

Міністерство освіти та науки України
Національний технічний університет «Харківський політехнічний інститут»

**МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ «ІНФОРМАТИКА»**

для студентів спеціальності 6.050202
«Автоматизація управління технологічними процесами»
денної та заочної форми навчання

Харків 2014

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсової роботи з дисципліни
«Інформатика»

для студентів спеціальності 6.050202
«Автоматизація управління технологічними процесами»
заочної форми навчання

Затверджено
редакційно-видавничою
радою університету,
протокол № 1 від 07.06.2013 р.

Харків
НТУ «ХПІ»
2014

Методичні вказівки до виконання курсової роботи з дисципліни
«Інформатика» для студентів спеціальності 6.050202 «Автоматизоване

управління технологічними процесами» заочної форми навчання / уклад. Хавін Г.Л. – Х. : НТУ «ХП», 2014. – 60 с.

Укладач Г.Л. Хавін

Рецензент І.Е. Яковенко

Кафедра технології машинобудування та металорізальних верстатів

ВСТУП

Для кваліфікованої підготовки інженерів необхідно, щоб кожний студент навчився ефективно використовувати обчислювальну техніку. Поперед усього це передбачає вміння складати програми для рішення задач, пов'язаних з обробкою науково-технічної інформації. Розв'язання інженерних задач, що базуються на створених математичних моделях реальних об'єктів, не можливе без практичної підготовки студентів по написанню програм у на різних алгоритмічних мовах, серед яких найбільш застосовною є мова програмування Турбо Паскаль. Змістом даних методичних вказівок по виконанню курсової роботи є вивчення та подальше закріплення навичок програмування на мові Турбо Паскаль.

Метою виконання курсової роботи з дисципліни «Інформатика» як одного з етапів навчання за спеціальністю 6.050202 «Автоматизація управління технологічними процесами» є застосування теоретичних знань, практичних навичок та вмінь, що одержані в процесі аудиторних занять, і використання їх у практичній роботі зі складання інженерних та математичних програм. Виконання курсової роботи являє собою один з етапів підготовки до вивчення спеціальних дисциплін.

Мета курсової роботи – подальше закріплення знань, що одержані на лекціях та при виконанні лабораторних робіт. Курсова робота присвячена навчанню самостійно розробляти алгоритми і складати програми на мові Турбо Паскаль.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Курсова робота передбачає:

- засвоєння студентами теоретичних основ мови програмування Турбо Паскаль;
- оволодіння студентами навичок складання алгоритмів та програм;

➤ практичну роботу студентів по програмуванню і самостійного аналізу результатів.

Курсова робота є результатом самостійної роботи студента по засвоєнню їм теоретичних основ мови Турбо Паскаль і методів розробки алгоритмів. При виконанні курсової роботи треба мати навички побудови алгоритмів і роботи у середовищі програмування. Для виконання курсової роботи студент повинен вивчити лекційний матеріал, зробити лабораторні роботи і ознайомитися з додатковою літературою. Курсова робота повинна носити творчий характер і виконуватися у відведений строк. Результатом виконання курсової роботи є розроблені алгоритми і програми, які виконують обчислення за заданими даними.

2 СТРУКТУРА, СКЛАД І ОБ'ЄМ КУРСОВОЇ РОБОТИ

Курсова робота складається з виконання мови Турбо Паскаль завдань окремих розділів. Варіант курсової роботи студент обирає за порядковим номером у списку групи. Варіанти курсової роботи надані у окремих таблицях до кожного розділу.

Курсова робота складається з виконання завдань у вигляді програм на мові Паскаль до кожної теми усіх чотирьох розділів. До складу курсової роботи (100%) входить виконання таких завдань за розділами:

- Розділ 1 – Середовище програмування Паскаль (10%);
 - тема 1 – Базові елементи мови Паскаль. Текстовий редактор Турбо Паскалю, запуск програми, редагування, поглядання програми і результатів
 - тема 2 – Константи та змінні. Типи даних;
 - тема 3 – Вираження, операнди та операції, оператор присвоєння;
 - тема 4 – Структура програми;
- Розділ 2 – Оператори умовного та безумовного переходу (15%);
- Розділ 3 – Структуровані типи даних, програмування алгоритмів з ітераційним циклом (30%);

- Розділ 4 – Загальні свідомості про процедури та функції, базові положення програмування, складення програм з використанням процедур і функцій (45%);

Курсова робота оформлюється у вигляді розрахунково-пояснювальної записки, що містить у себе: варіант завдання, програми на Паскалі, результати обчислень та їх аналіз.

Об'єм розрахунково-пояснювальної записки повинен складати не більш 15-20 листів паперу стандартного розміру А4, що надруковано або написано з виконанням вимог стандарту НТУ „ХПІ”. Звіт складається з окремих глав з назвою розділів, що логічно слідує із змісту, і параграфів. Курсова робота повинна починатися з титульного листу і закінчуватися списком джерел інформації.

3 ОСНОВНІ ЕТАПИ І ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

№ етапу	Вид роботи	Тиждень виконання	Об'єм виконання, %
1	Одержання завдання	1	-
2	Виконання завдань розділу 1	2	10
3	Побудова розгалужених програм за допомогою оператора умовного переходу (розділ 2)	5	25
4	Створення програми із структурованими типами даних та з ітераційним циклом (розділ 3)	10	55
5	Програмування з використанням процедури-функції (function) з розділу 4	13	70
6	Програмування з використанням підпрограми-процедури (procedure) з розділу 4	16	95
7	Перевірка та оформлення роботи	17	100
8	Захист роботи	18	-

4 МЕТОДИЧНІ ВКАЗІВКИ І КОНТРОЛЬНІ ЗАВДАННЯ ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ

РОЗДІЛ 1. СЕРЕДОВИЩЕ ПРОГРАМУВАННЯ ПАСКАЛЬ

Тема 1 Базові елементи мови Паскаль. Текстовий редактор Турбо Паскалю, запуск програми, редагування, поглядання програми і результатів

Мета виконання завдання – ознайомлення з базовими положеннями мови Паскаль. Навчання роботі у редакторі, опис типів даних, констант та змінних. Оволодіння записом операндів та операцій.

