпосилання;

17 – **(Мастер компонентів)** – відкриває вікно Майстра, який забезпечує зручний доступ до всіх компонентів системи;

18 – **(Запуск системи MathConnex)** – запуск системи для стимулювання блочно-заданих пристроїв.

Кнопки управління ресурсами. Заключна група з двох піктограм забезпечує звертання до центру ресурсів системи і до вбудованої довідкової бази даних системи:

19 – **(Ресурс-центр)** – дає доступ до центру ресурсів;

20 – **(Справка)** – дає доступ до ресурсів довідкової бази даних системи.

Панель форматування містить типові засоби управління шрифтами: набір стилів, три кнопки для зміни гарнітури шрифту (напівжирний, похилий і підкреслений), а також три кнопки для вирівнювання тексту в абзацах (за лівим краєм, за центром і за правим краєм). Перераховані засоби дозволяють формувати документи, надаючи їм вид, потрібний користувачеві (рис. 1.9).



Рис. 1.9. Панель форматування

1 – перемикач вибору стилів;

2 – перемикач вибору набору символів;

3 – перемикач вибору розмірів символів;

4 – установка жирних символів;

5 – установка скошених символів;

6 – установка підкреслених символів;

7 – установка лівостороннього вирівнювання;

8 – установка вирівнювання за центром;

9 – установка правостороннього вирівнювання.

7. Контрольні запитання

7.1. Поясніть призначення панелі інструментів “Математика”.

7.2. Що в MathCAD розуміють під документом?

7.3. Як можна запустити MathCAD?

7.4. Назвіть характерні елементи інтерфейсу MathCAD

7.5. Поясніть призначення та склад головного меню MathCAD

7.6. Поясніть призначення та склад панелі інструментів MathCAD?

7.7. Поясніть призначення та склад панелі математичних символів MathCAD?

7.8. Які способи введення квадратного кореня Ви знаєте?

7.9. Як вводиться і відображується оператор множення?

7.10. Як вводиться і відображується оператор ділення?

Додаток А

№	Обчислити	Обчислити	Обчислити	Обчислити	Обчислити
1	$5 \cdot \frac{6}{9} + \frac{(2-3) \cdot 9}{5} \cdot 2$	$4^2 + 6^{2+6}$	$\sqrt{36^{-1}}$	$\sin\left(\frac{\pi}{3}\right)^2 + \cos\left(\frac{\pi}{3}\right)^2$	$\ln\left[(2+33)^e\right]$
2	$\frac{9}{56+6} - (3+9) \cdot 8$	$6 \cdot 3^{0,25}$	$\sqrt[3]{9 \cdot 3}$	$\tan(2\pi + 6.28)$	$\log\left(\frac{10^5}{0.01}\right)$
3	$3 + \frac{9}{(87-2) \cdot 8}$	$23^4 + 32^{-6}$	$6 \cdot \sqrt[5]{4}$	$\sqrt{\cos(\pi \cdot 0.5) \cdot \sin(1)}$	$\ln\left[(e \cdot e)^e\right] + 25$
4	$\frac{2}{(9+6) \cdot 8} - 5$	$0.25^3 + 69^2$	$\sqrt{99\sqrt[6]{5}}$	$\cos(\pi) \cdot \sqrt{\sin(28\pi)}$	$\log(\cos(0 \cdot \pi))$
5	$8 \cdot \frac{9-9}{2} + \frac{1}{14}$	$\frac{2^e}{4^2}$	$\sqrt[6]{\frac{32+69}{\sqrt{25}}}$	$\frac{\sin(0.333\pi)}{\cos(1+\pi)}$	$4 \log(10^e)$
6	$\frac{2+36}{93} - 11 \cdot 6$	$5^{-3 \cdot 9} + 3^2$	$5^e \cdot \sqrt{89}$	$\tan(\pi + \sin(2\pi))$	$e^{\ln(e^{-1})}$
7	$(0.25+0.3) \cdot 2 \cdot (9+3)$	$25e^{0.92}$	$\sqrt{8} \cdot \sqrt{2}$	$\sqrt{\cos(1.667\pi) - \tan(\pi)}$	$\log(101 - \ln(e))$
8	$35 \cdot 2 - \left(\frac{36 \cdot 85}{5}\right)$	$69e^2$	$\sqrt[3]{125 + \frac{4 \cdot 10000}{2}}$	$\frac{1}{1 - \sin(0.25)}$	$\log\left(e^{-10} + \frac{15}{3} \cdot 2\right)$
9	$6 \cdot 9 - (32+3) \cdot \frac{9}{3}$	$3e^3$	$\left(\sqrt[4]{45^3}\right)^{0.15}$	$\cos(2\pi - \tan(0.333\pi))$	$\sin(\ln(e^\pi))$
10	$3 + \frac{9}{6+2} - 0.23$	$6^{2+3 \cdot 21} + 3e^8$	$\left(\sqrt[3]{2} + \sqrt{3}\right)^{-1}$	$\sqrt{\tan(8\pi - 0.5\pi) + 1}$	$\log(88 + 4^{-1})$
11	$(3+21) \cdot 6 + 2$	$e^5 + 9^2$	$\sqrt[10]{(26-1)^{-11}}$	$\sin(1.5\pi + \pi^2)$	$\ln(2.718) - \log(10^9)$
12	$\frac{365}{369} + (362 \cdot 32 + 13)$	$13^{2 \cdot 31} + 3^{-1}$	$\sqrt{5 - 0.5^{-1}} + \sqrt[9]{1}$	$\cos(\pi^3) + \tan(2\pi)$	$\ln(2) \cdot \log(e)$
13	$36 \cdot \frac{9}{52} + 89$	$\frac{-1}{5^7}$	$\sqrt{9^{(2+5)}} \cdot 2^2 \sqrt[12]{8}$	$\tan\left(\frac{\sin(0.667)}{\cos(0.667)}\right)$	$\log(\sqrt[9]{9})$
14	$32 \cdot (0.12 + 59) - \frac{325}{5}$	$5^{[(-1)^2]}$	$9\sqrt{2} + \sqrt[3]{3}$	$(\sin(2\pi) + \cos(4\pi))^2$	$\tan(\cos(\ln(e)))$
15	$65 \cdot \left(\frac{89}{22} + \frac{28}{11}\right) + 56$	0.25^{-23+19}	$\frac{0.75}{\sqrt[3]{0.5^2 + 69^{-1}}}$	$\sin\left(\frac{5 \cdot \pi \cdot \cos(\pi)}{\sin(\pi)}\right)^2$	$\log(125)^{\frac{1}{3}}$
16	$\frac{12}{96 \cdot 32} + \frac{69}{2}$	$2.3^{(-1-6) \cdot 2}$	$156^{-1} \cdot \sqrt[7]{74^{25}}$	$\tan\left(2\pi^{-1}\right)^{-1}$	$1 - \log(78 \cdot 12)$
17	$26 \cdot \frac{19}{3 \cdot 99} + 32 - 21$	6.2^e	$\sqrt[3]{e^e \cdot 2^{10}}$	$\cos[(0.33\pi)^{\sin(0 \cdot \pi)}]$	$\log\left(\frac{78}{\cos(0.33\pi)}\right)$
18	$20 + \frac{96}{5-32} \cdot 3.1$	$98^{-1} \cdot 98^2$	$45\sqrt{25} \cdot \sqrt[3]{125^{-3}}$	$10 + \sin\left(\frac{10\pi}{6}\right)$	$\log(78)^{2+\ln(7)}$
19	$\frac{369}{(3.2-3.13+9) \cdot 8} \cdot 5$	$(5.1^{2+1})^2$	$97 + \sqrt{89 + \sqrt{21} \cdot 3}$	$2 \cdot \sin(\pi) \cdot \cos(\pi)$	$\log\left(\frac{\ln(e)}{\ln(e^2)}\right)$
20	$65 + \frac{75}{3+65-21} - 65$	$0.1^3 \cdot 10^3$	$\frac{15+61}{\sqrt{5^{-1}}}$	$\cos\left(\frac{\sqrt{\pi^3}}{\pi+1}\right)$	$\log\left(\cos\left(\frac{\pi}{4}\right)\right)$

Практична робота № 2

ОБЧИСЛЕННЯ ВИЗНАЧЕНИХ ІНТЕГРАЛІВ ТА ПОБУДОВА ПІДІНТЕГРАЛЬНИХ ФУНКЦІЙ

1. Мета роботи

Навчитися програмувати в системі MathCAD 2001i

2. Задачі роботи

Засвоїти прийоми роботи з системою MathCAD 2001i!

3. Зміст роботи

3.1 Завантажити додаток MathCAD 2001i.

3.2 Ознайомитися з основними прийомами редагування документа.

3.3 Обчислити визначений інтеграл від заданої викладачем функції (додаток А).

3.4 Побудувати таблицю значень і графік підінтегральної функції, оформивши його у відповідності з вказівками викладача.

3.5 Зберегти результати у файлі з введенням необхідних коментарів і оформити звіт.

4. Вимоги до звіту

Звіт повинний включати:

- титульний лист з назвою роботи та відомостями про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи.
- постановку задачі дослідження, послідовність її виконання;
- результати обчислень згідно варіанту (на дискеті і в роздрукованому вигляді);
- відповіді на контрольні запитання.

5. Рекомендована література

5.1 Дьяконов В.П. MathCAD 2001i и MathCAD 2001i.-М.: СОЛОН Пресс, 2004.-832 с.

5.2 Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Основи роботи з системою MathCad

У найпростішому випадку робота з системою Mathcad зводиться до введення у вікні редагування завдання на обчислення і встановлення форматів для їх результатів. Для цього використовуються різні прийоми підготовки блоків. Спочатку розглянемо підготовку блоків двох типів: текстових і формульних. Помітимо, що обчислення виконуються автоматично, оскільки за замовчанням встановлений режим автоматичних обчислень.

6.1. Робота з текстовим редактором

Тексти в математичній системі Mathcad лише на перший погляд мають другорядне значення. Насправді, професійно зроблені в Mathcad документи перед усім повинні мати достатньо докладні текстові коментарі.

Документом називається повний математичний опис алгоритмів розв'язання задач.

Система має потужний редактор документів. Документ складається з блоків, тобто з окремих частин. Блоки можуть бути трьох видів – текстові, обчислювальні і графічні. Кожний блок займає на екрані певний простір, обмежений прямокутною областю.

Текстові блоки відіграють роль не виконуваних коментарів. Вони призначаються лише для підвищення наочності та зрозумілості документа. Обчислювальні блоки складаються з виконуваних математичних виразів, наприклад формул, рівнянь, рівностей і нерівностей тощо. Графічні блоки також є виконуваними і слугують для виводу результатів обчислень у графічному вигляді.

Для введення текстового коментаря достатньо ввести символ " (одні подвійні лапки – не плутайте з двома одиночними лапками або двома апострофами). В прямокутнику, що з'явиться, можна починати вводити текст (рис. 2.1).

Текстова область може мати довільні розміри і розташовуватися у будь-якому місці робочого документа. При цьому потрібно встановити той шрифт, який необхідний для введення тексту. Цей шрифт буде діяти тільки для текстових областей. Для збереження єдиного стилю усього документа можна встановити той же стиль для змінних (**Variables**) і для констант (**Constants**) в перемикачі вибору стилів.

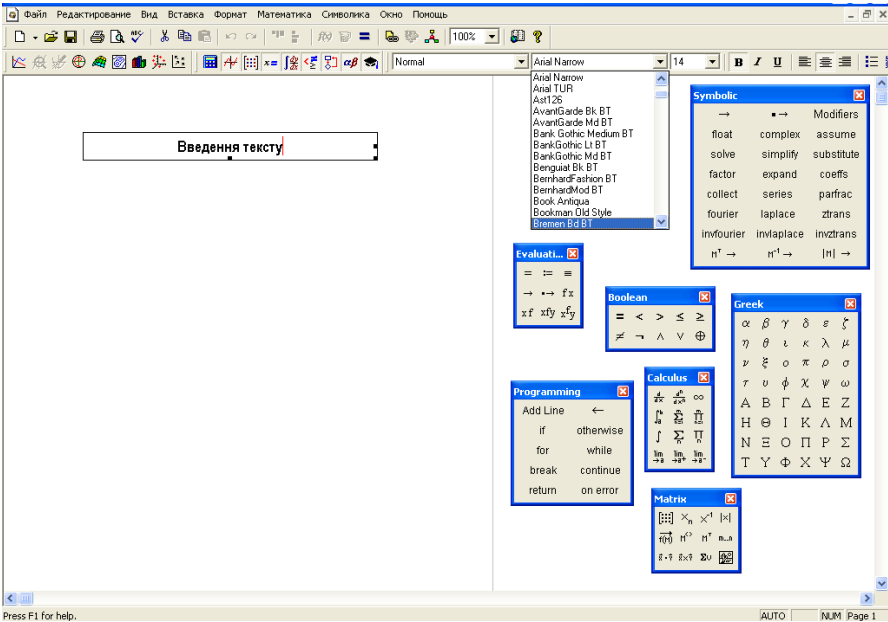


Рис. 2.1. Приклад введення текстового коментаря

В текстовому блоці курсор має вигляд червоної вертикальної риски і помічає місце введення. Текст редагується загальноприйнятими засобами: переміщенням курсору, клавішами управління курсором, встановленням режиму вставки або заміщення символів (клавіша **Insert**), стиранням (клавіші **Del** і **Backspace**), виділенням, копіюванням в буфер обміну, вставкою з буфера тощо.

Досить зручно здійснювати управління стилем тексту за допомогою панелі форматування. В текстовому блоці є також можливість вирівняти текст за лівою межею, правою межею або за центром. Для англomовних текстів передбачений орфографічний контроль із застосуванням вбудованого словника.

Клавіші переміщення курсору можна використовувати з натиснутою клавішею **Shift**, яка забезпечує прискорене переміщення у текстовому блоці. Наприклад, сполучення клавіш **Shift**+← і **Shift**+→ дозволяє переміщати курсор введення не за окремими символами, а за цілими словами.

Початково обмежимося найпростішими правилами введення тексту.

1. Введіть знак подвійних лапок "": при англійській розкладці клавіатури з'явиться прямокутник з курсором введення.

2. За допомогою списку вибору шрифту, який розкривається на панелі форматування (рис. 2.1), встановіть потрібний шрифт.

3. Почніть посимвольно набирати текст, використовуючи типові засоби текстового редактора (клавіші **Пробіл**, **Del**, **Backspace** тощо).

4. Натискайте клавішу **Enter** для переходу на новий рядок (якщо цього не робити, перехід на новий рядок буде здійснюватися автоматично).

5. Для завершення введення тексту відведіть покажчик миші в сторону від текстового блока і клацніть лівою кнопкою миші.

Текстовий блок в ході набору тексту розширюється. Блок має маркери зміни розміру у вигляді маленьких чорних прямокутників, зачепившись за які, блок можна розтягувати у тому або іншому напрямку. Це ілюструє рис. 2.1, на якому представлений розтягнутий блок.

Зверніть увагу, що розмір символів текстового коментаря при розтягуванні (або стисканні) блока не змінюється – змінюється тільки розмір блока. У блок більшого розміру можна ввести більше написів. Як тільки довжина рядка тексту досягає правої границі текстового блока – відбувається перенос тексту, що вводиться на новий рядок усередині блока.

Виділені рамкою текстові блоки можна переносити на інше місце, зачепившись за рамку покажчиком миші – він при цьому перетворюється в зображення чорної долоні. Якщо на початку переміщення натиснути клавішу **Ctrl**, то буде виконуватися перенос блока з його збереженням на початковому місці.

Для завершення створення текстового блока достатньо відвести покажчик миші в сторону від блока і клацнути лівою кнопкою. Можна також натиснути одночасно клавіші **Ctrl+Shift+Enter**.

Редагування документа – це зміна його виду і параметрів. Для здійснення можливості редагування тексту потрібно підвести покажчик миші до конкретного місця і клацнути лівою кнопкою миші. З'явиться рамка текстового блока, а на місці покажчика миші – курсор введення. Тепер можна корегувати текст, дописувати його, вставляти нові символи між старими (режим **Insert**), видаляти символи тощо. Словом, можна користуватися усіма прийомами редагування, добре відомими користувачам будь-якої програми підготовки текстів.

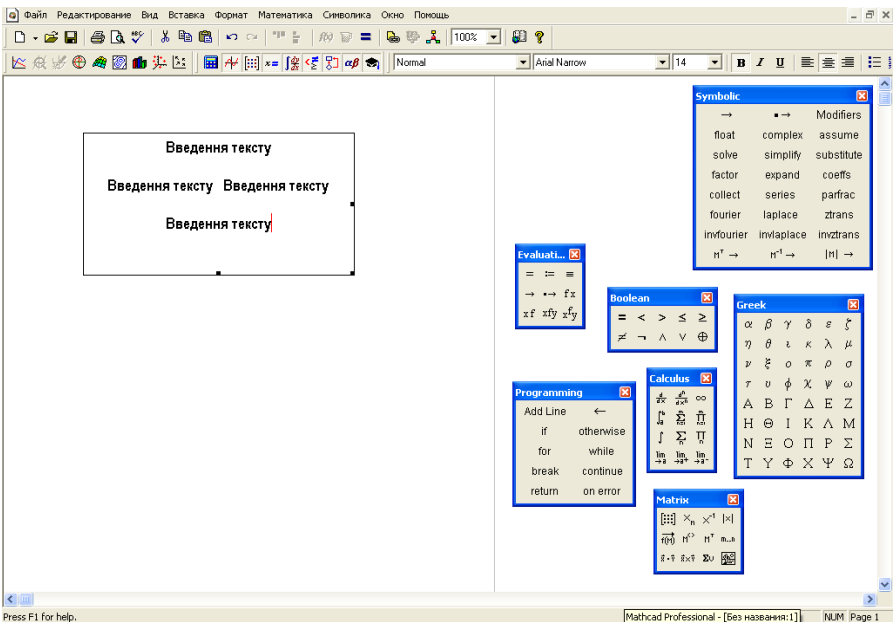


Рис. 2.2. Розтягування текстового блоку і його копіювання

Для введення тексту достатньо ввести знак “(лапки). Поточне положення курсору позначиться в обрамленні рамки. В цю рамку (блок) можна вводити текст. По мірі введення тексту ця рамка автоматично збільшується.