Методичні вказівки

Система програмування Турбо Паскаль складається з інтелектуального інтегрованого середовища і мови програмування. Інтелектуальне інтегроване середовища забезпечує функціонування середовища Турбо Паскаль, у тому разі роботу текстового редактора, вікон, діалогу та інше.

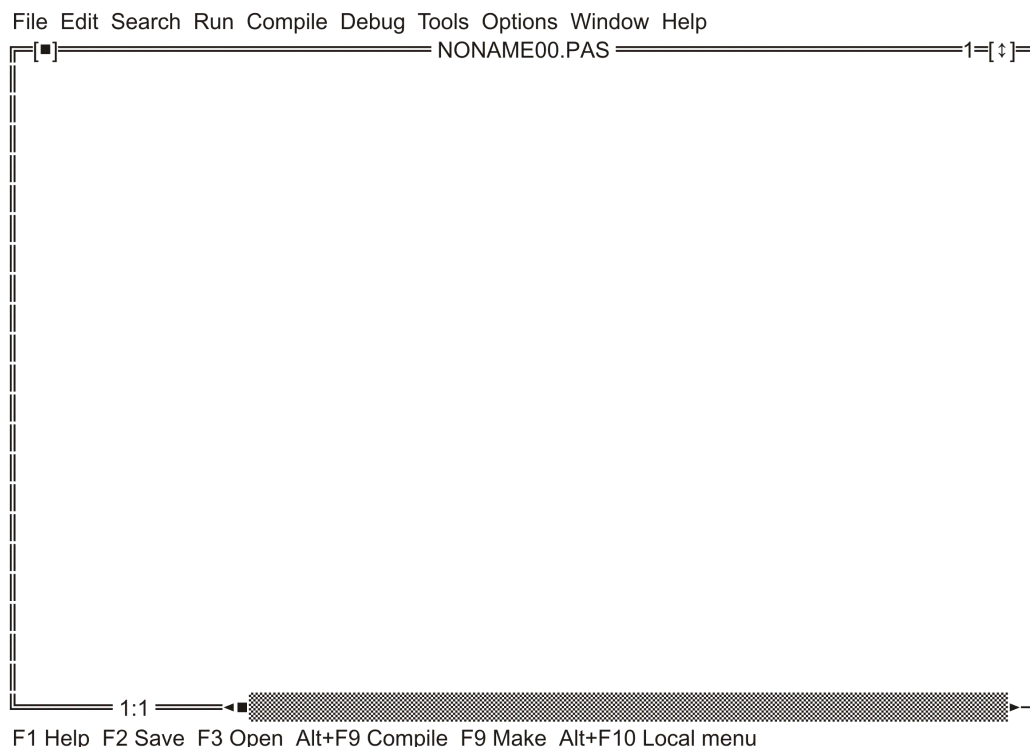


Рисунок 1.1 – Головне меню Турбо Паскалю

Для запуску Турбо Паскалю необхідно знайти у каталозі файл turbo.exe и завантажити його. Після цього на екрані з'явиться головне меню, що наведено на рис.1.1. На нижній строчці вказані призначення функціональних клавіш. Верхня строчка містить в себе меню (Main Bar), що дає можливість виконувати дії з програмою:

- File – робота з файлами;
- Edit – редагування;
- Search – пошук і заміна у тексті;
- Run – виконання програми;
- Compile – компіляція програми;
- Debug – коректування програми;
- Tools – використання зовнішніх програм;
- Options – встановлення режимів;
- Window – робота з вікнами;
- Help – допомога.

Область вікна, що обрамлена подвійною рамкою належить до активного вікна. У першій строчці наведено ім'я файлу, звідки було отримано текст програми, або NONAME00.PAS, якщо це новий файл, який не читався з диска. У правому куті рамки виводиться номер вікна (припускається створювати до 9 вікон) У вікні є два спеціальних поля, що використовуються при роботі з мишею, які виділено квадратними скобками. Окрім вікон редактора на екрані можуть знаходитися вікна: режиму корекції, виводу результатів, реєстрів, довідкової служби та інше.

Робота з функціональними клавішами Турбо Паскалю

Для роботи використовують функціональні клавіші F1, F2, ..., F12 і комбінації функціональних клавіш і клавіш Alt – додавання, Ctrl – управління; Shift – зсув. Сумісне натискання двох клавіш – символ «+». Функціональні клавіші дозволяють виконувати такі дії:

- F1 – виклик довідкової служби;

F2 – запис тексту у дисковий файл;

F3 – виклик у вікно текстового редактора;

F4 – використовується у режимі коректування: почати або продовжити виконання програми та зупинитися перед виконанням тієї її строки, на який знаходиться курсор;

F5 – розкрити активне вікно на весь екран;

F6 – зробити активним друге вікно, якщо відкрити декілька вікон;

F7 – використовується у режимі коректування: виконується покроковий режим роботи з викликом процедур та функцій;

F8 – використовується у режимі коректування: виконується покроковий режим роботи без виклику процедур та функцій;

F9 – компілювання програми без виконання;

F10 – перехід до головного меню середовища програмування;

Ctrl + F9 – компілювання програми з виконанням;

Alt + F5 – змінити вікно редактора на вікно MS DOS (виводу результатів);

Alt + X – вихід з середовища Турбо Паскаль.

Текстовий редактор Турбо Паскаль

Головною ознакою того, що Ви знаходитесь у режимі редагування є наявність у вікні курсору у вигляді пробліскового прямокутника. Для переходу у головне меню необхідно натиснути клавішу F10, повернутися до вікна редагування можна по натисненню клавіші Esc.

Введення символів проводиться з клавіатури, перехід на нову строку – натисканням клавіші Enter. Максимальна кількість символів у файлі не більш 64535, максимальна довжина строки – не більш 126 символів.

Клавіші роботи з вікнами:

PgUp – на сторінку вгору;

PgDn – на сторінку вниз;

Home – на початок даної строки;

End – у кінець даної строки;

Ctrl+PgUp – на початок тексту;

Ctrl+PgDn – у кінець тексту;

←→↑↓ – клавіші управління курсором.

Виключити попередній символ можна за допомогою клавіші Backspace. Натисканням клавіші Del вилучається символ, на якому встановлено курсор. Для розділення строки необхідно встановити курсор у задану позицію та натиснути клавішу Enter. Для вставлення символу використовується клавіша Ins – вставлення. При натисненні цієї клавіші вигляд курсору змінюється на пробісний символ підкреслення, а у режимі напису символу поверх іншого курсор переходить у режим великого пробісного прямокутника.

Редагування строк або блоків програми здійснюється натисканням комбінацій клавіш:

Ctrl+O+I – відключення режиму автовідступу;

Ctrl+Q+L – відновити дану строку (робитиме, якщо курсор не виходив з строки, що змінювалася);

Ctrl+K+B – помітити початок блоку;

Ctrl+K+K – помітити кінець блоку;

Ctrl+K+Y – витерти блок;

Ctrl+K+C – копіювати блок;

Ctrl+K+V – перемістити блок;

Ctrl+K+W – записати блок у дисковий файл;

Ctrl+K+R – прочитати блок з дискового файлу;

Ctrl+K+P – надрукувати блок.

Алфавіт мови Паскаль містить в себе великі та малі літери латинського алфавіту і знак підкреслення. Окрім цього він має десятинні та шістнадцятирічні цифри, а також спеціальні символи:

+ - плюс

- - мінус

/ - дрібна риса

= - дорівнює

> - більш; < - менш

^ - тильда

{ } - фігурні дужки

* - зірочка

: - двокрапка

' - апостроф

\$ - грошовий знак

@ - комерційне а

. - крапка

, - кома

; - крапка з комою

- номер

[] - квадратні дужки

- пробіл

Комбінації спеціальних символів можуть утворювати складені символи:

<code>:=</code> – привласнювання	<code><></code> – не дорівнює	<code>..</code> – діапазон значень
<code>(* *)</code> – альтернатива <code>{ }</code>	<code><=</code> – менш або дорівнює	<code>>=</code> – більш або дорівнює
<code>(. .)</code> – альтернатива <code>[]</code>		

Неподільна послідовність знаків алфавіту встановлює слова, що відокремлені один від одного роздільником у вигляді пробілу, символу кінця рядку або коментаря. Усі слова підрозділяються на зарезервовані слова, стандартні ідентифікатори та ідентифікатори користувача.

Зарезервовані слова відрізняються тим, що вони мають фіксоване начертання і назавжди визначене значення. Ці слова такі:

<code>absolute</code>	<code>destructor</code>	<code>function</code>	<code>mod</code>	<code>record</code>	<code>type</code>
<code>and</code>	<code>downto</code>	<code>goto</code>	<code>nil</code>	<code>repeat</code>	<code>unit</code>
<code>array</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>set</code>	<code>until</code>
<code>begin</code>	<code>end</code>	<code>implantation</code>	<code>of</code>	<code>shl</code>	<code>uses</code>
<code>case</code>	<code>externa</code>	<code>inline</code>	<code>or</code>	<code>shr</code>	<code>var</code>
<code>const</code>	<code>file</code>	<code>interface</code>	<code>object</code>	<code>string</code>	<code>while</code>
<code>div</code>	<code>for</code>	<code>interrupt</code>	<code>procedure</code>	<code>then</code>	<code>with</code>
<code>do</code>	<code>forward</code>	<code>label</code>	<code>program</code>	<code>to</code>	<code>xor</code>
<code>packed</code>	<code>constructor</code>	<code>virtual</code>	<code>in</code>		

Стандартні ідентифікатори використовуються для позначення заздалегідь визначених типів даних, констант, процедур та функцій. Більш детально з переліком цих ідентифікаторів та їх призначенням можна ознайомитися у рекомендованій літературі [1]. Ідентифікатори користувача це ті ідентифікатори, що необхідні йому для позначення констант, міток, змінних, процедур та функцій при написанні програми. Загальні правила написання ідентифікаторів такі:

- ідентифікатор може починатися тільки з літери або знака підкреслення, за виключенням міток, які можуть починатися цифрою;
- ідентифікатор може складатися тільки з літер, цифр та знака підкреслення;

- між двома ідентифікаторами повинно бути не менш одного пробілу;
- максимальна довжина ідентифікатора 127 символів, але значущі тільки перші 63.

При написанні ідентифікаторів можна використовувати як великі, так і малі літери. Приклади написання ідентифікаторів:

Atp_22

freet

TUIK12

5_sto – помилка, ідентифікатор починається з цифри.

Kr125T

er.124U – помилка, ідентифікатор містить крапку.

Порядок виконання контрольного завдання до теми 1

1. Ознайомитися з порядком роботи у середовищі Турбо Паскаль.
2. Вивчити функціональне призначення клавіш та роботи з текстовим редактором.
3. Ознайомитися та вивчити алфавіт мови Паскаль, правила запису ідентифікаторів та службових зарезервованих слів.
4. Для вказаного викладачем варіанту завдання з табл.1.1 вказати вірно або не вірно записані ідентифікатори.

Контрольні запитання

1. Що входить до алфавіту мови Паскаль? Які ви знаєте спеціальні та складені символи?
2. Що таке зарезервовані слова, стандартні ідентифікатори та ідентифікатори користувача?
3. Які базові правила запису ідентифікаторів Вам відомі?