Необхідно відзначити, що курсор MathCAD приймає три різні форми: візира – знак «плюс» червоного кольору, маркера введення тексту – вертикальна червона риска, маркера введення математичних виразів – ріжок («кляшка») синього кольору, розташування якої змінюється при натисканні на клавішу **<Пробіл>**.

В MathCAD є можливість вставки в текстову область математичних формул. Для цього треба виконати операцію **Вставка** > **Математическая область**. Ця операція доступна тільки тоді, коли текстовий маркер встановлений у відповідне місце текстової області. Виконання операції **Вставка** > **Математическая область** призведе до появи в текстовій області шаблону у вигляді маленького чорного квадрата. В ньому можна задавати формули, використовуючи також складальні панелі з математичними символами.

Перенесення, копіювання, вставка і видалення фрагментів. Пункт меню **Правка** > **Вырезать** або натискання комбінації клавіш **Ctrl+X** переносять (вирізують) виділений об'єкт документа до буферу обміну даних. Виділити об'єкт можна суцільним або пунктирним прямокутником. Для виділення суцільним прямокутником необхідно ввести в об'єкт курсор і двічі клацнути лівою кнопкою миші. Для пересилання виділеного об'єкта в буфер обміну можна використовувати натискання клавіші **F3**.

Об'єкт або групу об'єктів можна виділити і пунктирним прямокутником. Підведіть курсор до об'єкта (але не вводьте в нього), клацніть лівою кнопкою миші і, не відпускаючи, перемістіть мишу за діагоналлю. На екрані з'явиться прямокутник з пунктирних ліній, розміри якого задаються переміщенням миші. Як тільки в цей прямокутник потрапить хоча б частина певного об'єкта, цей об'єкт буде обведений прямокутником з пунктирних ліній.

Коли ліва клавіша миші буде відпущена, основний прямокутник зникне, але всі виділені об'єкти залишаться. Так можна виділити відразу декілька об'єктів.

Для виділення одиночних об'єктів доцільно використовувати мишу при натиснутій й утримуваній клавіші **Ctrl** або **Shift**. Підведіть покажчик миші до об'єкту і клацніть лівою кнопкою миші – об'єкт буде обведений пунктирним прямокутником. Після цього перейдіть до наступного об'єкта. Група виділених об'єктів веде себе як єдиний блок, який можна перетаскувати при натиснутій лівій кнопці миші. Для перенесення достатньо утримувати курсором миші край виділеного блоку і домогтися, щоб курсор перетворився в зображення долоні. Це буде означати можливість переміщення блоків. Нагадуємо: якщо курсор здобуває вид двосторонніх стрілок, значить, можна розтягти виділений блок.

При виділенні мишею текстових написів можна виділити і будь-який фрагмент тексту, впритул до одиночного символу, а в формулах – будь-яку змінну. Однак в математичних виразах не можна виділяти окремі символи в іменах функцій.

Можна також перетаскувати об'єкти з одного документа в інший, зачепивши їх курсором миші (в формі долоні), прямо з одного вікна в інше – метод **Drag and Drop** (Перетягни і відпусти). Він набагато полегшує перенесення і робить його більш наочним. Якщо відпустити ліву клавішу миші, то об'єкти з'являться в іншому документі і зникнуть в першому (де вони були до цього).

Пункт **Правка** > **Выделить все** або натискання комбінації клавіш **Ctrl+A** виділяють усі об'єкти пунктирними прямокутниками. Це дозволяє не тільки спостерігати взаємне розташування блоків, але і ясно бачити ступінь їх суміщення. Щоб уникнути останнього, клацніть по пункту **Формат** головного меню, а потім у спадаючому меню – по пункту **Отделить области.**

При перенесенні об'єктів або при їх редагуванні можливе часткове і навіть повне накладення блоків одних об'єктів на інші. У цьому випадку може зникнути частина зображення деяких об'єктів. Виділення усіх об'єктів корисно для контролю такої ситуації.

Редагування різко прискорюється і спрощується при використанні блокових команд:

F2 – поміщає копію блока в спеціальний буфер, тобто місце в пам'яті, де зберігаються дані про блок (сам блок зберігається);

F3 – стирає блок і поміщає в буфер його копію;

F4 – поміщає копію блока з буфера на місце документа, вказане поточним положенням курсору, якщо це місце не зайнято іншим блоком;

Ctrl+F9 – вводить порожній рядок між блоками в місце, зазначене положенням курсору;

Ctrl+F10 – знищує пустий рядок між блоками в місце, зазначене положенням курсору.

Примітка. Нерідко користувач починає набір текстів, забувши встановити ознаку текстового блока (ввести подвійні лапки "). Mathcad 2001 сприймає такий набір як введення математичного виразу. Однак, натиснувши клавішу **Пробіл**, можна перетворити набраний фрагмент в текстовий.

6.2. Робота з формульним редактором

Фактично інтегрована прикладна система Mathcad інтегрує в собі три редактори: формульний, текстовий і графічний. Формули – основні об'єкти робочого аркушу. Новий об'єкт за замовчанням вважається формулою. Для запуску формульного редактора достатньо встановити покажчик миші в будь-якому вільному місці вікна редагування і клацнути лівою кнопкою миші та почати введення букв, цифр, знаків операцій. При цьому створюється область формули, в якій з'явиться кутовий курсор введення, який охоплює поточний елемент формули, наприклад, ім'я змінної (функції) або число. При введенні бінарного оператора по іншу сторону знака операції автоматично з'являється заповнювач у вигляді чорного прямокутника. В це місце вводять черговий операнд.

Курсор можна переміщати клавішами переміщення курсору.

Примітка. Курсор введення не треба плутати з покажчиком миші, який має вигляд жирної похилої стрілки.

Курсор (маркер) введення вказує місце, з якого можна починати набір формул – обчислювальних блоків. Залежно від місця розташування курсор введення може змінювати свою форму. Так, в області формул він перетворюється в синій ріжок („ключку”), яка вказує напрямок та місце введення. Його також можна переміщати клавішами переміщення курсору. Для розширення охопленої куточком області (уприштул до повного охоплення виразу) можна користуватися клавішею **Пробіл**.

Імена змінних і функцій користувача в MathCad можуть мати довільну довжину і складатися з великих і маленьких букв, цифр від 0 до 9, символу нескінченності, грецьких букв, символів підкреслювання і відсотків, нижніх індексів. Розглянемо декілька прикладів.

В системах класу Mathcad 2001 редагування математичних виразів істотно поліпшено і багато в чому нагадує редагування текстових коментарів. Так, можна використовувати режим вставки (клавіша **Insert**, що змінює напрямок охоплення курсором уведення того чи іншого об'єкта), режими стирання (клавіші **Backspace** і **Del**) об'єкта з одної або іншої сторони відносно вертикальної риси курсору уведення, видалення об'єкта командою **Cut** (клавіша **F3**) і його введення на зазначене місце командою **Paste** (клавіша **F4**) і т. ін.

Для управління порядком операцій використовують дужки, які можна вводити вручну. Кутовий курсор дозволяє

автоматизувати такі дії. Щоб виділити елементи формули, які в рамках операції повинні розглядатись як одне ціле, використовують клавішу „Пробіл”. При кожному її натисненні, кутовий курсор „розширюється”, охоплюючи елементи формули, які примикають до даного елементу. Після уведення знака операції елементи в межах кутового курсору автоматично поміщаються у дужки.

Елементи формул можна вводити з клавіатури або за допомогою спеціальних панелей управління. Панелі управління відкривають за допомогою меню **Вид** або кнопками панелі управління **Математика**.

Підготовка і обчислення простих виразів. Розглянемо приклад обчислення відношення суми чисел 2 і 3 до кореня квадратного з числа 5. Спочатку уведемо почерзі символи 2 + 3. Вигляд формульного блока при цьому показаний на рис. 2.3. Зверніть увагу, що при такому послідовному уведенні ріжок охоплює останній операнд.

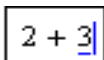

$$2 + 3$$

Рис. 2.3. Створення формульного блока і введення в нього суми чисел 2 і 3

Тепер потрібно ввести знак ділення. Однак, якщо зробити це відразу, то даний знак буде відноситися не до всієї суми, а тільки до останнього операнда – числа 3. Щоб знак ділення відносився до всієї суми, треба виділити увесь вираз 2 + 3. Для цього достатньо натиснути клавішу **Пробіл**. Результат буде мати вигляд, представлений на рис. 2.4.

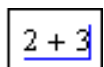

$$2 + 3$$

Рис. 2.4. Виділення усієї суми курсором введення

Тепер можна ввести знак ділення, натиснувши клавішу / (знак похилої риси). Формульний блок прийме вигляд, показаний на рис. 2.5.

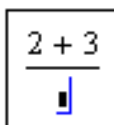
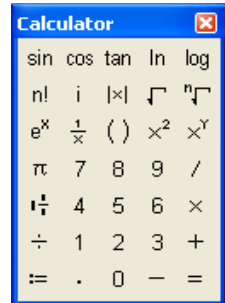

$$\frac{2 + 3}{}$$

Рис. 2.5. Формульний блок після введення знака ділення

Зверніть увагу, що знак ділення у вигляді похилої риски автоматично набув вигляд довгої горизонтальної риски під сумою, а під ним з'явилося місце для введення знаменника дробу у вигляді чорного квадрата, охопленого курсором введення. Зрозуміло, як це прийнято навіть при роботі з калькулятором, можна було б виділити суму дужками, записавши її у вигляді $(2 + 3)$, тоді знак ділення відносився б явно до суми. Однак не слід вводити дужки там, де без них можна легко обійтися.

Наступний етап – введення знака квадратного кореня. Починаючий користувач введе такий знак, використовуючи палітру математичних знаків для арифметичних операцій Calculator. Для такого введення достатньо клацнути мишею по знаку квадратного кореня. Однак більш досвідчений користувач пригадає, що багато знаків вводяться прямо натисканням відповідних клавіш або комбінації клавіш. Наприклад, для введення квадратного кореня достатньо натиснути клавішу зі знаком зворотної похилої риси $\sqrt{\quad}$ (її вигляд нагадує риску, з якої ми починали писати знак квадратного кореня). Тепер формульний блок матиме вигляд, представлений на рис. 2.6. Зверніть увагу на природність запису оператора квадратного кореня і наявність в ньому місця для введення підкореневого виразу.



$$\frac{2 + 3}{\sqrt{\quad}}$$

Рис. 2.6. Формульний блок після введення знака квадратного кореня

Наступний етап конструювання виразу – введення підкореневого виразу у вигляді числа 5. Для цього достатньо натиснути клавішу 5 на цифровій частині клавіатури. Формульний блок матиме вигляд, показаний на рис. 2.7.

$$\frac{2 + 3}{\sqrt{5}}$$

Рис. 2.7. Формульний блок із заданим виразом

Тепер вираз, по суті введений повністю, і нам залишилося побачити результат його обчислення. Для цього в кінці виразу потрібно поставити оператор виводу – знак рівності =. Однак відразу його вводити не можна, оскільки маркер виводу встановлений на останньому операторі. Потрібно виділити весь вираз. Для цього натисніть клавішу **Пробіа** – буде виділений весь чисельник, а потім натисніть клавішу **Пробіа** ще раз – буде виділений весь вираз (рис. 2.8).

$$\frac{2 + 3}{\sqrt{5}}$$

Рис. 2.8. Виділення всього виразу

Залишилося останнє – ввести оператор виводу =. Mathcad одразу автоматично відобразить результат обчислень, оскільки за замовчанням встановлений режим автоматичних обчислень. Вид формульного блока при цьому представлений на рис. 2.9.

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Рис. 2.9. Формульний блок після введення оператора виводу

Mathcad можна легко використовувати для різних експериментів обчислювального характеру. Припустимо, виникла необхідність обчислити результат, коли підкореневим виразом у знаменнику повинно бути число 5 в степені 1.25, Чи потрібно набирати новий вираз спочатку? Звичайно ж, ні. Достатньо модернізувати вже введений вираз (рис. 2.7). Для цього акуратно помістіть покажчик миші після числа 5 і клацніть лівою кнопкою. Ви побачите, що курсор введення відзначить це число (рис. 2.10).

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Рис. 2.10. Зміна одного з операндів у формульному блоці

Неважко здогадатися, що тепер треба ввести знак піднесення до степеня. Його можна вибрати в палітрі арифметичних операцій або просто натиснути клавішу із знаком ^. При цьому у формульному блоці з'явиться запис піднесення числа 5 до будь-якого степеня (рис. 2.10). Місце введення слугує для завдання потрібного степеня.

$$\frac{2 + 3}{\sqrt{5}} = \dots$$

Рис. 2.11. Вид формульного блоку після введення оператора піднесення до степеня

Зрозуміло, що тепер потрібно ввести показник степеня числа 5, тобто число 1.25. Це ілюструє рис. 2.12. Зверніть увагу на природність запису числа 5 в степені 1.25.

$$\frac{2 + 3}{\sqrt{5^{1.25}}} = \dots$$

Рис. 2.12. Вид формульного блоку після введення показника степеня

Таким чином, вираз створений, але обчислення не виконуються, до того часу, поки курсор введення (ріжок) виділяє число 1.25. Для обчислення достатньо відвести покажчик миші від формульного блоку (наприклад, вниз) і клацнути лівою кнопкою. Сконструйований вираз одразу ж буде обчислений (рис. 2.13).

$$\frac{2 + 3}{\sqrt{5^{1.25}}} = 1.829$$

Рис. 2.13. Обчислення відкоригованого виразу

Цей приклад наведений, щоб показати природність роботи з математичними виразами.

6.3. Операції виводу та присвоювання

Для обчислення будь-якого виразу достатньо встановити після нього оператор виведення (знак =). Покажемо це на декількох прикладах простих обчислень. Для введення десяткових чисел в якості роздільника цілої і дробової частин використовується точка, а не кома.

Введення	На екрані дисплея
$1.234 * 2.345 =$	$1.234 * 2.345 = 2.894$
$1/7 =$	$\frac{1}{7} = 0.143$
$\cos(0.5) =$	$\cos(0.5) = 0.878$
$e^2 =$	$e^2 = 7.389$

В математиці для надання обчисленням єдності часто використовуються змінні у вигляді деяких узагальнених позначень даних певного типу. Змінні мають імена (ідентифікатори) і для них характерна операція присвоювання значень. Починаючи з Mathcad 2000, оператор = можна використовувати як оператор першого присвоювання.

Введення	На екрані дисплея
$a = 2$	$a := 2$
$b = 3$	$b := 3$
$a + b =$	$a + b = 5$

Якщо тепер спробувати присвоїти змінним a і b нові значення за допомогою оператора =, то нічого з цього не вийде. Як тільки після назви змінної ми спробуємо поставити знак =, з'явиться старе значення змінної.

Введення	На екрані дисплея
$a =$	$a = 2$
$b =$	$b = 3$

Щоб все ж привласнити змінним нові значення, доведеться використовувати стандартний оператор присвоювання :=, який вводиться своїм першим символом : (двокрапка).

Введення	На екрані дисплея
a : 1	a := 1
b : 1	b := 1
a + b =	a + b = 2

З цих прикладів можна помітити деякі особливості роботи Mathcad при виконанні простих обчислень:

- деякі комбіновані оператори (наприклад, :=) вводяться одним символом;
- Mathcad вставляє пробіли до і після арифметичних операторів;
- оператор множення вводиться як зірочка але представляється точкою в середині рядка;
- оператор ділення вводиться як похила риска, але замінюється горизонтальною рисою;
- оператор піднесення до степеня вводиться знаком ^, але число в степені подається у звичайному вигляді (ступінь як верхній індекс);
- за замовчанням десяткові числа мають подання з трьома знаками після розділової точки;
- Mathcad розуміє найбільш поширені константи, наприклад e – основу натурального логарифму (переконайтесь, що програма розпізнає π або π);
- математичні вирази можуть редагуватися всередині формульного блока з використанням для цього курсору введення і типових прийомів редагування.

Таким чином, навіть без втручання користувача, Mathcad намагається надати математичним виразам звичайний вид. На рис. 2.14 представлений документ, в якому виконані описані вище прості обчислення.

Рис. 2.14 дає уявлення і про деякі додаткові прийоми роботи з системою Mathcad. Наприклад, як бути, якщо точність подання десяткових чисел трьома знаками після крапки не влаштовує користувача. Для цього треба використати форматування чисел. Підведіть покажчик миші до числа (на рис.2.14 це значення e^2) і двічі клацніть лівою кнопкою. З'явиться вікно форматування чисел, показане на рис. 2.14 в правій частині вікна документа. Введіть в поле **Кількість десятичних** (число цифр після десяткової крапки) число 15 замість 3 і отримайте результат з 15 знаками після десяткової точки.

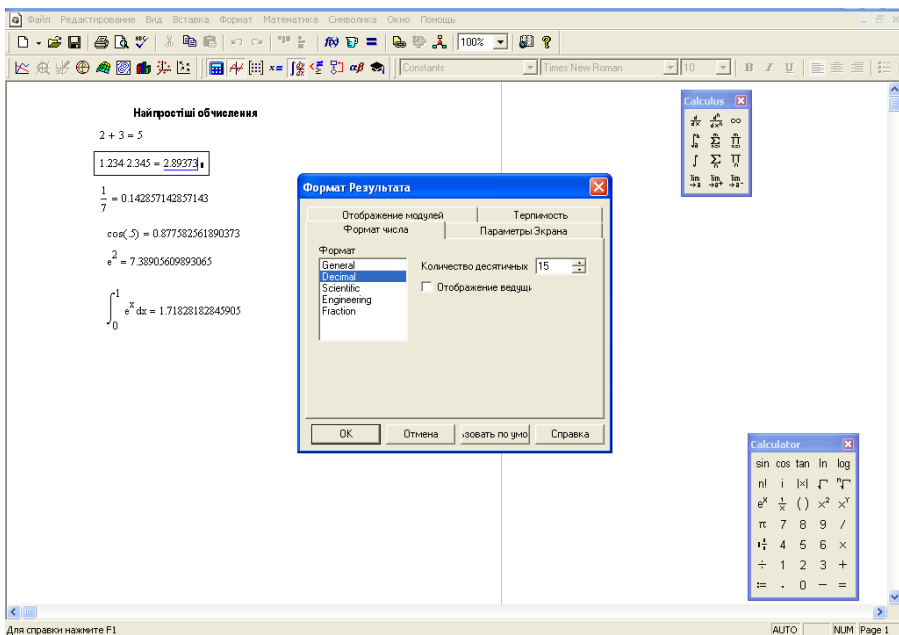


Рис. 2.14 Вікно Mathcad 2001i з найпростішими обчисленнями

В Mathcad 2001 введена нова можливість представлення результатів обчислень з заданою похибкою у вигляді раціональних чисел. Для цього у вікні форматування чисел з'явився додатковий варіант формату – **Fraction** (Дроби). При виборі цього формату Mathcad замість звичайного десяткового числа дає його значення у вигляді відношення цілих чисел.

Mathcad переводить в категорію простих обчислень навіть складні математичні розрахунки, що потребують застосування спеціальних чисельних методів. Пізніше ми розглянемо їх більш докладно, а поки відзначимо останній приклад на рис. 2.14 – обчислення визначеного інтеграла. Для цього був використаний шаблон інтеграла з представленої палітри **Calculus**.

Врахуйте те, що Mathcad веде чисельні розрахунки з кінцевою похибкою. Її максимальне значення задається значенням змінної **TOL**. За замовчанням $TOL=0.001$, тобто гарантується похибка не більш одиниці третього знака після десяткової крапки. Чим менше значення TOL, тим вище точність обчислень, але більший час, що витрачається на їх виконання. Без похибки виконання ряду обчислень можливо в символічному режимі.

Застосування шаблонів математичних операторів і символів

Підготовка обчислювальних блоків полегшується завдяки виводу шаблону при завданні того чи іншого оператора. Для цього в Mathcad існують палітри математичних символів і шаблонів операторів і функцій. Припустимо, виникла необхідність обчислити вже названий визначений інтеграл. Для цього спочатку потрібно вивести на екран палітру операторів математичного аналізу. Клацніть на кнопки із зображенням знака інтеграла і похідної – і палітра з'явиться у вікні програми. Потім слід встановити курсор в необхідне місце екрана (там з'явиться шаблон) і клацнути в палітрі на значку з зображенням знака визначеного інтеграла.

В складі складних шаблонів часто зустрічаються менші шаблони для введення окремих даних. Вони мають вигляд маленьких чорних прямокутників. Ми будемо називати їх, як і раніше, *місцями введення*. В шаблоні інтеграла їх чотири: для введення верхньої межі інтегрування, для введення нижньої межі інтегрування, для завдання підінтегральної функції і для вказівки імені змінної, за якою відбувається інтегрування. На рис. 2.15 показана робота з шаблонами деяких функцій і операторів.

Заповнення місць введення визначеного інтегралу при обчисленні

$$\int_1^2 dx \quad \int_1^2 e^x dx \quad \int_1^1 e^x dx \quad \int_0^1 e^x dx \quad \int_0^1 e^x dx = 1.718$$

Заповнення шаблону з палітри Evaluate и Calculator

sin() sin(1) sin(1) = 0.841

:= x := x := 2 + 3 x = 5

This expression is incomplete. You must fill in the placeholders.

Заповнення шаблону з палітри Boolean

:=	2 =	2 = 2	2 = 2 = 1
<	3 <	3 < 2	3 < 2 = 0
∧	1 ∧	1 ∧ 0	1 ∧ 0 = 0
∨	1 ∨	1 ∨ 0	1 ∨ 0 = 1

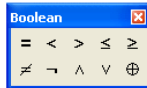
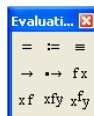
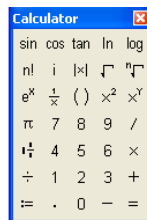
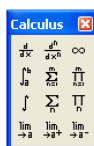


Рис. 2.15. Робота з шаблонами

Аналогічним чином задаються шаблони для обчислення функцій і для застосування операторів. Наприклад, на рис. 2.15 показано застосування логічних операторів з палітри таких операторів Boolean. Зверніть увагу також на виведення спливаючого повідомлення про помилку в центрі екрана. Зазвичай, ознакою помилки є забарвлення невірною виразу у червоний колір. Якщо в такий вираз ввести курсор миші, то з'явиться зазначене повідомлення про помилку.

7. Контрольні запитання

- 7.1. Які види блоків є в системі MathCad?
- 7.2. Як здійснюється введення математичного виразу?
- 7.3. Яким чином здійснюється виведення результату?
- 7.4. Які форми має курсор в системі MathCad? Їх призначення.
- 7.5. Які існують способи виділення блоків?
- 7.6. Опишіть основні способи копіювання окремих фрагментів.
- 7.7. Як організуються текстові області і що в них може входити?
- 7.8. Як можна видалити або перемістити окремі блоки?
- 7.9. Як здійснюється зміна шрифтів для математичних, текстових і графічних областей?
- 7.10. Які є особливості роботи Mathcad при виконанні простих обчислень?

Додаток А

Обчисліть визначений інтеграл, побудуйте таблицю значень і графік підінтегральної функції:

№	$f(x)$	№	$f(x)$
1	$\int_1^3 x^{-1} e^x dx$	11	$\int_0^1 \frac{dx}{\sqrt{x(1+x)}}$
2	$\int_1^2 x^{-2} e^{-2x} dx$	12	$\int_0^1 \frac{dx}{\sqrt{x(3+x)}}$
3	$\int_0^1 \cos x(x^2 + x + 1) dx$	13	$\int_0^1 \frac{dx}{\sqrt{x(1-x)}}$
4	$\int_0^\pi e^{\cos x} \cos 2x dx$	14	$\int_{0.5}^1 \frac{(x+2) dx}{\sqrt{x-1}}$
5	$\int_0^\pi \cos x(x - \sin x) dx$	15	$\int_0^1 \frac{(x+2) dx}{\sqrt{1-x}}$
6	$\int_{-1}^3 x e^{-x} dx$	16	$\int_0^1 \frac{dx}{\sqrt{x^3(3+x^2)}}$
7	$\int_1^2 e^{x^2-x^2} dx$	17	$\int_1^{10} x^{-2} e^{2x} dx$
8	$\int_0^1 \sin(x^2 + x + 1) dx$	18	$\int_0^\pi e^{\sin 2x} \sin 3x dx$
9	$\int_0^\pi e^{\sin^2 x} \sin 3x dx$	19	$\int_{-1}^1 x e^{-6x} dx$
10	$\int_0^1 x^{\frac{3}{2}} e^{-x} dx$	20	$\int_0^1 x^2 e^{-3x} dx$

ПРАКТИЧНА РОБОТА №3

ОБЧИСЛЕННЯ МАТРИЦЬ

1. Мета роботи

Навчитися використовувати вбудовані функції в системі MathCAD 2001i

2. Задачі роботи

Засвоїти прийоми роботи з вбудованими функціями MathCAD 2001i

3. Зміст роботи

3.1 Завантажити MathCAD 2001i.

3.2 Ознайомитися з основними прийомами використання вбудованих функцій.

3.3 Виконати завдання, у відповідності із варіантом. Варіант вибрати згідно списку у журналі.

3.4 Зберегти результати у файлі з введенням необхідних коментарів і оформити звіт.

4. Вимоги до звіту

Звіт повинний включати:

– титульний лист з назвою роботи та відомостями про виконавця;

– колонтитули з П.І.Б. виконавця і шифром групи.

– постановку задачі дослідження, послідовність її виконання;

– результати обчислень згідно варіанту (на дискеті і в роздрукованому вигляді);

– відповіді на контрольні запитання.

5. Рекомендована література

5.1 Дьяконов В.П. MathCAD 2001i и MathCAD 2001i.-М.: СОЛОН Пресс, 2004.-832.

5.2 Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6.1. Використання вбудованих функцій

Довільні залежності між вхідними і вихідними параметрами задаються за допомогою функцій. Функції приймають набір параметрів і повертають значення: скалярне або векторне (матричне).

В формулах можна використовувати стандартні вбудовані функції, а також функції, визначені користувачем.

Щоб використати функцію у виразі, потрібно визначити значення вхідних параметрів у дужках після імені функції. Mathcad має багато вбудованих елементарних, спеціальних і статистичних функцій. Найбільш відомі з них – елементарні – можуть вводитися прямо їх позначеннями, наприклад $\sin(1)$, $\cos(0.5)$, $\text{asin}(0.5)$, $\sinh(1)$, $\ln(2)$ тощо. Інформацію про інші функції можна почерпнути у довідковій системі.

На першому етапі освоєння системи багато користувачів плутаються в позначенні функцій і не мають уявлення, які функції є в системі і як саме їх вводити. Особливо це стосується зворотних функцій: багато хто задає убудовану функцію asin Mathcad, як це прийнято в математиці – \arcsin , і отримує повідомлення про помилку.

Для полегшення введення математичних функцій слугує кнопка **f(x)**, яка виводить вікно „**Вставити функцію**” з повним переліком функцій, які розбиті на тематичні розділи (рис. 3.1).

В лівій частині цього вікна вибирається категорія, до якої відноситься функція, а в правій частині – конкретна функція. В нижній частині вікна видається інформація про обрану функцію. При введенні функцій через це діалогове вікно автоматично додаються дужки і шаблони для значень параметрів.

Обрана функція може бути перенесена у вікно документа клацанням на кнопці внизу вікна з переліком функцій. Функції мають параметри (аргументи), які записуються в круглих дужках після імені функції. Функції можуть мати один параметр (наприклад, **sin (x)** або **cos (0.5)**), два параметра (наприклад, **ln (m,x)**) або багато параметрів. Параметри можуть мати чисельне значення, бути константою, визначеною раніше змінною або математичним виразом, який повертає численне значення.

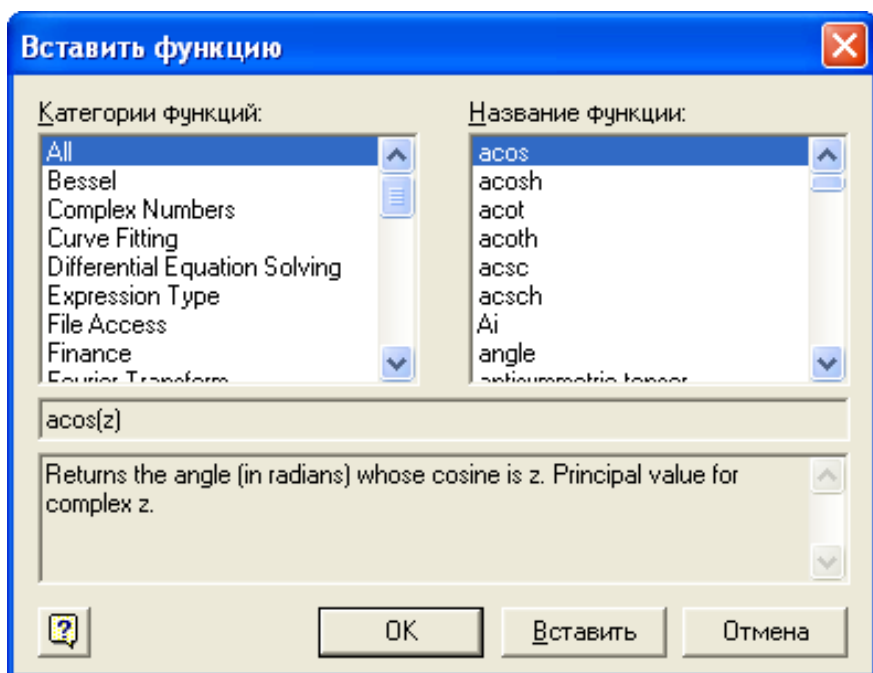


Рис. 3.1. Вікно **Вставить функцию**

Оскільки функції мають властивість повертати результат, то їх можна використовувати в складних математичних виразах, наприклад:

$$(2 + 3i) \cdot \sin(3 \cdot e^{-1})$$

Відзначимо, що i в цьому виразі є мнима одиниця (квадратний корінь з -1), і те що більшість функцій може мати комплексні аргументи і повертати комплексні значення. Цікаво, що знак множення у мнимій частині комплексних чисел Mathcad не потрібний, як це і прийнято при написанні виразів у математиці.

Користувальницькі функції повинні бути спочатку визначені. Визначення задається за допомогою оператора присвоювання. У лівій частині вказується ім'я користувальницької функції, а в дужках формальні параметри – змінні, від яких вона залежить. Праворуч від знака присвоювання ці змінні повинні використовуватись у виразі. При використанні користувальницької функції в наступних формулах їх вводять вручну. В діалоговому вікні „**Вставить функцию**” вона не відображується.

6.2. Застосування ранжируваних змінних і векторів

До цього моменту ми обходилися добре відомими типами змінних, які зберігають значення звичайних чисел (цілих, дійсних). Mathcad припускає для таких змінних і комплексні значення, наприклад $2 + 3i$. На відміну від мов програмування Mathcad не вимагає точного завдання типів звичайних змінних, наприклад таких, як цілочислові змінні одинарної або подвійної точності, речовинні змінні з плаваючою крапкою тощо. Тип змінної автоматично визначається привласненням їй значенням.

Однак є і спеціальні типи змінних, які називають ранжируемими. Вони мають множинні значення. Наприклад, якщо записати $n := 1..5$, то змінна n буде представляти цілі числа від 1 до 5 з кроком 1, тобто значення 1, 2, 3, 4, 5. Можливість доступу окремо до кожного значення відсутня.

Примітка. Для введення знака $:=$ треба натиснути клавішу $:$ (двокрапка), а для введення знака $..$ (дві крапки) – клавішу $;$ (крапка с комою).

Якщо потрібно задати ряд чисел з кроком d , то ранжирувана змінна записується наступним чином:

$$X := Xstart, Xstart + d .. Xend$$

Тут $Xstart$ – початкове значення змінної X , $Xend$ – кінцеве значення змінної X . Наприклад, $x := 1,1.25..2$ дає ранжовану змінну X зі значеннями 1, 1.25, 1.5, 1.75 і 2.

Примітка. Зверніть увагу на те, що другий член при завданні ранжируваної змінної – це сума її початкового значення з приростом d . Можна задати приріст від'ємним, але у цьому випадку значення $Xstart$ повинно бути більше, чим $Xend$. В явному вигляді приріст зазвичай не задається і фігурує як різниця чисел, розділених комами, наприклад в нашому прикладі $X := 1,1.25..2$ приріст є $1.25 - 1 = 0.25$.

На перший погляд ранжирувана зміна нагадує вектор – одномірний масив індексованих змінних. Наприклад, вектор x з трьох елементів 1, 1.25 і 1.5 може бути представлений трьома компонентами X_0 , X_1 , і X_2 із вказаними значеннями. Індекси можуть починатися з 0 або 1. Ці значення задає спеціальна системна мінлива **ORIGIN**, яка за замовчанням має нульове значення. Якщо змінна не є вектором, то задавати індекси неприпустимо, і Mathcad буде відзначати це як помилку, виділяючи вираз червоним кольором і супроводжуючи його повідомленням про помилку у вигляді спливаючої підказки.

На рис. 3.2 представлена низка прикладів завдання і виводу ранжированих змінних і вектора V.

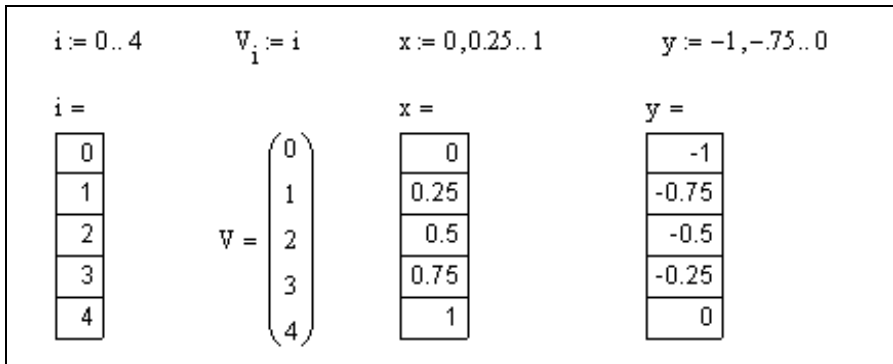


Рис. 3.2. Приклади завдання і виведення ранжированих змінних і вектора V

Як видно з рис.3.2, таблиця значень ранжированої змінної формується під оператором її виводу, а таблиця значень елементів вектора – *праворуч* від оператора виводу. Це підкреслює відмінність ранжированої змінної від вектора. До того ж вектор подається в округлених дужках, тоді як таблиця значень ранжированої змінної прямокутна. Крім того, вектор дає доступ до будь-якого свого елемента, тоді як ранжирована змінна такого доступу не забезпечує – спроба доступу супроводжується повідомленням про помилку.

Примітка. Ранжирована змінна не надає повноцінного засобу заміни циклів, звичай реалізованих за допомогою мов програмування. Для створення циклів й інших традиційних програмних конструкцій Mathcad має спеціальні засоби.

6.3. Введення матриць і векторів

Матриці у вигляді двовимірних масивів широко застосовуються при розв'язанні задач механіки, електротехніки, радіотехніки й інших галузей науки і техніки. Вектори і матриці розглядаються в програмі MathCad як одномірні та двомірні масиви даних. Кількість рядків і стовпців матриці задається в діалоговому режимі **Вставити матрицу**. Вектор задається як матриця, що має один стовпець.

Матрицю можна представити як таблицю, що має m рядків (rows) і n стовпців (columns). Якщо $m = n$, то матрицю називають квадратною. Кількість елементів або розмір матриці є $m \times n$. Вектор довжиною m розглядається як одномірна матриця розміру $m \times 1$. Для операцій з матрицями, включаючи їх введення, служить палітра матричних операцій Matrix (Матриці) (рис. 3.3).



Рис. 3.3. Палітра матричних операцій

В ній, зокрема, є шаблон для введення матриць, який має вигляд витягнутих округлених дужок з місцями для введення елементів матриць. При введенні шаблону з'являється невеличке діалогове вікно, яке запитує необхідну кількість рядків і стовпців матриці (рис. 3.4).

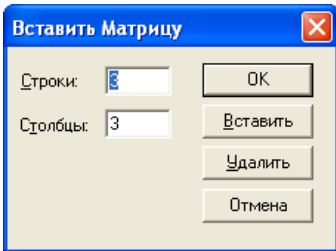


Рис. 3.4. Діалогове вікно **Вставити матрицу**

Після клацання на кнопці **OK** в формулу вставляється матриця, яка містить замість елементів матриці шаблони. Замість кожного заповнювача потрібно ввести число, змінну або вираз.