Таблиця 1.1 - Варіанти завдань до виконання контрольного завдання теми 1

Варіант	Завдан- ня	Варіант	Завдан- ня	Варіант	Завдан- ня	Варіант	Завдан- ня
1	<i>x05</i> <i>-256A</i> <i>2PY</i> <i>D00r</i>	6	<i>'TIP'</i> <i>IA6</i> <i>D5/h</i> <i>vjrf</i>	11	<i>-hg</i> <i>RRRR</i> <i>(FF)c</i> <i>G,F</i>	16	<i>Aed</i> <i>+56ee</i> <i>Gd_kl</i> <i>-2fgl</i>
2	<i>-R4</i> <i>F000</i> <i>C16-1</i> <i>Ili2i3i4</i>	7	<i>S123</i> <i>""df""</i> <i>-H</i> <i>77GG</i>	12	<i>DDDyt</i> <i>FOst</i> <i>R0.CX</i> <i>(ert)</i>	17	<i>[23gg]</i> <i>D15</i> <i>-bfd</i> <i>0A1</i>
3	<i>(i)</i> <i>Ir22</i> <i>Brevno</i> <i>Ir123</i>	8	<i>Irum</i> <i>Mode</i> <i>'dnk'</i> <i>0A1</i>	13	<i>Stop</i> <i>D6/9</i> <i>G1</i> <i>x-t</i>	18	<i>Df/pi</i> <i>lerp</i> <i>F15</i> <i>S057</i>
4	<i>V3w2</i> <i>24h18</i> <i>-der</i> <i>(dfr)</i>	9	<i>Slo/vo</i> <i>G95</i> <i>'-dd</i> <i>S132</i>	14	<i>\dfg</i> <i>Err</i> <i>Free</i> <i>lk12</i>	19	<i>J27-1</i> <i>Kwe</i> <i>-123</i> <i>F 95</i>
5	<i>J123</i> <i>-fff</i> <i>E12+05</i> <i>Dr_56</i>	10	<i>0R95</i> <i>Vvv_5</i> <i>-ewe</i> <i>Q12</i>	15	<i>>pl</i> <i>K397</i> <i>-vf</i> <i>As_100</i>	20	<i>L15l</i> <i>-fds</i> <i>//15</i> <i>G_yi</i>

Тема 2 Константи та змінні. Типи даних

Мета виконання завдання – вивчення вживання констант та змінних при програмуванні на мові Паскаль. Визначення типів даних: скалярних та структурованих, цілочислених та речовинних, символьних та булевських. Розглядання типів даних користувача — перелічених та інтервальних.

Методичні вказівки

Константи та змінні визначаються ідентифікаторами (іменами), за допомогою яких до них можна звертатися для одержання їх поточних значень. Константи – елементи даних, значення яких встановлено у описовій частині програми. У процесі виконання програми константи не змінюються. Для визначення констант служить зарезервоване слово – Const.

Формат: Const

<ідентифікатор> = <значення константи>;

Приклад:

Const

Krr = 555; *AIC* = 34.54; *WWW* = 'стрічка 33'.

Константи True, False, Maxint попереднього визначення не потребують. Вони приймають значення true, false, 32767 відповідно.

Змінні на відміну від констант можуть приймати різні значення. Кожна змінна та константа належить до визначеного типу даних. Тип констант не потребує описування. Навпаки, тип змінних потребує описування поперед тим, як із змінними будуть виконуватися які-небудь дії. Для цього призначено зарезервоване слово Var.

Формат: Var

<ідентифікатор> = <тип>;

Приклад: Var

St1, *sg2*, *sl3*: real;

Окрім констант та змінних існують типізовані константи, які у прикладному аспекті рівнозначні змінній з поперед ініціалізованим значенням. В програмі усі дії з нею можуть здійснюватися як із змінною.

Формат: Const

<ідентифікатор> : <тип> = <значення>;

Приклад:

Const

Ferrit: word = #105;

Тип – це множина значень, які можуть приймати об'єкти програми, та сукупність операцій, допустимих над цими значеннями. Для описування типів використовують зарезервоване слово – Type.

Формат: Type

<Ім'я типу> = <значення типу>;

Усі дані підрозділяють на дві групи: скалярні (звичайні) та структуровані (складні). Скалярні типи у свою чергу підрозділяються на стандартні та типи користувача. До стандартного типу даних відносять цілочисленні, речовинні, літерні, булевські типи даних та показники.

У десятинній системі числа можуть бути записано двома способами: з фіксованою та з плаваючою крапкою. Речовинні числа з фіксованою крапкою записують за звичайними правилами арифметики. Ціла частина від дрібної відокремлюється десятинною крапкою. Якщо десятинна крапка відсутня, тоді число вважається цілим. Поперед числа може знаходитися знак "+" або "-". Якщо знак відсутній, тоді за умовчанням число вважається позитивним. Речовинні десятинні числа з плаваючою крапкою подаються у експонентному вигляді: $mE+p$, де m – мантиса (ціле або дрібне число з десятинною крапкою), " E " – означає "десять у степені", p – порядок (ціле число).

Приклад:

255 - ціле десятинне число; \$1FF - шістнадцятирічне число;

378.912 - речовинне число; -15.05 - негативне речовинне число;

$3.14E+02 = 3.14 \cdot 10^2 = 314$; $100E-04 = 100 \cdot 10^{-4} = 0.01$.

Типи даних користувача - перераховуємий та інтервальний розробляються програмістом. До структурованих типів даних відносять рядки, масиви, множина, запис, файли та дані процедурного типу або типу object. Цілочислені типи даних уявляють собою значення, що можуть використовуватися у арифметичних виразах та займати у пам'яті від 1 до 4 байт. Діапазон дії та пам'ять, що потребується подана у таблиці 1.2.

Таблиця 1.2 – Цілочислені дані

Тип	Діапазон	Пам'ять (байт)
byte	0 ... 255	1
shortint	-128 ... 127	1
integer	-32768 ... 32767	2
word	0 ... 65535	2
longint	-2147483648 2147483647	4

Приклад: Var

Y1, Y2, Y3: byte;

Z1, Z2, Z3: integer;

U1, U2: word;

Речовинні типи даних уявляють собою речовинні значення, що використовують у арифметичних виразах та займають у пам'яті від 4 до 6 байт. Речовинні значення можуть бути у вигляді як з плаваючою крапкою, так і з фіксованою крапкою. Речовинні типи подано у таблиці 1.3.

Таблиця 1.3 – Речовинні дані

Тип	Діапазон	Мантиса	Пам'ять (байт)
real	$2.9 \cdot 10E-39 \dots 1.7 \cdot E+38$	11-12	6
single	$1.5 \cdot 10E-45 \dots 3.4 \cdot E+38$	7-8	4
double	$5.0 \cdot 10E-324 \dots 1.7 \cdot E+308$	15-16	8
extended	$1.9 \cdot 10E-4951 \dots$ $1.1 \cdot E+4932$	19-20	10
comp	$-2.E+63+1 \dots 2.E+63-1$	10-20	8

Приклад: Var

SA, SB, SC: real;

AS, BS, CS: double;

Літерний (альтернативний) тип визначається множиною значень кодової таблиці ПЕОМ. Для змінної літерного типу треба один байт. Він описується зарезервованим словом `char`.

Приклад:

`Var`

`fgg: char;`

`kgg: char;`

У програмі значення змінних та констант типу `char` повинні братися у апострофи. Наприклад, `'A'` позначає літеру А, `'.'` – кому з крапкою.

Булевський тип має два значення: `True` (істина) та `False` (неправда). Цій тип широко застосовується у логічних виразах і виразах відношення. Булевський тип описується зарезервованим словом `Boolean` і займає один байт пам'яті.

Приклад:

`Var`

`Free1, Free2: Boolean;`

З типами даних користувача – перечисленими та інтервальними, а також з структурованим типом даних можна ознайомитися у посібниках [1-4].

Порядок виконання контрольного завдання до теми 2

1. Ознайомитися з константами та змінними даними у середовищі Турбо Паскаль.
2. Вивчити описання типів даних у середовищі Турбо Паскаль.
3. Ознайомитися та вивчити типи даних та діапазон їх зміни.

Контрольні запитання

1. Яка різниця між константами та змінними у мові Паскаль?
2. Що таке тип даних? Які Вам відомі типи даних?

3. Які бувають скалярні типи даних? Наведіть приклади.
4. Що таке тотожність та сумісність типів? Наведіть приклади.

Тема 3 Вираження, операнди та операції, оператор присвоєння

Мета роботи

Мета роботи – вивчення порядку уживання виражень, операндів та виконання операцій. Будуть розглянуті операції відношення, пріоритети вживання операцій, а також оператор присвоєння.

Методичні вказівки

Вираження завдають порядок виконання дій над елементами даних і складаються з операндів (констант, змінних, звернень до функцій), круглих дужок та знаків операцій. Операції визначають дії, що треба виконати над операндами. Круглі дужки ставлять так само, як у арифметичних вираженнях. Операції у Паскалі підрозділяються на арифметичні, відношення, логічні, операцію `@`, рядкові та інші. Відповідно з цим класифікують вираження. Операції можуть бути бінарними та унітарними. У першому випадку операція виражає відношення між двома операндами і записується проміж ними. У другому відноситься до одного операнда і записується поперед нього.

Арифметичні операції виконують арифметичні дії у вираженнях над значеннями операндів цілочислених та речовинних типів. До цих операцій відносять операції складання (+), віднімання (-), множення (*), ділення (/) та інші. ([1]), що виконуються так само, як у звичайних арифметичних вираженнях.

Вираження відношення визначають істинність або хибність добутку. Операції відношення виконують порівняння двох операндів та знаходять, чи істинне значення вираження або хибне. Перелік операцій відношення подано у таблиці 1.4.

Таблиця 1.4 – Операції відношення

Операція	Назва	Вираження	Добуток
=	дорівнює	$A = B$	True, якщо A дорівнює B
<>	не дорівнює	$A <> B$	True, якщо A не дорівнює B
>	більш	$A > B$	True, якщо A більш B
<	менш	$A < B$	True, якщо A менш B
> =	більш або дорівнює	$A > = B$	True, якщо A більш або дорівнює B
< =	менш або дорівнює	$A < = B$	True, якщо A менш або дорівнює B
in	належність	$A \text{ in } M$	True, якщо A знаходиться у списку M

З порядком використання булевських операцій та операції типу @ можна ознайомитися у посібнику [2]. Виконання кожної операції відбувається з урахуванням пріоритету. Значення пріоритетів доведені у таблиці 1.5.

Таблиця 1.5 – Порядок виконання пріоритетів

Операції	Пріоритет	Вид операції
&, not	Перший (вищий)	Унітарна
*, /, div, mod, and, shl, shr	Другий	Операції типу множення
+, -, or, xor	Третій	Операції типу складання
=, <, >, <=, >=, in	Четвертий (нижчий)	Операції відношення

Для визначення пріоритетності операцій використовують базові правила:

- операнд, що знаходиться між двома операціями з різними пріоритетами, пов'язується з операцією, що має більш високий пріоритет;
- операція, що знаходиться між двома операціями з рівними пріоритетами, пов'язується з операцією, що знаходиться ліворуч;
- вираження, що записано у дужках, перед виконанням обчислюється як окремий операнд;

- операції з рівним пріоритетом обробляються зліва направо з можливим регулюванням порядку виконання дужками.

Оператор присвоювання ($:=$) приписує виконувати вираження, завдане в його правій частині, та присвоїть добуток змінної ідентифікатору, який розташовано у лівій частині. Змінна та вираження повинні мати один й той же тип, за виключенням випадку, коли змінна має речовинний тип, а вираження - цілочислений. Припускається присвоювання будь-яких типів даних окрім файлових.

Формат: <ідентифікатор>:=<вираження>;

Приклад:

$F:= 10$; $ddd:= 124.67$; $Rew:= 'Auto_600'$;

$R:= (a+b)/c-f*a/b+(a+c)/f$.