Для матриць визначені наступні операції: додавання, множення на число, множення тощо. Припустиме також використання матриць замість скалярних виразів: у цьому випадку передбачається, що вказані дії повинні бути застосовані до кожного елементу матриці, і результат також буде представлений у вигляді матриці. Наприклад, вираз $M+3$, де M – матриця, означає, що до кожного елемента матриці додається число 3. Якщо потрібно явно вказати необхідність поелементного використання операції до матриці, використовують знак векторизації, для введення якого слугує кнопка **Векторизация** на панелі інструментів **Матрица**.

Для роботи з елементами матриці використовують індекси елементів. Нумерація рядків і стовпців починається з нуля. Індекс елемента задається числом, змінною або виразом і відображується як нижній індекс. Елементи матриць є індексованими змінними і характеризуються двома індексами:

номером елемента в рядку i і номером елемента в стовпці. Наприклад, $M_{i,j}$ означає елемент матриці, розташований в рядку i і стовпці j . Пара індексів, яка визначає елемент матриці, розділяється комою.

При побудові графіків іноді потрібно виділити вектор, який представляє собою стовпець матриці. Номер стовпця матриці відображається як верхній індекс, поміщений в кутові дужки, наприклад $M^{<0>}$. Для його введення використовується кнопка **Столбец** на панелі інструментів **Матрица**.

Матриці одного розміру можна складати і віднімати, можлива заміна рядків на стовпці – операція транспонування (значок M^T в палітрі, представлена на рис.3.3). Інші важливі матричні операції, наприклад, обернення матриці і розв'язок матричних рівнянь, будуть розглянуті пізніше.

Елементи векторів і матриць можна задавати явно, присвоюючи індексованій змінній те або інше значення. Аналогічно можна отримувати або виводити значення потрібного елемента.

Рис. 3.5 ілюструє процедуру введення матриці і використання найпростіших векторних і матричних операцій.

ШАБЛони МАТРИЦЬ ТА ЇХ ЗАСТОСУВАННЯ

ШАБЛони МАТРИЦЬ

$\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$
 $\begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix}$

ПРИКЛАД МАТРИЧНОЇ ОПЕРАЦІЇ

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} 8 & 10 & 12 \\ 14 & 16 & 18 \end{pmatrix}$$

ПРИКЛАД ОПЕРАЦІЇ З ВЕКТОРОМ

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot 10 = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix}$$

ПРИКЛАД ВИВЕДЕННЯ ЕЛЕМЕНТІВ МАТРИЦЬ

$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$
 $M_{0,0} = 1$
 $M_{2,2} = 9$
 $M_{0,2} = 3$

Рис. 3.5. Приклади найпростіших операцій з матрицями

6.4. Заборона обчислення математичних виразів

Математичні вирази можна зробити пасивними. Для цього достатньо виділити вираз і вибрати команду **Disable Evaluation** (Заборонити виконання) в контекстному меню, яке з'являється при клацанні на виразі правою кнопкою миші (рис. 3.5). Вираз буде помічений значком чорного прямокутника і не буде виконуватись ні в автоматичному, ні в ручному режимах обчислень. Такий вираз можна розглядати просто як складний текстовий коментар. Для зняття з виразу статусу не обчислюваного і не виконуваного, достатньо виділити вираз і вибрати команду **Enable Evaluation** (Дозволити виконання) в контекстному меню.

6.5. Переривання обчислень

При завданні складних обчислень робота системи може бути доволі тривалою. Для її переривання можна натиснути клавішу **Esc**. Mathcad виведе повідомлення про переривання обчислень і невелике вікно з двома кнопками; **OK** – підтвердити переривання і **Cancel** (Скасування) – скасувати переривання. Після переривання можна відновити роботу, натиснувши клавішу **F9** або клацнувши на панелі інструментів на кнопці з зображенням жирного знака рівності.

7. Контрольні запитання

- 7.1. Які існують способи вставлення функцій?
- 7.2. Які є складові користувальницької функції?
- 7.3. Які змінні називають ранжируемими?
- 7.4. Як подається вектор – одномірний масив змінних?
- 7.5. Який засіб служить для операцій з матрицями?
- 7.6. Який вигляд має шаблон для введення матриць?
- 7.7. Які існують операції для роботи з матрицями?
- 7.8. Як можна зробити пасивними математичні вирази?
- 7.9. Як можна виконати переривання складних обчислень?
- 7.10. Для чого використовують індекси елементів?

8. Завдання для практичної роботи №3:

8.1. Знайти матрицю С – суму матриць А і В.

8.2. Транспонувати матрицю С.

8.3. Знайти матрицю D, помноживши матрицю А на коефіцієнт а.

8.4. Помножити матрицю D на транспоновану матрицю С.
(Порядок важливий, оскільки закон перестановки не діє).

Варіант	Матриця А	Матриця В	Коефіцієнт а
1	$\begin{pmatrix} 44 & 746 \\ -654 & 654 \\ 6 & 45 \end{pmatrix}$	$\begin{pmatrix} 5 & 8 \\ -54 & 54 \\ 74 & 46 \end{pmatrix}$	10
2	$\begin{pmatrix} -54 & 654 \\ 8 & 28 \\ 6 & -4 \\ 648 & 22 \end{pmatrix}$	$\begin{pmatrix} 56 & 2 \\ 121 & -72 \\ -21 & 21 \\ 21 & 231 \end{pmatrix}$	-25
3	$\begin{pmatrix} 564 & 55 & 564 \\ 544 & 22 & -422 \\ 544 & -75 & 64 \\ -103 & 341 & 644 \end{pmatrix}$	$\begin{pmatrix} 54 & 89 & 62 \\ -354 & 218 & 31 \\ 34 & -32 & 21 \\ 86 & 24 & 54 \end{pmatrix}$	0.14
4	$\begin{pmatrix} 59 & 26 & 47 & 25 \\ -344 & -32 & 5 & 84 \\ 3 & -30 & 25 & 30 \\ 1 & -503 & -65 & 52 \end{pmatrix}$	$\begin{pmatrix} 54 & -877 & 32 & 5 \\ 78 & 23 & -20 & 564 \\ 245 & -16 & 6 & 30 \\ 1 & -503 & -65 & 52 \end{pmatrix}$	2.1
5	$\begin{pmatrix} 45 & -31 & -31 & 89 \\ 24 & 358 & -31 & 2 \\ -534 & 64 & 4 & 6 \end{pmatrix}$	$\begin{pmatrix} 45 & -8 & 69 & 32 \\ -14 & 85 & -24 & -8 \\ -33 & -24 & 94 & 90 \end{pmatrix}$	1.25
6	$\begin{pmatrix} 45 & 0 & 21 & 0 \\ 1 & -31 & 0 & -58 \end{pmatrix}$	$\begin{pmatrix} 4 & 23 & 0 & -54 \\ -45 & 1 & 0 & -1 \end{pmatrix}$	3.14
7	$\begin{pmatrix} 5 & -36 & 45 \\ 56 & 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 21 & -12 & 0 \\ 1 & -5 & 3 \end{pmatrix}$	5
8	$\begin{pmatrix} 1 & -25 \\ 0 & 0.258 \end{pmatrix}$	$\begin{pmatrix} -0.12 & 3 \\ 0 & 125 \end{pmatrix}$	-12
9	$\begin{pmatrix} 45 & -54 & 0 & -1 & -25 & 36 \\ 47 & 56 & 21 & 37 & 19 & 91 \end{pmatrix}$	$\begin{pmatrix} 5 & 78 & 91 & 2 & 321 \\ 366 & -5 & -88 & 0 & 14 \end{pmatrix}$	-45
10	$\begin{pmatrix} 51 & 23 & 2 & 87 & -5 \\ 0 & 25 & -44 & 31 & 71 \\ 88 & 65 & 74 & 91 & 4 \end{pmatrix}$	$\begin{pmatrix} 97 & 74 & 56 & -11 & -55 \\ 4 & -512 & 63 & 14 & 55 \\ 11 & -222 & 7 & 13 & -74 \end{pmatrix}$	69

11	$\begin{pmatrix} 54 & 76 & 354 & 5 & 45 \\ -12 & 5 & 98 & -10 & 0 \\ 120 & -1 & -23 & 87 & 13 \\ 78 & 45 & 62 & -145 & 25 \end{pmatrix}$	$\begin{pmatrix} 87 & 4 & 56 & 33 & -125 \\ -768 & 112 & 1 & 36 & 77 \\ 14 & 256 & 45 & 89 & 4 \\ 8 & 10 & 35 & -14 & -21 \end{pmatrix}$	-13
12	$\begin{pmatrix} 54 & 7 & -8 \\ 99 & 6 & 5 \\ -21 & 31 & 45 \end{pmatrix}$	$\begin{pmatrix} 54 & 21 & -36 \\ 25 & -75 & 31 \\ 47 & 221 & 0 \end{pmatrix}$	12.5
13	$\begin{pmatrix} 54 & 55 & 69 & 84 & 553 \\ -23 & 47 & 0 & 21 & 14 \\ 7 & 12 & 66 & 7 & 44 \\ 44 & 56 & 213 & -222 & 47 \\ 76 & -23 & 47 & 69 & -55 \end{pmatrix}$	$\begin{pmatrix} -47 & 85 & 42 & 146 & 256 \\ 128 & 64 & 32 & 16 & 8 \\ 4 & 2 & 1024 & 1280 & -54 \\ -56 & -4 & -532 & 21 & 0 \\ 0 & 12 & 25 & 87 & 141 \end{pmatrix}$	21
14	$\begin{pmatrix} 22 & -16 & 4 & 2 \\ 2 & 41 & -41 & -1 \\ 0 & 0 & 0 & 0 \\ 5 & 23 & 87 & -42 \\ 64 & 88 & 9 & 11 \end{pmatrix}$	$\begin{pmatrix} 78 & 9 & 25 & 4 \\ 12 & -22 & 34 & 588 \\ 125 & -10 & 0 & 58 \\ 21 & 0 & 144 & 26 \\ 87 & 14 & -13 & 87 \end{pmatrix}$	15
15	$\begin{pmatrix} 5 & -3 & 1 \\ 14 & 58 & 963 \\ -54 & 76 & -12 \\ 0 & 0 & 0 \\ 14 & 45 & 63 \end{pmatrix}$	$\begin{pmatrix} 14.5 & 18 & -13 \\ 23 & -36 & 98 \\ 32 & 0.1 & 21 \\ -9 & -33 & 0 \\ 87 & 1 & 12 \end{pmatrix}$	0.25
16	$\begin{pmatrix} 1 & -22 \\ 12 & 56 \\ 0 & -4 \\ 0 & 1 \\ 141 & 12 \end{pmatrix}$	$\begin{pmatrix} 54 & -66 \\ 21 & 54 \\ -78 & 22 \\ 0 & 0 \\ 1 & 4 \end{pmatrix}$	0.5
17	$\begin{pmatrix} 44 & 746 \\ -64 & 4 \\ 99 & 5 \end{pmatrix}$	$\begin{pmatrix} -5 & 8 \\ 4 & 0 \\ 74 & 21 \end{pmatrix}$	2
18	$\begin{pmatrix} 4 & 77 \\ -8 & 56 \\ 6 & -4 \\ 6 & 3 \end{pmatrix}$	$\begin{pmatrix} 56 & 5 \\ 0 & -72 \\ -21 & 12 \\ 21 & 0 \end{pmatrix}$	3
19	$\begin{pmatrix} 42 & 55 & 88 \\ 36 & 22 & -1 \\ 54 & -75 & 64 \\ -3 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 4 & 8 & 62 \\ -1 & 2 & 31 \\ 34 & -32 & 21 \\ 8 & 2 & 54 \end{pmatrix}$	9
20	$\begin{pmatrix} -5 & -1 & -3 & -9 \\ 0 & 8 & -31 & 2 \\ -534 & 0 & 4 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 18 & 69 & 32 \\ 0 & -85 & 4 & 8 \\ -33 & 0 & 94 & 90 \end{pmatrix}$	7

ПРАКТИЧНА РОБОТА № 4

ПОБУДОВА ГРАФІКІВ В MATHCAD 2001i

1. Мета роботи

Навчитися будувати графіки в системі MathCAD 2001i

2. Задачі роботи

Оволодіти прийомами побудови графіків в MathCAD 2001i

3. Зміст роботи

3.1. Завантажити MathCAD 2001i.

3.2. Ознайомитися з основними прийомами побудови графіків.

3.3. Зберегти результати у файлі з введенням необхідних коментарів і оформити звіт.

4. Вимоги до звіту

Звіт повинний включати:

– титульний лист з назвою роботи та відомостями про виконавця;

– колонтитули з П.І.Б. виконавця і шифром групи.

– постановку задачі дослідження, послідовність її виконання;

– результати обчислень згідно варіанту (на дискеті і в роздрукованому вигляді);

– відповіді на контрольні запитання.

5. Рекомендована література

5.1. Дьяконов В.П. MathCAD 2001i и MathCAD 20011.-М.: СОЛОН Пресс, 2004.-832 с.

5.2. Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Побудова графіків

Для створення графіків в системі Mathcad існує програмний графічний процесор. Основна увага при його розробці була приділена забезпеченню простоти завдання графіків і їх модифікації за допомогою відповідних параметрів. Процесор дозволяє будувати найрізноманітніші графіки, наприклад в декартовій і полярній системах координат, графіки поверхонь, тривимірні фігури, графіки рівнів тощо.

6.1. Види графіків

Графіки в системі Mathcad є такими ж об'єктами, як і константи, змінні, формули тощо. В меню **Вставка** в підменю **Графік** є список наступних шаблонів графіки (рис. 4.1):

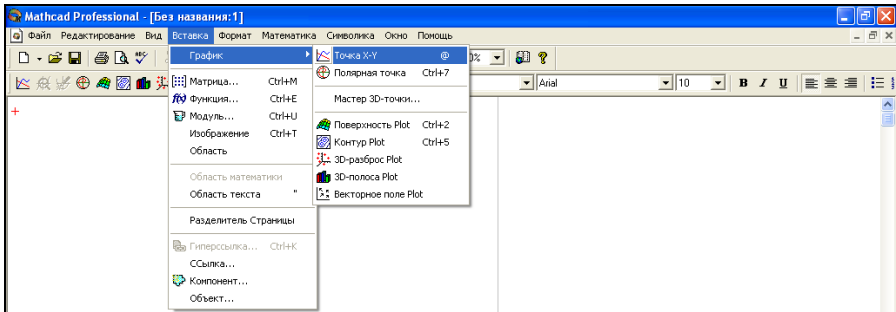


Рис. 4.1. Вікно шаблонів графіки

- **Точка X-Y** (Графік функції однієї змінної в системі декартовій системі координат);
- **Полярная точка** (Графік в полярній системі координат);
- **Мастер 3D точки** (Майстер тривимірної графіки);
- **Поверхность Plot** (Графік поверхні – функції двох змінних);
- **Контур Plot** (Контурний графік – графік ліній одного рівня);
- **3D Разброс Plot** (Точковий графік – графік поверхні, побудований точками);
- **3D Полоса Plot** (Тривимірна гістограма);
- **Векторное поле Plot**.

6.1.1. Графік функції однієї змінної (двовимірний графік)

Графіки в системі Mathcad можуть мати різні розміри і можуть переміщатися у вікні редагування документа. Для виводу шаблону двовимірної графіки в декартовій системі координат слугує команда **Точка X-Y** (декартів графік) або клавіша **@**. Вона виводить в місце, де знаходиться курсор, шаблон двовимірного графіка. – звичайний графік на площині з уявленими (або дійсно намальованими) (горизонтальною – **X**, і вертикальною –**Y**) осями, які розташовані під прямим кутом одна до одної.

Кожна точка графіка в декартовій системі координат характеризується своїми координатами **x** і **y=f(x)**, де **x** – абсциса точки, а **y** – її ордината. Точки з'єднуються одна з одною різними лініями (суцільною, пунктирною тощо.). Можуть бути показані вихідні (вузлові) точки графіка у вигляді жирних точок, квадратиків, тощо. Можлива також побудова на одному рисунку графіків декількох функцій.

Незаповнений шаблон графіка представляє собою великий порожній прямокутник з шаблонами місць введення даних у вигляді темних маленьких прямокутників, розташованих біля осей абсцис і ординат майбутнього графіка. В них потрібно ввести вирази, які задають координати точок графіка за осям **X** і **Y**. У загальному випадку, це можуть бути функції певної змінної **x**.

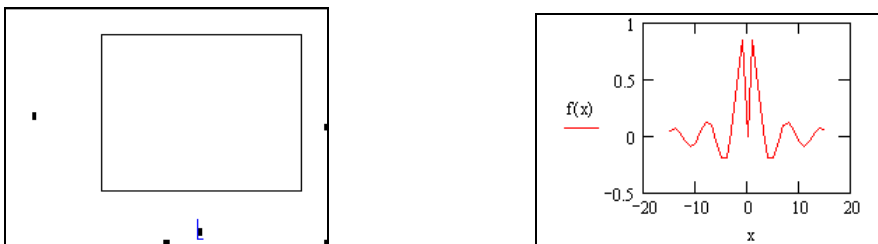


Рис. 4.2. Шаблон графіка і його зображення

Зазвичай використовують діапазон або вектор значень. Граничні значення за осями обираються автоматично у відповідності з діапазоном зміни величини, однак їх можна задавати і вручну.

В одній графічній області можна побудувати декілька графіків. Якщо будуються графіки декількох функцій в одному шаблоні, то функції необхідно розділяти комами. Для виділення даних в шаблоні зручно використовувати клавіші переміщення курсору. Дані в шаблоні також можна виділити за допомогою миші.

Якщо графік вже побудований, то при його виділенні з'являються крайні місця введення з автоматично введеними числами, які слугують для вказівки граничних значень абсцис і ординат, тобто задають масштаб графіка. В принципі можна не заповнювати ці місця введення самостійно але тоді масштаб, скоріше за все, виявиться незручним, наприклад, буде представлений не округленими десятковими числами. Незважаючи на це, рекомендується завжди спочатку використовувати автоматичне масштабування і лише потім обирати більш придатний масштаб.

Різні криві зображуються різним кольором, а для форматування графіка потрібно двічі клацнути на області графіка. Для управління відображенням побудованих ліній слугує вкладка **Линии**, що відкривається у діалоговому вікні. Поточний формат кожної лінії наведений у списку, а під списком розташовані елементи управління, які дозволяють змінити формат.

Особливості побудови графіків функції однієї змінної

Для найбільш поширених графіків в декартовій системі координат Mathcad передбачає два способи побудови графіків функцій однієї змінної **$f(x)$** :

- спрощений спосіб без завдання ранжируваної змінної **x** (межі зміни **x** автоматично задаються від -10 до $+10$);
- звичайний спосіб з завданням ранжируваної змінної **x** .

При спрощеному способі побудови графіків залишиться ввести **x** в місце введення горизонтальної осі і відвівши покажчик миші в сторону, клацнути лівою кнопкою, отримати готовий графік. Аналогічно можна будувати на одному малюнку і графіки декількох функцій – просто описати їх на вертикальній осі, використовуючи коми для розподілу функцій. Графіки будуть побудовані лініями різного типу і кольору.

При звичайному способі побудови графіків необхідно ввести саму функцію і інтервал зміни її аргументу (наприклад, **x**). Однак, прості функції, якщо вони в подальшому не використовуються, можна вказати безпосередньо в шаблоні

графіка. Далі треба приблизно намітити місце верхнього лівого кута графіка і встановити на це місце курсор. Потім вибрати команду **X-Y Plot** (Декартів графік) в підменю **Вставка** ➤ **Графік** або натиснути клавішу **@**.

Починаючі користувачі зазвичай задають ранжирувану змінну x цілочисловою, наприклад $x := -15..15$. При цьому вони забувають, що в даному випадку графік задається невеликим числом точних цілочислових значень x : $-15, -14, -13, \dots, -1, 0, 1, 2, 3, \dots, 14, 15$. В деяких випадках (рис. 4.3) це призведе до грубого спотворення форми графіків.

В лівій частині рисунка побудовані графіки функції $\sin(x)/x$, що має переборну особливість типу $0/0 \rightarrow 1$ у точці $x = 0$. Однак система Mathcad веде себе в даному випадку як типова система для чисельних розрахунків — поза залежністю від значення знаменника дроби вона вважає значення дроби рівним нулю, якщо чисельник дорівнює нулю. При зазначеному завданні змінної x в точці $x = 0$ отримуємо значення функції $f_1(x) = 0$, що і дає провал першого графіка в точці $x = 0$.

Примітка. В Mathcad 11 введена нова функція $\text{sine}(z) = \sin(z)/z$ яка враховує переборну особливість у точці $z=0$ і може використовуватися з комплексним аргументом.

Графіки функції **$\sin(45/x)$** , показані у правій частині рис.4.3, при $x \rightarrow 0$ дають різке підвищення частоти синусоїди. При недостатньо малому кроці зміни x , яка дорівнює 1, особливість цієї функції практично непомітна.

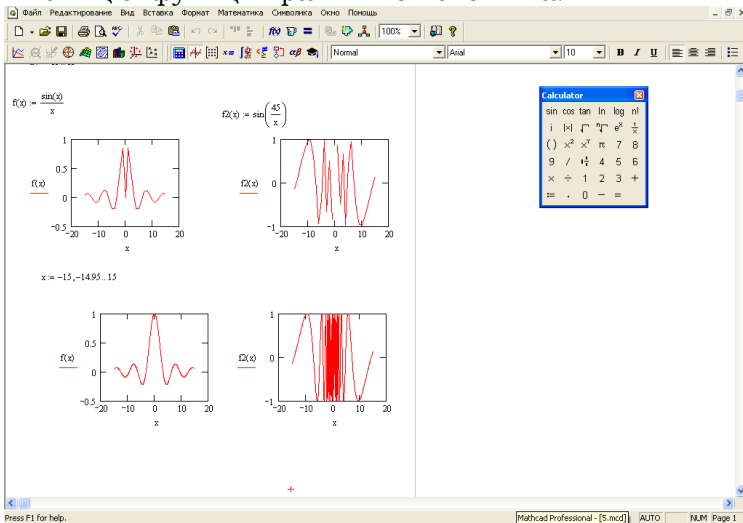


Рис.4.3. Приклади побудови графіків

Нагадаємо, що для спрощеної побудови двовимірного графіка певної функції $f(x)$ потрібно ввести вираз для правої частини цієї функції, відмітити його курсором введення (синім різком) і потім вивести шаблон двовимірного графіка.

Форматування двохмірних графіків

Якщо в побудованих графіках є щось, що не влаштовує користувача, то можна застосувати команди зміни формату графіків. Вікно завдання форматів графіків з'являється, якщо двічі клацнути мишею на графіку або клацнути один раз, якщо графік виділений.

Графіки можна переміщувати вздовж поля вікна документа та змінювати у розмірах. Для цього потрібно виділити графік, клацнувши на ньому лівою кнопкою миші.

Побудова графіків поверхонь

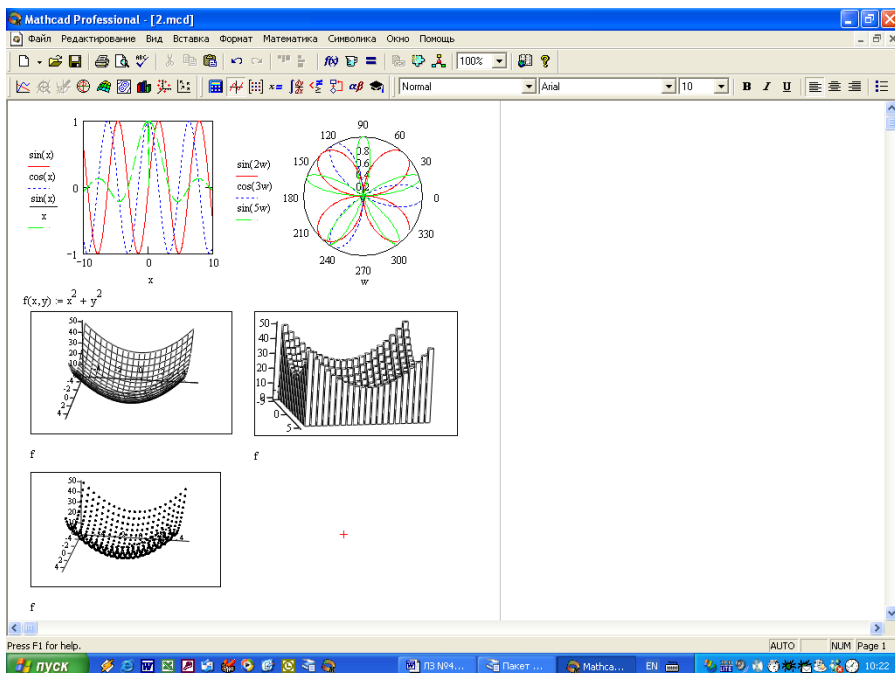


Рис.4.4. Приклади побудови поверхонь

7. Контрольні запитання

- 7.1. Назвіть способи побудови графіків і таблиць результатів.
- 7.2. Як побудувати декілька графіків на одному рисунку?
- 7.3. Як вводяться дані в незаповнений шаблон графіка?
- 7.4. Як можна здійснити форматування графіка?
- 7.5. Як можна побудувати графік однієї змінної?
- 7.6. Як здійснюється форматування двохмірних графіків?
- 7.7. Як будується графік поверхні?

Варіанти завдань для практичної роботи №4

Варіант	Побудувати графік функції	Варіант	Побудувати графік функції
1	$\sqrt[3]{27-(3x)^2-5y^2}$	11	$-x^2 \cdot 3y^2 + 99$
2	$\frac{1}{\sqrt[3]{27-x^2-y^2}}$	12	$2\cos(x)^2 \cdot y^2$
3	$-x^2 - 3y^2 + 32xy$	13	$\sqrt[3]{87x-5y^3}$
4	$25\cos(x)^2 \cdot \sin(y)^5$	14	$\frac{y^2 + 5x^3}{\sqrt[3]{7-x^3-y^2}}$
5	$\sqrt[3]{27-x^2-y^2}$	15	$-x^2 \cdot 3y^2 + 99xy$
6	$\frac{y^2}{\sqrt[3]{27-x^2-y^2}}$	16	$\tan(3y^2) + x^2 y$
7	$-x^2 + 3y^2$	17	$\sqrt[5]{87x^4-5y^3}$
8	$\cos(x^2) \cdot y^2$	18	$\frac{y^2 + 5x^3}{\sqrt[3]{7-y^2}}$
9	$\sqrt[3]{87x+(3x)^2-5y^2}$	19	$3y^2 + x^2 y$
10	$\frac{y^2+5x}{\sqrt[3]{27-x^2-y^2}}$	20	$\tan(3y^2) + \cos(x^2) \cdot y$

ПРАКТИЧНА РОБОТА № 5

СИМВОЛЬНІ ОБЧИСЛЕННЯ В MathCAD

1. Мета роботи

Вивчення можливостей символьних перетворень в системі MathCad Professional

2. Задачі роботи

Освоїти прийоми роботи з системою MathCad Professional

3. Зміст роботи

3.1. Запустити додаток MathCad Professional.

3.2. Освоїти роботу з розділом **Символи**. По черзі включити декілька пунктів даного розділу, ознайомитися з принципом їх розташування і виклику їх опцій.

3.3. Повторити всі наведені у вказівках приклади.

3.4. Обчислити невизначений інтеграл від заданої викладачем функції двома способами (через пункти **Упростить** і **Переменные**) (додаток А) і порівняти результати.

3.5. Знайдіть похідні від заданої викладачем функції двома способами (через пункти **Упростить** і **Переменные**) (додаток Б, В) і порівняйте результати.

3.6. Зберегти результати у файлі з введенням необхідних коментарів і оформити звіт.

4. Вимоги до звіту

Звіт повинний включати:

- титульний лист з назвою роботи та відомостями про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи.
- постановку задачі дослідження, послідовність її виконання;
- результати обчислень згідно варіанту (на флешці і в роздрукованому вигляді);
- відповіді на контрольні питання.

5. Рекомендована література

5.1 Очков В.Ф. MathCAD 7 Pro для студентов и инженеров.-М.: Компьютер Пресс, 1998.- 384 с.

5.2 Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Загальні положення

Меню символних засобів „Символы”

За допомогою пункту **Символы** головного меню викликається спадаюче меню символних засобів (рис. 5.1), з них частина містить свої підменю (рис. 5.2).

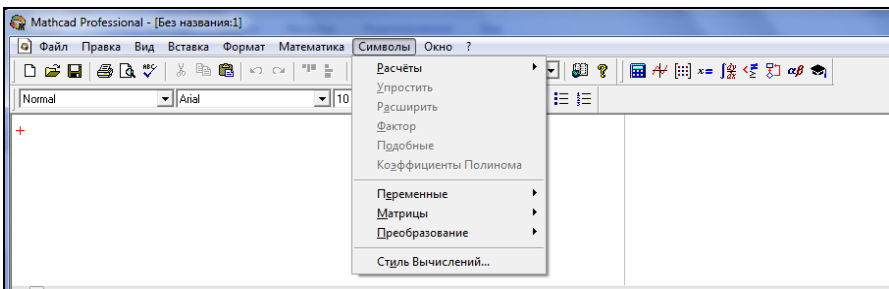


Рис. 5.1. Меню символних засобів MathCAD

Зміст представлених функцій достатньо очевидний. Розглянемо деякі з них. Щоб символні операції виконувалися, необхідно вказати над яким виразом ці операції повинні виконуватись, тобто треба виділити вираз (правила виділення описувалися вище). Потім обрати в головному меню **Символы** і в його підменю необхідну операцію (рис. 5.1). Для низки операцій слід не тільки вказати вираз до якого вони відносяться, але й намітити змінну, відносно якої виконується та або інша символна операція. Сам вираз у цьому випадку не виділяється, адже і так ясно, що якщо маркер введення виділяє змінну певного виразу, то цей вираз вже відзначений наявністю у ньому виділеної змінної.

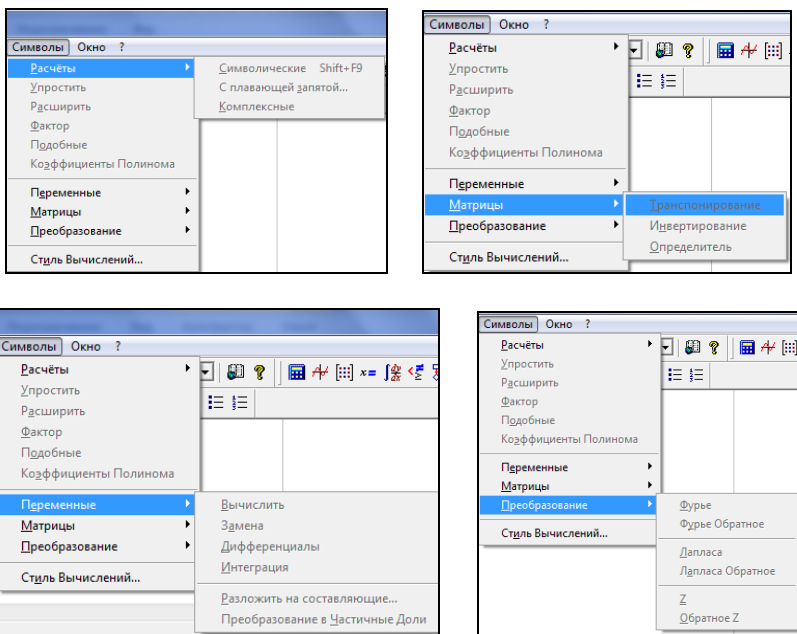


Рис. 5.2. Символьні засоби MathCAD

Символы/Упростить – спростити виділений вираз з виконанням таких операцій, як скорочення подібних доданків, зведення до спільного знаменника, використання основних тригонометричних тотожностей тощо. Дана операція дозволяє спростувати математичні вирази, які містять алгебраїчні тригонометричні функції, а також вирази із степеневими багаточленами (поліномами). За його допомогою можна спростити складні і погано впорядковані алгебраїчні вирази.

На рис. 5.3 показано, як можна спростити аналітичний вираз, розрахувати похідну, обчислити невизначений інтеграл. Якщо система не знаходить розв'язку, то повторює введений вираз або повідомляє про помилку.

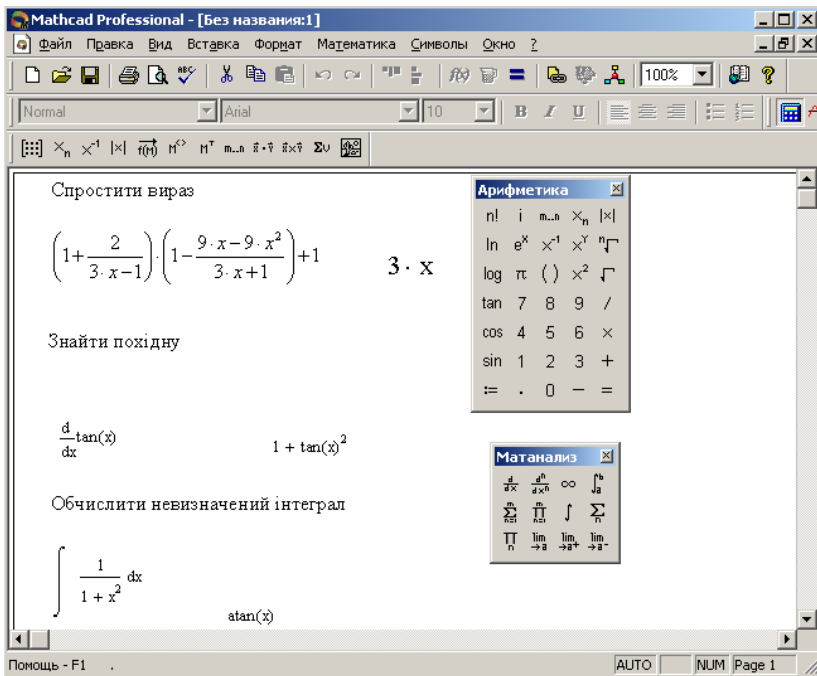


Рис. 5.3. Приклад спрощення виразу

Приклад: встановіть режим відображення обчислень по горизонталі. Для цього клацніть по рядку **Стиль вычислений** в меню **Символы** і встановіть відповідні мітки у вікні діалогу. Для того щоб ввести вираз, клацніть лівою клавiшею миші по вільному місцю в робочому документі і введіть вираз з клавіатури. Спочатку введіть перший співмножник – натисніть на панелі **Арифметика** кнопку (), потім клавiші в наступній послідовності:

<1><+><2>кнопка на панелі **Арифметика** / <3><*><x><-><1>.

Перед тим, як вводити знак множення і другий співмножник, натисніть декілька разів клавiшу **Пробіл** до тих пір, поки весь перший співмножник не буде поміщений у синій куточок (клячку), потім введіть знак множення і аналогічно попередньому другий співмножник тощо. Для того щоб спростити вираз, помістіть його спочатку в клячку, потім клацніть в меню **Символы** по рядку **Упростить**. Результат (перетворений вираз) буде відображений в робочому документі праворуч від вихідного виразу.

Розкладання математичних виразів. Дія пункту меню **Расширить** спадаючого меню **Символы** головного меню MathCad у певному змісті протилежне дії пункту **Упростить**. За допомогою пункту **Расширить** більш складні функції перетворюються через більш прості. Алгебраїчні вирази, представлені у стиснутому вигляді, зводяться до розгорнутих.

Диференціювання математичних виразів. Пункт **Дифференциалы** спадаючого меню **Переменные** (рис. 5.3) дозволяє диференціювати вираз відносно виділеної змінної. Він повертає символічне значення похідної виразу за тією змінною, яка виділена курсором. Для обчислення похідних вищого порядку потрібно повторити обчислення необхідну кількість разів. Нижче наведений фрагмент документа з обчисленням похідної (рис. 5.4).