Порядок виконання контрольного завдання до теми 3

Для вказаного викладачем варіанту завдання до роботи 3 (табл.1.6) записати його на мові Паскаль з використанням оператора присвоювання та розписати послідовність виконання операцій.

Таблиця 1.6 – Варіанти до контрольного завдання до теми 3

Варі-ант	Завдання	Варі-ант	Завдання
1	$f = ((b^2 - 1)^2 - \sqrt{a^2 - b^2} / (b - 1)^2 + 3)^2$	11	$f = ((1 - b^2) bc / a + c^2) + 5$
2	$f = ((a - 1)^2 / c - 1) + (a + c / (c - 1)^2)$	12	$f = (1 - bc / (b - 1)) + (\sqrt{a - 1} / bc)$
3	$f = (1 - a^2 / 5) + (\sqrt{a + b} / (3 + b)^2 - 1)$	13	$f = (\sqrt{b + 1} + ab / c^2 - b) + 3$
4	$f = (1 - b^2)^2 + ((abc - 1) / (a^2 + b^2))$	14	$f = (ab + c)^2 + \sqrt{(ab + 1) / c}$
5	$f = (\sin(a - 1) + bc + 4) / ((a - 1)^2 + 1)$	15	$f = (1 - a^2 / bc) + (1 + a^2 / bc - 5)$
6	$f = a^2 + \sin(a + 1) + \sqrt{a} / bc$	16	$f = (1 - c^2)^2 + (1 - a^2 / (1 + a + b))$
7	$f = (1 - b^2)^3 + \sqrt{a - 1} / c + 10$	17	$f = (1 + c^2 / b) + \sqrt{b - a} / ((c - b)^2 + 1)$
8	$f = 1 + b^2 + c^2 + \sqrt{b - a} / (1 + b^2)$	18	$f = ((ab / c - 1) / (a^2 - b^2) + c^2) + 1$
9	$f = ((ab - 1)^2 / c - 1) + (a + c^2 / \sqrt{c - 1})$	19	$f = (\sqrt{ab - b^2} / a^2 + (\sqrt{c^2 - b} / \sqrt{ac^2 + b}))$

10	$f = (\sqrt{a} + 1 / (a + 1) + (bc - 1) / c)$	20	$f = (ab + c^2) - \sqrt{(a^2b + 1) / c}$
----	---	----	--

Приклад виконання завдання: завдано арифметичне вираження:

$f = ((a^2 + 1)^2 + (b - 1) / a - b * c) + 10$. На мові Паскаль це вираження можна записати у вигляді:

$$f := (sqr(sqr(a) + 1) + (b + 1) / a - b * c) + 10.$$

Порядок виконання операцій: спочатку виконуються операції у дужках - возведення у квадрат змінної a , потім складання з 1 і возведення у квадрат. Далі підраховується величина $b - 1$ і ділиться на a , потім b множиться на c і усі три додатки складаються, після чого додається число 10 і все присвоюється змінної f .

Зміст звіту

Запис завдання на мові Паскаль з використанням оператора присвоювання та послідовність виконання операцій з поясненнями.

Контрольні запитання

1. Які операції мови Паскаль Вам відомі? Стисло опишіть їх.
2. Що таке унітарні та бінарні операції?
3. Опишіть арифметичні вираження. Які арифметичні операції Вам відомі та як вони виконуються?
4. Опишіть вираження відношення. Які операції відношення Вам відомі та як вони виконуються?
5. Що таке пріоритет операцій? Які базові правила визначення пріоритету операцій Вам відомі?
6. Як працює оператор присвоювання? Які правила користування та виключення Вам відомі? Наведіть приклади.

Тема 4 Структура програми

Мета роботи

Мета роботи – вивчення побудови програми. Розглядаються усі розділи, синтаксис написання та деякі рекомендації до складання програм на Паскалі. Викладені свідомості про застосування операторів вводу — виводу даних.

Методичні вказівки

Програма об'єднує послідовність дій, що виконуються над даними за допомогою визначених операцій, що до реалізації заданого алгоритму. Базові характеристики програми: точність одержаних здобутків, час виконання та потрібна пам'ять. Програма на Паскалі складається з рядків, максимальна довжина яких не перевищує 127 символів. При наборі тексту за допомогою редактора рядки на екрані можна розташовувати довільно. Рядок може починатися з будь-якої колонки. Кількість операторів у рядку довільна. При написанні програми можна рекомендувати, наприклад, таку схему: зарезервовані слова Program, Procedure, Function у будь-якому випадку починаються з великої букви; імена констант, змінних, процедур тільки маленькими; оператори записуються на 1 рядок нижче та на 1-2 позиції правіше відносно до попереднього. Редактор і компілятор обробляють програми та модулі не більш 64 Кбайт. Якщо є потреба обробляти більші програми, то для цього використовуються спеціальні бібліотечні модулі (.TPU - файли) або оверлейні структури ([1-4].

Синтаксично програма складається з обов'язкового заголовку і блока. Блок може містити у себе інші блоки. Кожен блок складається з двох частин: описової та виконавчої. Перша частина може бути відсутня, без другої блок не має розуміння. Блок, що не входить ні в який другий блок, має назву глобального. Якщо у глобальному блоку знаходяться інші блоки, то вони мають

назву локальних. Глобальний блок – це головна програма, вона повинна бути завжди, локальні блоки – це процедури та функції, їх присутність необов'язкова. Об'єкти програми (типи, змінні, константи та інше.) відповідно також мають назву глобальних або локальних.

Програма починається із заголовку, що складається у загальному випадку із зарезервованого слова Program, ім'я програми та параметрів, за допомогою яких програма взаємодіє з операційною системою. Заголовок може бути відсутній, але його рекомендується вживати. Параметрами програми звичайно є стандартні ідентифікатори стандартних файлів вводу – виводу Input та Output:

```
Program Trees (Input, Output);
```

```
Program Beer (Input, Output).
```

Після заголовку записується програмний блок, що складається у загальному випадку з 7 розділів: списку імен бібліотечних модулів (він визначається зарезервованим словом Uses), що підключаються, описання міток, описання констант, визначення типів даних, описання змінних, описання процедур та функцій, операторів. Структура програми така:

```
Program <ім'я> (Input, Output);
```

```
Uses (ім'я_1, ім'я_2,...);
```

```
Label ...;
```

```
Const ...;
```

```
Type ...;
```

```
Var ...;
```

```
Procedure <ім'я>;
```

```
<тіло процедури>
```

```
Function <ім'я>;
```

```
<тіло функції>
```

```
Begin
```

```
<оператори>
```

End.

Любий розділ, окрім розділу операторів, може бути відсутнім. Розділ *Uses* завжди записується після заголовку програми. Інші розділи можуть зустрічатися у програмі будь-яку кількість разів та у довільному порядку. Головне, щоб усі описання об'єктів були зроблені до того, як вони будуть використовуватися.

Розділ Uses. Цій розділ складається із зарезервованого слова *Uses* списку імен стандартних бібліотек, що підключаються та бібліотек користувача.

Формат: *Uses* <ім'я_1>, <ім'я_2>,...;

Наприклад: *Uses* Dos, Lib_1, Lib_S;

Розділ описування міток. Поперед будь-якого оператора мови Паскаль можна поставити мітку, на яку можна перейти за допомогою оператора безумовного переходу *goto*. Мітка складається з імені та двокрапки, що слідує за нею. Ім'я мітки - ідентифікатор або цифра, не більш як 127 символів. Мітки повинні бути описані. Розділ описання починається із зарезервованого слова *Label*, за яким слідують імена міток через кому.

Формат: *Label* <ім'я,...>;

Наприклад: *Label* V1, M123, 1080;

Після запису мітки у розділі операторів (тобто у виконавчій частині програми) слідує двокрапка, що є відзнакою мітки від ідентифікатора:

Label L1, L2; (описання міток)

Begin

...

L1: <оператор> (використання L1 у розділі операторів)

...

L2: <оператор> (використання L2 у розділі операторів)

End.

Мітки треба описувати, але можна не використовувати.

Розділ описання констант. Розділ розпочинається зарезервованим словом Const, за яким слідує вирази, що надають ідентифікаторам постійні числові або рядкові значення. Вираження відокремлюють один від одного крапкою з комою.

Формат: Const <ідентифікатор> = <значення>;

Приклад:

Const

Minimax: word = 1000; (Типізована константа)

Zag_1 = ' описання констант '; (Рядкова константа)

RR = 19.35; (Речовинна константа)

У Turbo Pascal більш констант визначені стандартно. До них можна звертатися без попереднього описування.

Розділ описання типів даних. Тип даних може бути описано або безпосередньо у розділі описання змінних, або визначити ідентифікатором типу. Стандартні типи описання не потребують, на відзнаку від типів утворених користувачем. В загальні Паскаль не потребує обов'язково визначення ідентифікатора типу і в останньому випадку, враховуючи що тип можна завдати перерахуванням у розділі описання змінних.

Розділ описання типів даних починається зарезервованим словом Type, за яким сліdkують одне або декілька визначень типів, відокремлених крапкою з комою.

Формат: Type <ім'я типу> = <значення типу>;

Приклад:

Type

ABC = array [1..50] of real;

Dni = 1.31;

Розділ описання змінних. Описання змінних обов'язкове і воно починається зарезервованим словом Var, потім через кому перечислюються імена змінних і через двокрапку слідує їх тип та крапка з комою.

Формат: Var <ідентифікатор> : <тип>;

Приклад:

Var

I, J, K: integer;

A, B, C: real;

L, M, N: boolean;

Розділ описання процедур. Буде розглянуто більш докладно далі.

Розділ описання операторів. Це базовий розділ у якому виконуються потрібні дії для одержання результату. Він починається зарезервованим словом Begin (початок), далі сліdkують оператори мови, відокремлені один від одного крапкою з комою. Завершує розділ зарезервоване слово End (кінець) і крапка.

Begin

<оператор;>

.....

<оператор>

End.

Оператори виконуються у той послідовності, в якій вони були записані відповідно з синтаксисом та правилами пунктуації. Традиційно слова Begin і End використовують у глобальному блоку, а begin і end - у локальних блоках.

Коментар. Коментар – це пояснювальний текст, що записується у будь-якому місці програми, де дозволено пробіл. Текст коментаря обмежено

символами { } та (* *). Він може містити у себе будь-які комбінації латинського та російського алфавіту, цифри та інші символи алфавіту мови Паскаль.

Приклад: { Коментар до програми }
(* Головна програма *)

У обмежувачах (* *) пробіл між дужками та зірочкою заборонено. У тексті коментаря не можуть знаходитися знаки обмежувачів, з яких коментар починається. Однак, наприклад обмежувач { } може бути серед тексту коментаря (* *). Обмежень на довжину тексту коментаря немає.

Процедури вводу – виводу даних. Для вводу - виводу даних використовується 4 оператора: read, readln, write, writeln. Вони мають однаковий формат у вигляді:

read (або readln, write, writeln) (x1,x2,...,xn);

де $x_1, x_2, \dots, x_n$ – змінні допустимих типів даних. Ці значення набираються мінімум через пробіл на клавіатурі та висвічуються на екрані. Після набору даних одного оператора типу read або readln натискається клавіша <Enter>. Значення повинні уводитися у строгої відповідальності з синтаксисом мови Паскаль. У протилежному випадку виникають помилки вводу – виводу. При цьому повідомлення має вигляд: I/O error XX, де XX - код помилки. Оператор readln діє аналогічно read, різниця міститься у тому, що після того як буде прочитане останнє значення для одного оператора readln, для кожного другого дані будуть читатися з початку нового рядку. Оператори write та writeln аналогічні read та readln, тільки вони роблять вивід числових даних, символів, рядків та булевських виражень.

Приклад: Var
A,B,C: integer;
X,Y,Z: real;

```

...
readln (A,B);
C:= A+B;
readln (X,Y);
Z:= X+Y;
..