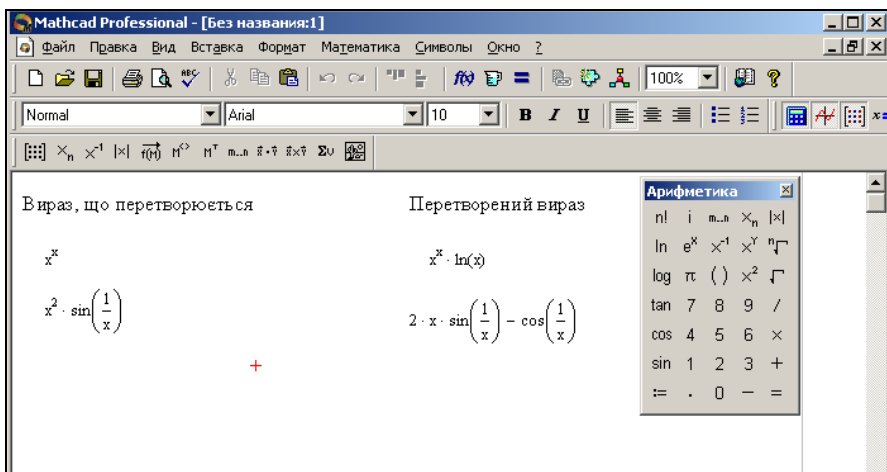


Рис. 5.4. Приклад обчислення похідної

Інтегрування математичних виразів. Пункт **Интеграция** спадаючого меню **Переменные** (рис. 5.2) дозволяє призначений для інтегрування виразів по виділеній змінній. Для необхідного перетворення вводиться тільки підінтегральна функція. Нижче наведений фрагмент документа з обчисленням інтеграла (рис. 5.5).

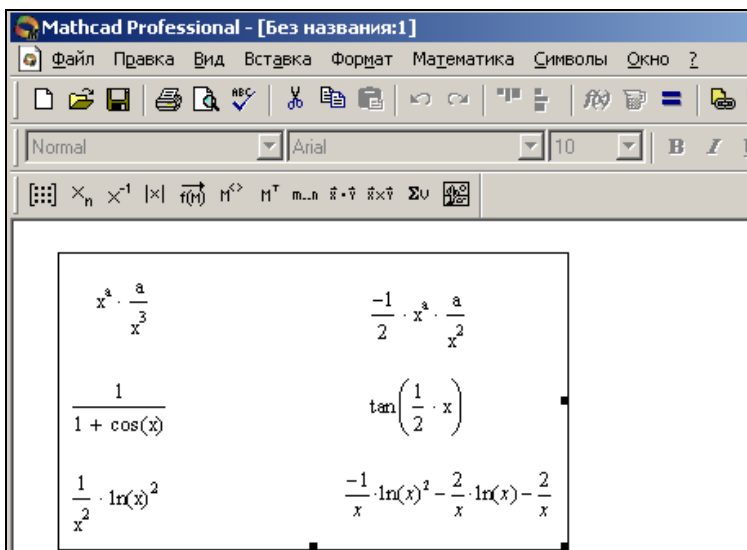


Рис. 5.5. Интегрирования математического выражения

Встановлення стилю виведення результатів обчислень. За допомогою останнього пункту **Стиль вичислень** спадаючого меню викликається однойменне діалогове вікно (рис. 5.6) для установки стилю представлення виразів, над якими виконуються символічні операції.

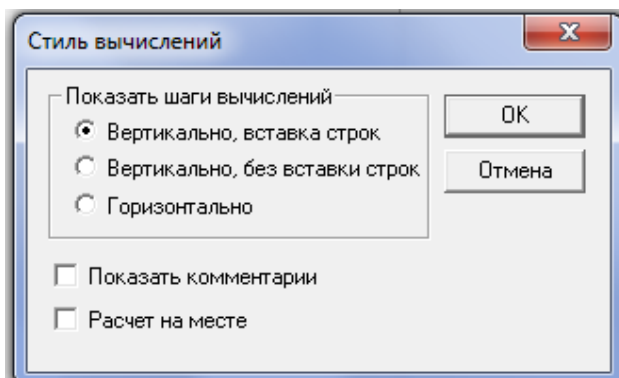


Рис. 5.6. Меню **Стиль обчислень** MathCAD

У цьому вікні можна встановити три типи виводу результату символьних перетворень:

– вертикально, зі вставкою рядків – розташування результату під основним виразом з включенням порожніх рядків праворуч;

– вертикально, без вставки рядків – розташування результату прямо під основним виразом;

– горизонтально – розташування результату поруч (по горизонталі) з основним виразом.

Крім того, встановивши прапорець у прямокутниках, можна ввести ще два режими:

- показати коментарі;

- розрахунок на місці – замінити вихідний вираз результатом його символьного перетворення.

Всі варіанти стиля виводу результатів символьних операцій представлені на рис. 5.7.

Стилі виводу результатів (на прикладі диференціювання)

1. Вертикально зі вставкою рядка	2. Вертикально без вставки рядка
$\frac{\cos(x) - \cos(3 \cdot x)}{x}$	$\frac{\cos(x) - \cos(3 \cdot x)}{x}$
$\frac{(-\sin(x) + 3 \cdot \sin(3 \cdot x))}{x} - \frac{(\cos(x) - \cos(3 \cdot x))}{x^2}$	$\frac{(-\sin(x) + 3 \cdot \sin(3 \cdot x))}{x} - \frac{(\cos(x) - \cos(3 \cdot x))}{x^2}$
3. Горизонтально з показом коментаря	
$\frac{\cos(x) - \cos(3 \cdot x)}{x}$ by differentiation, yields	$\frac{(-\sin(x) + 3 \cdot \sin(3 \cdot x))}{x} - \frac{(\cos(x) - \cos(3 \cdot x))}{x^2}$
4. Горизонтально без коментаря	
$\frac{\cos(x) - \cos(3 \cdot x)}{x}$	$\frac{(-\sin(x) + 3 \cdot \sin(3 \cdot x))}{x} - \frac{(\cos(x) - \cos(3 \cdot x))}{x^2}$

Рис. 5.7. Варіанти стиля виводу результатів символьних операцій

7. Контрольні запитання

- 7.1. Назвіть призначення символічних засобів системи MathCad?
Як до них забезпечується доступ?
- 7.2. В чому полягає відмінність способів диференціювання і інтегрування через підпункти **Упростити** і **Переменные**?
- 7.3. Доповніть можливі у порівнянні з представленими на рис. 5.5 стилі подання результатів обчислень.
- 7.4. Які вимоги необхідно виконати перед виконанням символічних перетворень?
- 7.5. В яких випадках неможливо виконати над наявним математичним виразом ті або інші символічні перетворення і чому? Як про це довідається користувач?
- 7.6. Перерахуйте основні можливості символічних перетворень доступних в системі MathCad.

Додаток А

Обчисліть невизначений інтеграл $\int f(x)dx$

№	$f(x)$	№	$f(x)$
1	$\frac{1}{\sin^2 x(1-\cos x)}$	11	$\frac{1}{(1+\sin x-\cos x)^2}$
2	$\frac{\cos x-\sin x}{(1+\sin x)^2}$	12	$\frac{1+\sin x}{1+\sin x-\cos x}$
3	$\frac{1}{\sin x(1-\sin x)}$	13	$\frac{1+\cos x}{1+\cos x+\sin x}$
4	$\frac{\cos x}{5+4\cos x}$	14	$\frac{\sin x}{1+\cos x+\sin x}$
5	$\frac{\cos x}{1+\sin x-\cos x}$	15	$\frac{\cos x}{1+\cos x+\sin x}$
6	$\frac{\cos x}{2+\sin x}$	16	$\frac{\cos x}{1+\cos x-\sin x}$
7	$\frac{\cos x}{(1-\cos x)^2}$	17	$\frac{\cos 2x-\cos x}{(1+\operatorname{tg}x)^2}$
8	$\frac{1}{\cos x(1-\cos x)}$	18	$\frac{\sin x}{2+\cos x}$
9	$\frac{\sin x}{(1+\sin x)^2}$	19	$\frac{\operatorname{tg}x}{5+\sin x+\cos x}$
10	$\frac{\cos x}{(1+\cos x+\sin x)^2}$	20	$\frac{2x}{2+\sin^2 x}$

Додаток Б

Знайдіть похідну функції $f(x)$

№	$f(x)$	№	$f(x)$
1	$\tan\left(x^3 + x^2 \sin \frac{2}{x}\right)$	11	$\arcsin\left(x^2 \sin \frac{1}{2x}\right)$
2	$\arctan\left(x \cos \frac{1}{5x}\right)$	12	$\cos\left(x^2 \sin \frac{1}{9x}\right)$
3	$\sin\left(x \sin \frac{3}{x}\right)$	13	$\operatorname{arctg}\left(x^2 \sin \frac{1}{3x}\right)$
4	$\arcsin\left(x^2 \cos \frac{1}{9x}\right)$	14	$\cos\left(x \cos \frac{3}{x}\right)$
5	$x + \arcsin\left(x^2 \sin \frac{6}{x}\right)$	15	$\operatorname{tg}\left(x \sin \frac{7}{x}\right)$
6	$\tan\left(2^{x^r \cos \frac{1}{8x}} - 1 + x\right)$	16	$\cos(x \sin^2 x)$
7	$\arctan x \cdot \sin \frac{7}{x}$	17	$\arcsin\left(2x^2 + \cos \frac{1}{9x}\right)$
8	$2x^2 + \operatorname{arctg} x \cos \frac{1}{x}$	18	$\arcsin\left(3x^2 + \operatorname{tg} \frac{1}{9x}\right)$
9	$2x^2 + \arcsin x^2 \cos \frac{1}{x}$	19	$\arccos\left(x^2 + \cos \frac{1}{9x} + \sin \frac{1}{2x}\right)$
10	$2x^2 + \operatorname{tg} x^2 \cos \frac{1}{x}$	20	$\arcsin\left(x^3 + \cos x + \operatorname{tg} \frac{1}{9x}\right)$

Додаток В

Знайдіть часткові похідні $\frac{d^2 f}{dx dy}$, $\frac{d^2 f}{dx^2}$, $\frac{d^2 f}{dx dz}$, $\frac{d^2 f}{dy^2}$, $\frac{d^2 f}{dz^2}$

№	$f(x, y, z)$	№	$f(x, y, z)$
1	$xyz \exp(x+2y+3z)$	11	$xyz^2 \exp(x+2y^2+3z)$
2	$\sin(xyz) \cos(x+2y+3z)$	12	$\ln(x^2 + y^3 + xyz)$
3	$(x^2 - y^2 + z) \sin(x+2y+3z)$	13	$yz \exp(x+2z)$
4	$xy \exp(x+2z)$	14	$(x^2 - y^2 + z) \cos(x+2y+3z)$
5	$\sqrt{x^2 + 2xy + 3xz}$	15	$3^{xy+xz+yz}$
6	$xy^2 \exp(x^2 + 2y + 3z)$	16	$\frac{3xyz + 1}{3x + 2y - z}$
7	$\sqrt{(xyz)^3} \sin \frac{xy}{z}$	17	$\frac{\exp(3xyz + 1)}{3x + 2y - z}$
8	$\exp \frac{xy}{z}$	18	$\frac{3^{xyz+1}}{3x + 2y - z}$
9	$\sqrt{x^2 + 2 \sin y + 3z^3}$	19	$\frac{\ln(3xyz + 1)}{3x + 2y - z}$
10	$\sqrt{x^2 + 2xyz + 3z^3}$	20	$\frac{3xyz + 1}{(\ln 3x + 2y)}$

ПРАКТИЧНА РОБОТА № 6

РОЗВ'ЯЗУВАННЯ РІВНЯНЬ В СИСТЕМІ MathCAD 2001i

1. Мета роботи

Вивчення можливостей розв'язання різних типів рівнянь та їх систем в системі MathCAD 2001i

2. Задачі роботи

Освоїти способи розв'язування різних типів рівнянь та їх систем аналітичними і чисельними методами в середовищі MathCad 2001i

3. Зміст роботи

- 3.1 Запустити додаток MathCad 2001i.
- 3.2 Освоїти способи розв'язування рівнянь в аналітичній формі.
- 3.3 Повторити всі наведені у вказівках приклади.
- 3.4 Розв'язати нелінійне рівняння (Додаток А) чисельно й графічно, порівняти результати.
- 3.5 Розв'язати систему нелінійних рівнянь (Додаток Б).
- 3.6 Розв'язати систему лінійних рівнянь (Додаток В).
- 3.7 Збережіть результати у файлі з введенням необхідних коментарів й оформіть звіт.

4. Вимоги до звіту

Звіт повинний включати:

- титульний лист з назвою роботи та відомостями про виконавця;
- колонтитули з П.І.Б. виконавця і шифром групи.
- постановку задачі дослідження, послідовність її виконання;
- результати обчислень згідно варіанту (на флешці і в роздрукованому вигляді);
- відповіді на контрольні запитання.

5. Рекомендована література

5.1 Очков В.Ф. MathCAD7 Pro для студентов и инженеров.-М.: Компьютер Пресс, 1998.-384 с.

5.2 Плис А.И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Основні положення

Розв'язок рівнянь в аналітичній формі. Система MathCad дозволяє розв'язувати лінійні, нелінійні рівняння і системи рівнянь як аналітично, так чисельно і графічно. Ці функції здійснюються наступним чином. Пункт **Перемінне** меню **Символи** головного меню MathCad викликає спливаюче меню (рис. 6.1).

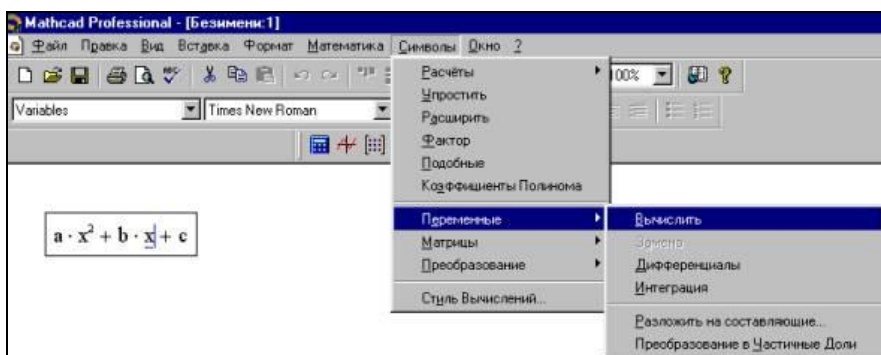


Рис. 6.1. Меню для розв'язку рівнянь в аналітичній формі

За допомогою пункту спливаючого меню **Вычислить** рівняння (нерівність) розв'язується відносно виділеної змінної, тобто здійснюється пошук такого значення змінної, при якому початковий вираз дорівнює нулю. На рис. 6.2 представлений приклад розв'язання квадратного рівняння.

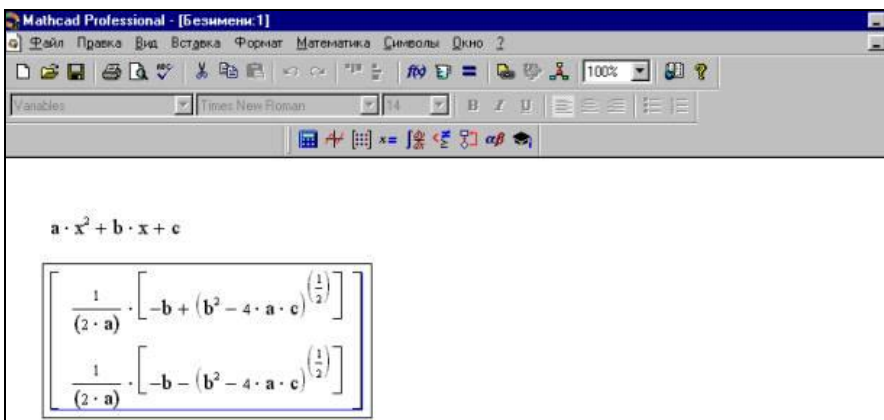
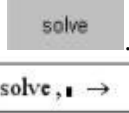


Рис. 6.2 Приклад розв'язання квадратного рівняння

Інший спосіб розв'язання рівнянь можна здійснити, скориставшись панеллю символічних обчислень, на якій необхідно вибрати кнопку



Після чого з'явиться на екрані повідомлення

Введіть у позначеній позиції ліворуч від ключового слова **solve** вираз для правої частини рівняння, а в позиції праворуч від **solve** – ім'я змінної, відносно якої потрібно розв'язати рівняння і клацніть на вільному місці в робочому документі. Результат (значення кореня рівняння) буде відображений в робочому документі праворуч від стрілки (рис. 6.3).

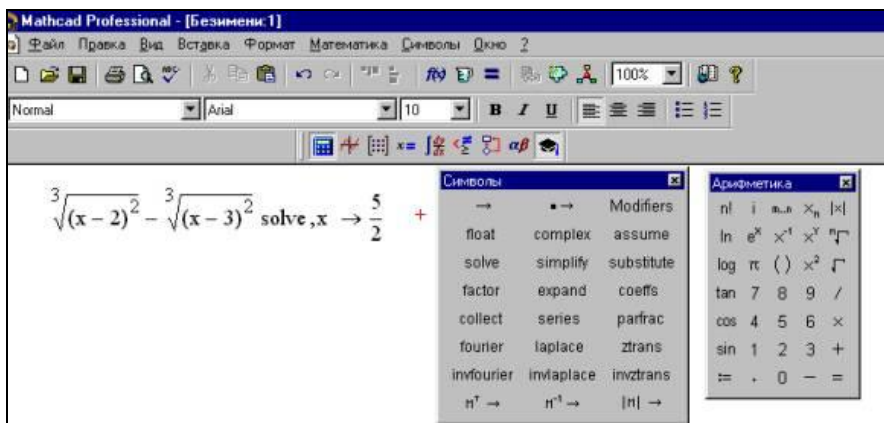


Рис. 6.3. Відображення результату

Багато нелінійних рівнянь, наприклад трансцендентні і системи з них, не мають аналітичних розв'язків і розв'язуються графічними або чисельними методами.

Функція **root(вираз, ім'я змінної)** здійснює пошук значення змінної, при якій вираз дорівнює нулю. Пошук кореня здійснюється ітераційними методами, причому перед цим треба задати початкове значення **x**.

Приклад наближеного розв'язку рівняння

Якщо необхідно знайти розв'язок рівняння з декількома змінними або системи рівнянь, задається блок рівнянь, який має наступну структуру:



Рис. 6.4. Приклад наближеного розв'язку рівняння

- Given – *службове слово, яке позначає початок блока;*
- рівняння;
- обмежувальні умови;
- вирази з функціями **Find** і **Minerr**;
- перевірка розв'язання задачі (за необхідністю).

У даному випадку використовуються наступні функції:

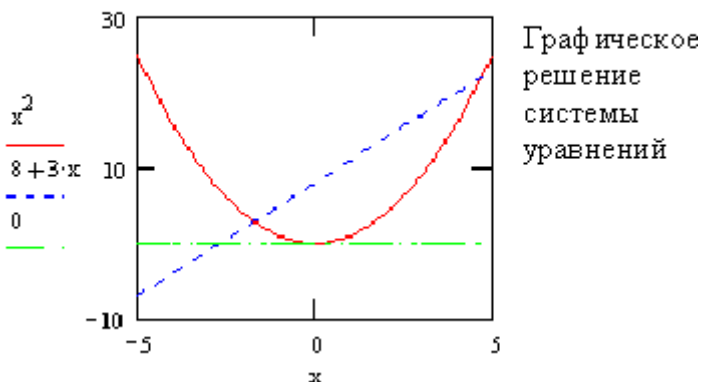
а) **Find**(x_1, x_2, \dots, x_n) – повертає значення однієї змінної або значення вектора змінних X , які відповідають точному розв'язку.

б) **Minerr**(x_1, x_2, \dots, x_n) – повертає значення однієї змінної або значення вектора змінних X , який відповідає наближеному розв'язку з мінімальною середньоквадратичною похибкою.

Обмежувальні умови служать для обмеження області розв'язання за допомогою функції **Find** або мінімізації середньоквадратичної похибки за допомогою функції **Minerr**. Вони задаються наступними конструкціями: $>$ (більше), $<$ (менше), \geq (більше або дорівнює), \leq (менше або дорівнює).

Приклад: Розв'язок системи рівнянь $\begin{cases} y = x^2 \\ y = 8 + 3x \end{cases}$

$x := -5, -4.75.. 5$



Блок першого розв'язання задачі

$x:=0$ $y:=0$ початкові значення

Given

$$y - x^2 = 0$$

$$y - 8 - 3x = 0$$

$$\begin{bmatrix} X0 \\ Y0 \end{bmatrix} := \text{Find}(x, y) \quad \begin{bmatrix} X0 \\ Y0 \end{bmatrix} = \begin{bmatrix} -1.702 \\ 2.895 \end{bmatrix}$$

$$X0^2 = 2.895 \quad \text{Перевірка розв'язання}$$

$$8 + 3X0 = 2.895$$

Блок другого розв'язання задачі

$x:=3$ $y:=0$ початкові значення

Given

$$y-x^2=0$$

$$y-8-3x=0$$

$$\begin{bmatrix} X0 \\ Y0 \end{bmatrix} := \text{Find}(x, y) \quad \begin{bmatrix} X0 \\ Y0 \end{bmatrix} = \begin{bmatrix} 4.702 \\ 22.105 \end{bmatrix}$$

$X0^2=22.105$ Перевірка розв'язку

$$8+3X0=22.105$$

Для завдання початкового значення, щоб уникнути тривіальних помилок, можна побудувати графік досліджуваної функції. Задайте в робочому документі функцію $f(x)$ і побудуйте її графік в декартових координатах. Щоб графічно знайти корені рівняння – абсциси точок перетинання графіка функції з віссю ординат, клацніть на полі графіка правою кнопкою миші. В контекстному меню, що з'явиться, виберіть пункт **Трассировка** і встановіть (стрілками клавіатури або мишею) маркер (перехресні пунктирні лінії) у точці перетинання графіка функції з віссю абсцис. У вікні діалогу відобразиться координати маркера: значення координати x при рівності координати y нулю або малій величині і є шукане наближене значення кореня (рис. 6.5).

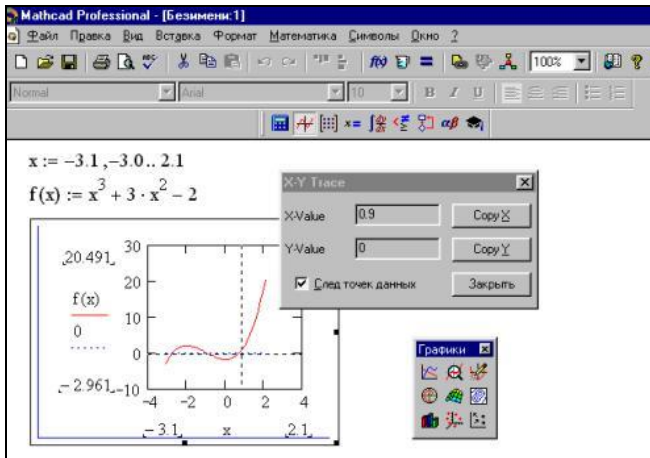


Рис. 6.5. Побудова графіка досліджуваної функції

Вектори, матриці, розв'язування систем лінійних рівнянь.

Вектор в системі – одномірний масив числових даних поєднаний ім'ям і який подається одним рядком або стовпцем. Для завдання вектора натисніть клавіші **Ctrl** і **M** або натисніть на піктограму матриці на панелі математичних символів і операторів, а потім на панелі матричних операторів. З'явиться вікно **Вставити матрицу** (рис. 6.6), в якому можна задати потрібну кількість рядків і стовпців матриці (масиву).

Задайте потрібне число рядків і стовпців. Якщо кількість рядків дорівнює одиниці, вектор буде одним рядком, якщо кількість стовпців дорівнює одиниці, вектор буде одним стовпцем.

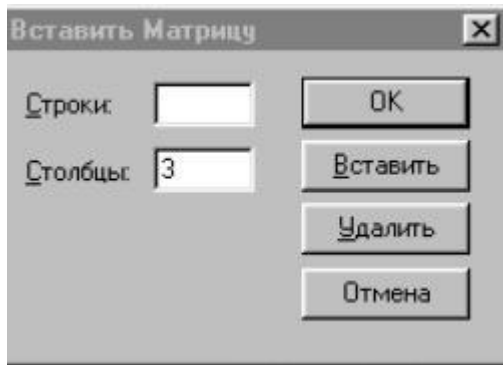


Рис. 6.6. Вікно **Вставити матрицу**

Приклади:

$$x := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad \text{завдання вектора } X$$

$y:=2x$ завдання вектора $y=2x$

$$y = \begin{pmatrix} 0 \\ 2 \\ 4 \\ 6 \\ 8 \end{pmatrix} \quad \text{виведення вектора } Y$$

$k:=0\dots 4$ поелементне створення вектора Y

$y_k:=\exp(x_k)$ k -індекс елемента

$$y = \begin{pmatrix} 1 \\ 2.718 \\ 7.389 \\ 20.086 \\ 54.598 \end{pmatrix} \quad \text{виведення вектора } Y$$

Матриця – це двомірний числовий масив з визначеним ім'ям. Завдання матриці аналогічно завданню вектора, але треба задати кількість і рядків, і стовпців. За допомогою пункту **Матриці** спадаючого меню **Символи** викликається спливаюче меню з трьома основними матричними операціями: транспонування і обернення матриць, а також обчислення визначника.

Приклад:

$$M := \begin{pmatrix} 1 \\ 2.718 \\ 7.389 \\ 20.086 \\ 54.598 \end{pmatrix} \quad \text{завдання матриці}$$

$MT := M^T$ транспонування матриці

$$MT = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix} \quad \text{виведення транспонованої матриці}$$

Більш розширені функції можна отримати за допомогою панелі матричних обчислень. За допомогою операцій над матрицями зручно розв'язувати системи лінійних рівнянь. Операції над матрицями і векторами здійснюються тими ж діями, що й для змінних.

Приклад: розв'язок системи лінійних рівнянь

$$\begin{cases} 4x_1 + 0.24x_2 - 0.08x_3 = 8 \\ 0.09x_1 + 3x_2 - 0.15x_3 = 9 \\ 0.04x_1 - 0.08x_2 + 4x_3 = 20 \end{cases}$$

$$A := \begin{pmatrix} 4 & 0.24 & -0.08 \\ 0.09 & 3 & -0.15 \\ 0.04 & -0.08 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 8 \\ 9 \\ 20 \end{pmatrix}$$

завдання матриці коефіцієнтів і вільних членів

$$A^{-1} = \begin{pmatrix} 0.25 & -0.02 & 4.261 \cdot 10^{-3} \\ -7.645 \cdot 10^{-3} & 0.334 & 0.012 \\ -2.657 \cdot 10^{-3} & 6.885 \cdot 10^{-3} & 0.25 \end{pmatrix} \text{ інвертування матриці } A$$

$X := A^{-1} \cdot B$ розв'язок системи лінійних рівнянь

$$X = \begin{pmatrix} .909 \\ 3.195 \\ 5.045 \end{pmatrix} \text{ результат розв'язку}$$

При знаходженні коренів полінома ступеня n краще використовувати функцію **polyroots(v)**, де v – вектор довжини $n+1$, що містить коефіцієнти полінома. Функція **polyroots(v)** повертає відразу всі корені полінома як дійсні, так і комплексні. Попередньо коефіцієнти полінома повинні бути представлені у вигляді вектора (рис. 6.7).



$$v := \begin{pmatrix} -10 \\ 7 \\ -5 \\ 0 \\ 1 \end{pmatrix} \quad \text{polyroots}(v) = \begin{pmatrix} -2.926 \\ 0.463 + 1.223i \\ 0.463 - 1.223i \\ 2 \end{pmatrix}$$

Рис. 6.7. Подання коефіцієнтів полінома у вигляді вектора

Додаток А

Розв'язати нелінійні рівняння (графічно і чисельно).

Варіант	Функція	a	b	c
1	$\tan ax - bx$	1.2618	1.8433	
2		2.5237	3.6866	
3	$\ln ax - bx + c$	1.2195	1.3744	0.5
4		2.7439	3.0924	1.0
5	$a \sin bx - cx$	2.33	2.857	2.0
6		4.0	3.8125	3.25
7	$x^4 - 3x - 20$	$x > 0$		
8	$x^4 + 5x - 7$			
9	$x^3 - 2x - 5$	$x > 0$		
10	$x^3 + 2x - 7$			
11	$e^x - x - 2$			
12	$2e^x x $			
13	$x^2 - \cos x$	$x > 0$		
14	$\ln x + 0.5x - 1$			
15	$\ln x - 0.5x + 1$	$x > 1$		

Додаток Б

Розв'язати систему нелінійних рівнянь.

№	Система рівнянь	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
1	$\begin{cases} ax - \tan xy = 0 \\ (y^2 - b^2) + \ln x = 0 \end{cases}$	1.0	7.5			
2		2.0	6.0			
3	$\begin{cases} ax + by + x^2y = 0 \\ \cos y + cx = 0 \end{cases}$	0.16	2.1	1.0		
4		0.24	3.5	2.0		
5	$\begin{cases} \sin(x + a) + by + c = 0 \\ \cos(y + d) + fx = 0 \end{cases}$	0.4	3.5	-1.5	0.2	0.5
6		0.8	2.0	-1.0	0.6	0.6
7	$\begin{cases} \tan ax - \cos by = 0 \\ cy^3 - x^2 - 4x - 3 = 0 \end{cases}$	1.0	1.5	2.0		
8		2.0	2.0	2.1		
9	$\begin{cases} ax^3 + by^2 - 1 = 0 \\ \sin(cx + dy) - fx = 0 \end{cases}$	3.0	14.0	3.0	0.1	-1
10		5.0	12.0	3.1	0.2	-2
11	$\begin{cases} x^{2/3} + y^{2/3} = 4 \\ x^2 - 2y = 0 \end{cases}$	$x > 0$				
12	$\begin{cases} y - \sqrt{x+1} = 0 \\ x^2 + y^2 - 2y = 0 \end{cases}$	$x > 0$				
13	$\begin{cases} x \cos x - y = 0 \\ x^2 + y^2 - 1 = 0 \end{cases}$	$x > 0$				
14	$\begin{cases} 2x^2 + y^2 = 1 \\ y - x^{2/3} = 0 \end{cases}$	$y > 0$				
15	$\begin{cases} y - \sin x = 0 \\ x^2 + y^2 - 1 = 0 \end{cases}$	$x > 0$				

Додаток В
Розв'язати систему лінійних рівнянь

№ варіанта	Матриця коефіцієнтів системи			Стовпець вільних членів
1	1.84	2.25	2.53	-6.09
	2.32	2.6	2.82	-6.98
	1.83	2.06	2.24	-5.52
2	2.18	2.44	2.49	-4.34
	2.17	2.31	1.49	-3.91
	3.15	3.22	3.17	-5.27
3	1.54	1.7	1.62	-1.97
	3.69	3.73	3.59	-3.74
	2.45	2.43	2.25	-2.26
4	1.53	1.61	1.43	-5.13
	2.35	2.31	2.07	-3.69
	3.83	3.73	3.45	-5.98
5	2.36	2.37	2.13	1.48
	2.51	2.4	2.1	1.92
	2.59	2.41	2.06	2.16
6	5	0	1	11
	1	3	-1	4
	-3	2	10	6
7	2	0	-1	-3
	-1	3	1	2
	1	-1	4	3
8	2	0	-1	1
	1	-3	1	2
	1	1	3	4
9	5	1	-1	-5
	-1	3	1	5
	1	-2	4	1
10	3	1	-1	-1
	-2	4	1	5
	1	1	3	-3
11	3	1	-1	6
	2	4	1	9
	1	-1	3	4
12	2	-1	0	-2
	2	5	-2	-4
	1	-1	3	2
13	3	-1	1	1
	0	2	-1	3
	-1	1	5	-5
14	4	1	-1	7
	2	3	0	7
	1	-1	5	11
15	2	0	-1	1
	1	-4	2	-5
	1	1	3	6

ПРАКТИЧНА РОБОТА № 7

РОЗВ'ЯЗОК ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

1. Мета роботи

Вивчення основ розв'язування диференціальних рівнянь в системі MathCAD 2000i

2. Задачі роботи

Засвоїти прийоми чисельного розв'язування диференціальних рівнянь та їх систем в середовищі комп'ютерної математики MathCAD 2000i

3. Зміст роботи

3.1 Запустити додаток MathCAD 2000i.

3.2 Повторити всі приклади, наведені в методичних вказівках.

3.3. Розв'язати задачу Коші чисельно і графічно для диференціального рівняння першого порядку (додаток А).

3.4 Знайти розв'язок задачі Коші для диференціального рівняння другого порядку і побудувати графік знайденого розв'язку (додаток Б).

3.5 Побудувати таблицю значень і графік розв'язку для нормальної системи диференціальних рівнянь (додаток В).

3.6 Зберегти результати у файлі з введенням необхідних коментарів і оформити звіт.

4. Вимоги до звіту

Звіт повинний включати:

– титульний лист з назвою роботи та відомостями про виконавця;

– колонтитули з П.І.Б. виконавця і шифром групи.

– постановку задачі дослідження, послідовність її виконання;

– результати обчислень згідно варіанту (на дискеті (флешці) і в роздрукованому вигляді);

– відповіді на контрольні запитання.

5. Рекомендована література

5.1. Очков В.Ф. MathCAD7 Pro для студентов и инженеров. - М.: Компьютер Пресс, 1998.-384 с.

5.2. Плис А.И., Сливина Н.А. MathCAD: Математический практикум для экономистов и инженеров: Учебное пособие.-М.: Финансы и статистика, 1999.- 656 с.

6. Загальні положення

Звичайним диференціальним рівнянням порядку n називається рівняння $f(x, y(x), y'(x), \dots, y^n(x)) = 0$, яке зв'язує незалежну змінну x , шукану функцію $y=y(x)$ і її похідні $y'(x), \dots, y^n(x)$

Системою звичайних диференціальних рівнянь називається система рівнянь

$f_i(x, y_i(x), y_2(x), \dots, y_1(x), y_2(x), \dots) = 0, i = \overline{1, n}$ яка зв'язує незалежну змінну x , шукані функції $y_1 = y_1(x), y_2(x), \dots$ та їх похідні.

Всі функції MathCad призначені для чисельного розв'язування задачі Коші нормальних систем звичайних диференціальних рівнянь. Задача Коші для рівнянь зводиться до розв'язку задачі для системи.

$$\begin{cases} y'_1 = f_1(x, y_1, y_2, \dots, y_n) \\ y'_2 = f_2(x, y_1, y_2, \dots, y_n) \\ \dots \dots \dots \dots \dots \dots \\ y'_n = f_n(x, y_1, y_2, \dots, y_n) \end{cases} \begin{cases} y_1(x_0) = y_{0,1} \\ y_2(x_0) = y_{0,2} \\ \dots \dots \dots \dots \dots \dots \\ y_n(x_0) = y_n \end{cases}$$

Чисельний розв'язок цієї задачі полягає у побудові таблиці наближених значень $y_{i,1}, y_{i,2}, \dots, y_{i,n}, i = 1, 2, \dots, N$, розв'язку $y_1(x), y_2(x), \dots, y_n(x)$ на відрізку $[x_0, x_N]$ в точках x_0, x_1, \dots, x_N , які називаються вузлами сітки.

Позначивши

$$Y(x) = (y_1(x), y_2(x), \dots, y_n(x)),$$

$$Y_0 = (y_{0,0}; y_{0,1}; \dots; y_{0,n}),$$

$$Y' = (y_1'(x), y_2'(x), \dots, y_n'(x)),$$

$$F(x, Y) = (f_1(x, y_1, \dots, y_n), f_2(x, y_1, \dots, y_n), \dots, f_n(x, y_1, \dots, y_n)),$$

де Y – шуканий розв'язок; Y_0 – вектор початкових умов;

$F(x, Y)$ – вектор правих частин, запишемо систему диференціальних рівнянь у векторній формі:

$$Y' = F(x, Y), \quad Y(x_0) = Y_0.$$

В Mathcad розв'язати задачу Коші для такої системи можна за допомогою, наприклад, наступних функцій: **rkfixed(y, x1, x2, npoints, D)** – розв'язок задачі на відрізку методом Рунге-Кутта з постійним кроком.

Значення параметрів визначається математичною постановкою задачі:

Y – вектор початкових умов **Y₀, y_i=(Y)_i**;

x₁, x₂ – початкова і кінцева точки відрізка інтегрування системи;

Для функцій, що здійснюють розв'язок у заданій точці, **x₁**

– початкова точка, **x₂** – задана точка;

npoints – кількість вузлів на відрізку [**x₁, x₂**];

При розв'язанні задачі на відрізку, результат містить **npoints+1** рядків;

D – ім'я вектора-функції **D(x, y)**, який містить праві частини **F(x, Y), D_i(x, y)=f_i(x, y₁, ..., y_n)**.

Результат роботи функції – матриця, яка містить **n+1** стовпець; її перший стовпець містить координати вузлів сітки, другий стовпець – обчислені наближені значення розв'язку **y_i(x)** у вузлах сітки, (**k+1**)-й – значення розв'язку **y_k(x)**.

При розв'язанні задачі Коші для диференціального рівняння першого порядку результати обчислень усіх наведених вище функцій буде матриця, у першому стовпці якої містяться координати вузлів сітки **x₀, x₁, ..., x_n**, а в другому – значення наближеного розв'язку у відповідних вузлах.

Задача Коші для диференціального рівняння порядку вище першого (**n-го** порядку) зводиться до задачі Коші для еквівалентної нормальної системи диференціальних рівнянь **n-го** порядку.

Нехай **y⁽ⁿ⁾=f(x, y, y', ..., y⁽ⁿ⁻¹⁾)**, **y(x₀)=y₀**,

y'(x₀)=y_{0,1}, ..., y⁽ⁿ⁻¹⁾(x₀)=y_{0,n-1}. Позначивши **y₁(x)= y(x)**, **y₂(x)= y'(x)**, ... , **y_k(x)=y^(k-1)(x)**, ... , **y_n(x)=y⁽ⁿ⁻¹⁾(x)**, **y₀(x)= y(x₀)**, **y_{0,1}(x)=y'(x₀)**, ... , **y_{0,k}(x)=y^(k)(x)**, ... , **y_{0,n-1}(x)=y⁽ⁿ⁻¹⁾(x)**, отримаємо задачу Коші для нормальної системи диференціальних рівнянь, розв'язання якої в Mathcad описане вище:


$$\left\{ \begin{array}{l} y'_1 = y_2, \\ y'_2 = y_3, \\ \dots\dots\dots \\ y'_{n-1} = y_n, \\ y'_n = f(x, y_1, y_2, \dots, y_{n-1}) \end{array} \right. \quad \left\{ \begin{array}{l} y_1(x_0) = y_{0,0} \\ y_2(x_0) = y_{0,1} \\ \dots\dots\dots \\ y_n(x_0) = y_{0,n-1} \end{array} \right.$$

Розв'язок задачі Коші для диференціального рівняння першого порядку. Знайдемо на відрізку $[\pi, 0]$ наближений розв'язок рівняння $y' = \sin xy$, яке задовольняє початковим умовам $y(0) = 1$, і побудуємо графік знайденого розв'язку. Розв'яжемо задачу чисельно, використовуючи алгоритм Рунге-Кутта з фіксованим кроком на сітці з 20 рівновіддалених вузлів.

Фрагмент робочого документа Mathcad, що містить обчислення і графік, наведений на рис. 7.1.



Привласніть змінній **ORIGIN** значення, яке дорівнює **1**, щоб нумерація компонент вектора починалася з одиниці. При розв'язку задачі Коші для рівняння першого порядку вектор розв'язку має єдину компоненту y_1 . Вектор правих частин також містить одну компоненту. Перед зверненням до функції **rkfixed** привласніть змінній y_1 початкове значення, яке дорівнює одиниці, а змінній $D(x, y)$ – вираз для правої частини рівняння, що дорівнює **$\sin(x, y_1)$** .

В наведеному фрагменті результати обчислень функції **rkfixed(y, 0, π , 20, D)** привласнені матриці **Y**; вона містить у першому стовпці координати 20 вузлів рівномірної сітки на відрізку $[0, \pi]$, а в другому – наближені значення розв'язку в цих вузлах (в наведеному документі представлена тільки частина таблиці). Для того, щоб ввести в робочий документ таблицю значень розв'язання, наберіть на клавіатурі символи **Y =**, і таблиця значень буде виведена в робочому документі праворуч від знака дорівнює. Клацніть по полю таблиці і перегляньте її, використовуючи стрілки прокрутки. Для того, щоб побудувати

графік розв'язку, клацніть по кнопці  на панелі графіків



і введіть в якості змінної на осі абсцис **Y<1>** (стовпець координат вузлів сітки), а на осі ординат - **Y<2>** (стовпець значень розв'язку у вузлах сітки). Щоб ввести номер стовпця,

введіть ім'я матриці, клацніть на кнопці  на панелі  і введіть в позначеній позиції номер стовпця.

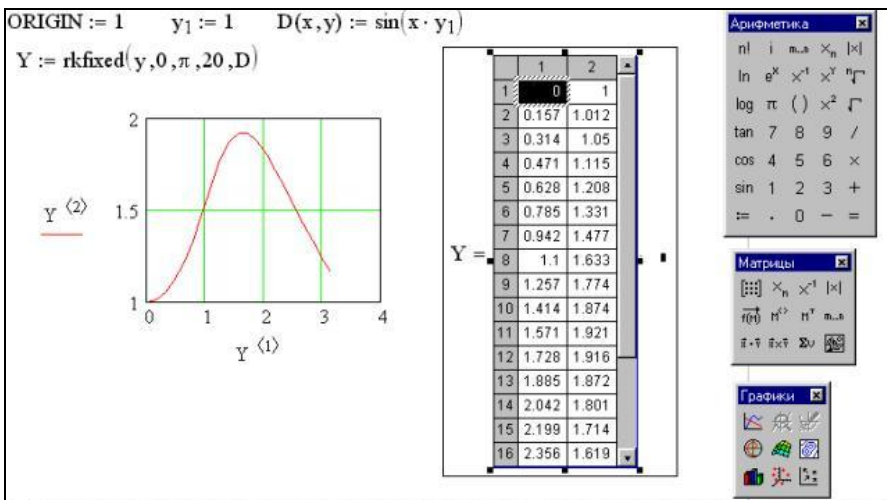


Рис. 7.1. Фрагмент робочого документа Mathcad, що містить обчислення і графік задачі Коші для диференціального рівняння першого порядку

Розв'язання задачі Коші для диференціальних рівнянь вищих порядків. Знайдемо на відрізку $[0,3]$ наближений розв'язок рівняння $y'' = e^{-xy}$, який задовольняє початковим умовам $y(0) = 1, y'(0) = 1$ і побудуємо графік знайденого розв'язку.

Зведемо розв'язок задачі для рівняння другого порядку до задачі для еквівалентної нормальної системи другого порядку. Позначимо $y_1(x) = y(x)$ і $y_2(x) = y'(x)$.

Оскільки $y''(x) = (y'(x))' = y_2'(x)$, то отримаємо

$$\begin{cases} y_1' = y_2 \\ y_2' = \exp(-xy_1) \end{cases} \begin{cases} y_1(0) = 1, \\ y_2(0) = 1. \end{cases}$$

Розв'яжемо задачу чисельно, використовуючи алгоритм Рунге-Кутта з фіксованим кроком на сітці з 20 рівновіддалених вузлів.

Фрагмент робочого документа Mathcad, що містить обчислення і графік, наведений на рис. 7.2.

Примітка. Визначте вектор \mathbf{y} і вектор-функцію $\mathbf{D}(\mathbf{x}, \mathbf{y})$ як матриці розмірності 2×1 . Привласніть компонентам вектора \mathbf{y} початкові значення, а компонентам вектора $\mathbf{D}(\mathbf{x}, \mathbf{y})$ – вираз для правих частин рівнянь системи. В інших випадках дійте так само, як в попередньому прикладі.

Розв'язок задачі Коші для нормальної системи диференціальних рівнянь. Знайдемо на відрізку $[0,3]$ наближене розв'язок задачі Коші:

$$\begin{cases} y'_1 = -y_2 + \sin xy_3, \\ y'_2 = -y_1^2, \\ y'_3 = -y_3 - y_1, \end{cases} \quad \begin{cases} y_1(0) = 1, \\ y_2(0) = 0, \\ y_3(0) = 1 \end{cases}$$

і побудуємо графіки для знайденого розв'язку. Розв'яжемо задачу чисельно, використовуючи алгоритм Рунге-Кутта з фіксованим кроком на сітці з 30 рівновіддалених вузлів (рис.7.3).

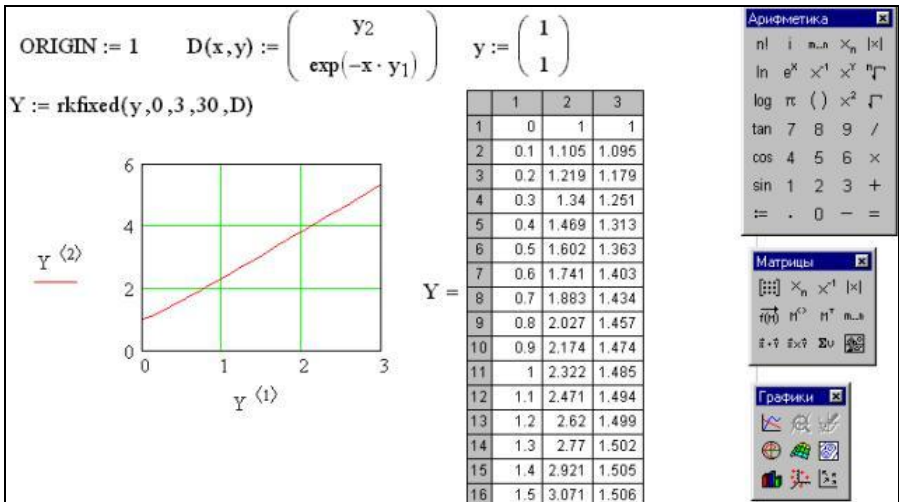


Рис. 7.2. Фрагмент робочого документа Mathcad, що містить обчислення і графік розв'язання задачі Коші для диференціального рівняння $y'' = e^{-xy}$

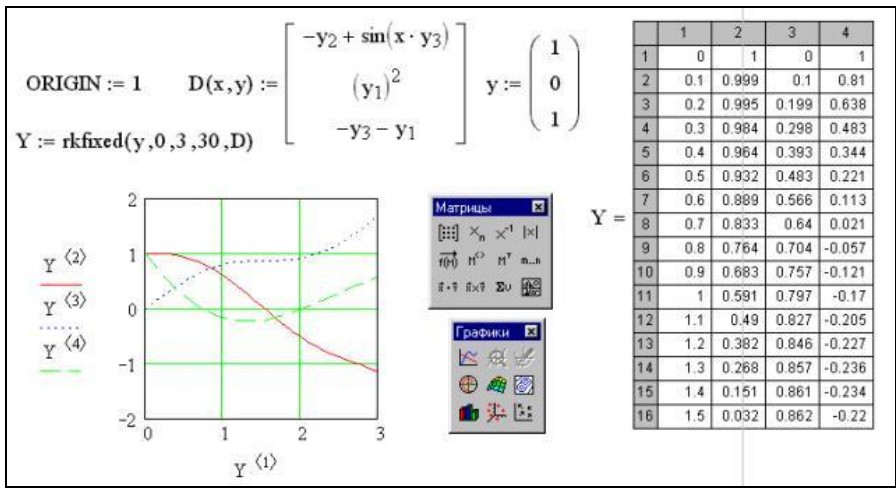


Рис. 7.3. Розв'язок задачі Коші для нормальної системи диференціальних рівнянь

Примітка. Для того, щоб побудувати графіки знайденого розв'язку (графіки функцій $y_1(x), y_2(x), y_3(x)$), введіть в якості змінної на осі абсцис $Y^{<1>}$ (стовпець координат вузлів сітки), а на осі ординат введіть, розділяючи комою, $Y^{<2>}, Y^{<3>}, Y^{<4>}$ (стовпці, що містять відповідно значення $y_1(x), y_2(x), y_3(x)$ у вузлах сітки).

Додаток А

N	$f(t,y)$	t_0	T	y_0	N	$f(t,y)$	t_0	T	y_0
1	$\frac{y}{t} + t^2$	1	2	0	16	$-\frac{y}{t} + 3t$	1	2	1
2	$yctgt + 2t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	0	17	$\frac{2ty}{1+t^2} + 1 + t^2$	1	2	3
3	$-y \cos t + \frac{\sin(2t)}{2}$	0	1	0	18	$\frac{2t-1}{t^2} y + 1$	1	2	1
4	$-ytgt + \cos^2 t$	$\frac{\pi}{4}$	$\frac{\pi}{4} + 1$	0.5	19	$-\frac{3y}{t} + \frac{2}{t^3}$	1	2	1
5	$\frac{y}{t+2} + t + 2t$	-1	0	1.5	20	$-2ty - 2t^3$	1	2	e^{-1}
6	$\frac{y}{t+1} + e'(t+1)$	0	1	1	21	$\frac{y}{t} - \frac{2}{t^2}$	1	1	1
7	$\frac{y}{t} + t \sin t$	$\frac{\pi}{2}$	$\frac{\pi}{2} + 1$	1	22	$-ty - t^3$	0	1	3
8	$-\frac{y}{t} + \sin t$	π	$\pi + 1$	$\frac{1}{\pi}$	23	$\frac{2}{t+1} y + e'(t+1)^2$	0	1	1
9	$-\frac{y}{2t} + t^2$	1	2	1	24	$-2ty + te^{-t^2} \sin t$	0	1	1
10	$-\frac{2t}{1+t^2} y + \frac{2t^2}{1+t^2}$	0	1	$\frac{2}{3}$	25	$\frac{2y}{t+1} + (t+1)^3$	0	1	0.5
11	$\frac{2t-5}{t^2} y + 5$	2	3	4	26	$y \cos t - \sin 2t$	0	1	3
12	$-\frac{y}{t} + \frac{t+1}{t} e^t$	1	2	e	27	$4ty - 4t^3$	0	1	-0.5
13	$\frac{y}{t} - 2 \ln t / t$	1	2	1	28	$\frac{y}{t} - \frac{\ln t}{t}$	1	2	1
14	$\frac{y}{t} - \frac{12}{t^3}$	1	2	4	29	$3t^2 + t^2(1+t^3)/3$	0	1	0
15	$-\frac{2y}{t} + t^3$	1	2	$-\frac{5}{6}$	30	$y \cos t + \sin t$	0	1	-1

Додаток Б

Задача Коші для звичайного диференціального рівняння 2 порядку

$$mx'' + Hx' + kx = f(t), \quad t \in [0, T], \quad \begin{matrix} x(0) = x_0 \\ x'(0) = v_0 \end{matrix}$$

описує рух вантажу масою m , підвішеного до кінця пружини. Тут $x(t)$ – зсув вантажу від положення рівноваги, H – константа, яка характеризує силу опору середовища, k – коефіцієнт пружності пружини, $f(t)$ – зовнішня сила.

Початкові умови: x_0 – зсув вантажу у початковий момент часу $t=0$, v_0 – швидкість вантажу в початковий момент часу. Промодельовати рух вантажу на часовому інтервалі $[0, T]$ при заданих в індивідуальному варіанті трьох наборах (I, II, III) значень параметрів задачі.

N		H	k	m	f(t)	x ₀	v ₀	T
1	I	0.5	1	1	0	10	0	20
	II	“-	“-	“-	\sqrt{t}	0	“-	“-
	III	“-	“-	“-	\sqrt{t}	-10	“-	“-
2	I	1	1	0.5	tsin(t)	0	0	20
	II	“-	“-	“-	0	“-	-10	“-
	III	“-	“-	“-	tsin(t)	“-	-50	“-
3	I	1	5	0.75	0	-10	0	5
	II	“-	“-	“-	“-	0	10	“-
	III	“-	“-	“-	“-	-10	10	“-
4	I	1	1	1	cos(t)	0	0	20
	II	“-	“-	3	“-	“-	“-	“-
	III	“-	“-	6	“-	“-	“-	“-
5	I	0.5	5	1	0	20	0	15
	II	“-	50	“-	“-	“-	“-	“-
	III	“-	0.5	“-	“-	“-	“-	“-
6	I	1	5	1	0	0	1	15
	II	“-	0.5	“-	“-	“-	“-	“-
	III	“-	50	“-	“-	“-	“-	“-
7	I	1	1	5	-t	15	0	40
	II	0.1	“-	“-	“-	“-	“-	“-
	III	10	“-	“-	“-	“-	“-	“-
8	I	1	1	0.5	sin(t)	0	0	20
	II	“-	“-	5	“-	“-	“-	“-
	III	“-	“-	50	“-	“-	“-	“-

Додаток В

Дані задачі Коші для систем звичайних диференціальних рівнянь першого порядку з постійними коефіцієнтами на відрізку $[0, 1]$.

$$Y'(t) = AY(t), \quad Y(0) = Y_0,$$

$$Z'(t) = BZ(t), \quad Z(0) = Z_0,$$

де A і B – задані матриці, Y_0 , Z_0 – задані вектори.

N	A		Y_0	N	B		Z_0
1	-1.999	-0.019	0	6	-10.850	9.787	1
	-0.063	-1.051	1		32.515	-499.55	0
2	-13.237	15.299	2	7	-6.905	0.03	1
	33.885	522.183	0		-0.145	-6.095	5
3	-0.717	-23.827	1	8	-1.905	-0.015	1
	114.483	-640.393	2		-0.13	-2.295	0
4	-17.359	-0.573	2	9	-64.712	-85.344	1
	5.366	-21.351	1		-128.964	-170.918	0
5	-229.934	301.266	1	10	-2.018	-0.818	1
	227.624	-303.576	1		-0.082	-1.282	1

Приклад оформлення титульного аркушу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Чернігівський державний інститут економіки і управління

Кафедра „Економічної кібернетики та інформатики”

ЗВІТ

про виконання лабораторної роботи

”_____”

(назва роботи)

з дисципліни:

“Алгоритмізація та програмування”

Виконав: студент

групи ЕК – 111 _____ Петренко С. М.

Перевірив:

к.т.н., доцент _____ Зацерковний В. І.

Чернігів 2011

Зміст

ПЕРЕДМОВА	3
РОЗДІЛ 1. ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ Turbo Pascal	
Лабораторна робота №1	8
Лабораторна робота №2	23
Лабораторна робота №3	31
Лабораторна робота №4	40
Лабораторна робота №5	55
Лабораторна робота №6	70
РОЗДІЛ 2. ПРОГРАМУВАННЯ В СЕРЕДОВИЩІ MathCad 2001i	
Практична робота №1	82
Практична робота №2.....	94
Практична робота №3.....	112
Практична робота №4.....	122
Практична робота №5.....	130
Практична робота №6	141
Практична робота №7.....	153
Приклад оформлення титульного аркушу	163