```

Оператори `write` та `writeln` мають можливість запису виражень, визначаючих ширину поля виводу, з чим можна більш докладно ознайомитися у підручнику [2].

Порядок виконання контрольного завдання до теми 4

1. Для даних з табл. 1.7 вказаного викладачем варіанту завдання написати програму вводу – виводу на Паскалі з використанням операторів `readln` та `writeln`.
2. Для вказаного викладачем варіанту завдання з табл.1.6 написати текст програми на Паскалі, ввести значення з табл.1.8 і одержати результат.
3. Для вказаного викладачем варіанту завдання з табл.1.9 написати текст програми на мові Паскаль обчислення функції $f(x,y,z)$, де $z=g(x,y)$; ввести значення і одержати результат.

Таблиця 1.7 - Варіанти завдання до використання операторів вводу - виводу

Варіант	Завдання	Варіант	Завдання	Варіант	Завдання
1	25,14	7	213	13	333,333
	0,01221		100000		-9999
	-13		-10000,0		751,003
	4444		56,06		-5,6
2	34,345	8	-21,4	14	21,021
	-100000		256		-1765
	-0,0006		0,00028		6

	753,8		12345		0,0009
3	1000000	9	123,321	15	-0,98756
	99,9999		2,33333		341
	-6		-0,1		45,879
	6789		42356,1		6789
4	123456	10	342,456	16	23,321
	-234,45		-756,980		2,3333
	0,0007		-5678		-0,105
	-2222		0,000124		2356,1
5	678	11	14,9	17	3123,321
	-12457		-8888		22,33333
	0,0006		-1122		-10,1
	-369,78		565,65		42,1
6	53,05	12	1,0	18	-123,321
	-341,9		675		0,33333
	-22222		753,008		1110,1
	100,001		-5,8		56,1

Таблиця 1.8 – Значення змінних для виконання завдання з табл.1.6

Варіант	a	b	c	Варіант	a	b	c
1	4,2	1,7	2,2	11	3,3	7,2	6,3
2	14,2	0,7	9,2	12	4,2	3,1	7,8
3	4,1	3,9	1,7	13	1,1	3,9	8,2
4	0,6	1,9	2,7	14	1,6	6,2	8,8
5	2,0	3,2	8,0	15	0,9	0,4	3,1
6	0,7	1,8	4,6	16	3,2	7,4	5,1
7	2,4	9,9	3,3	17	2,1	5,7	8,0
8	1.2	3,4	13,3	18	1,1	2,0	3,6
9	0.3	1,5	5,5	19	6,4	4,0	1,6
10	8,5	7,7	2,9	20	4,5	5,4	3,3

Таблиця 1.9 – Значення даних для виконання завдання

Варіант	$f(x, y, z)$	$z = g(x, y)$	a	b
1	$ x + zy - z^2 / \sqrt{8 + \cos^2 y}$	$2xy / (x + \cos y)$	1,12	0,87
2	$\sqrt{\frac{y}{x^2 + 1}} + e^{-z}$	$\sqrt{x + 2y} \cdot \sin x^2$	6,55	2,78

3	$\frac{x+x^2+\cos x}{y(y-z)+\ln x}$	$\frac{\sqrt{15y}}{y+\operatorname{ctgx}}$	1,88	3,42
4	$2^{-x}\sqrt{y+\sqrt[4]{z}}$	$\ln(x^4+y)$	-0,85	1,25
5	$\frac{1+\cos^2(x+z)}{ x^3-2\ln\sqrt{y} }$	$\frac{\sqrt{x}\sin^2 y}{x+e^y}$	1,54	3,26
6	$\frac{y+\sqrt{y^2+x^2}}{(z+x)e^x}$	$\frac{\sin(x/y)}{2x^2}$	-2,77	-1,48
7	$\frac{y\ln x-zx^3}{1+\operatorname{tg}^2(x^2)}$	$\frac{\operatorname{arctgx}}{x+y}$	0,47	-0,67
8	$\sqrt{ y }e^{-(x+y)}+\cos(z^3)$	$\frac{x\sqrt{x^3+y}}{x-1}$	1,52	0,75
9	$e^{x-1}+\arcsin(y^2/x)$	$2\sin(\pi x+y)$	3,24	-1,54
10	$\ln(\sqrt{x}+\sqrt{y})\cdot z^3$	$\frac{\cos^2\pi(2+x)}{4-y^2\sqrt{x}}$	3,44	4,82
11	$\cos x+2\operatorname{tg}^2(z/y)$	$\sqrt{x}\sin y$	0,24	-1,78
12	$4y^2\ln xz $	$\ln(\sqrt[4]{4x^4+4y^4})$	3,24	-1,74
13	$\frac{z^3}{x+y^3/(x+z^2)}$	$\frac{ y+8x }{\sin x+\operatorname{tgy}}$	-0,72	-1,42
14	$e^{-x}(\operatorname{tg}^2 z)^y$	$2xy/(x+\cos y)$	-4,24	0,57
15	$\frac{x\ln y-zy^3}{1+\operatorname{tg}^2(y^2)}$	$\frac{x\sqrt{x^2+y^2}}{x-1}$	2,67	3,42
16	$\frac{x+\sqrt{y^2+x^2}}{(z+y)e^x}$	$\frac{\cos(x/y)}{2x^2}$	0,34	0,17
17	$\frac{1+\sin^2(y+z)}{ y^3-2\ln\sqrt{x} }$	$\frac{\sqrt{y}\sin^2 x}{y+e^x}$	3,26	1,54
18	$\frac{y+y^2+\cos x}{x(x-z)+\ln y}$	$\frac{\sqrt{15x}}{x+\operatorname{ctgy}}$	3,42	1,88
19	$4x^2\ln yz $	$\ln(\sqrt[4]{4x^4+4y^4})$	-1,74	3,24
20	$\frac{\sqrt{x^2+y^2}}{(z+y)e^y}$	$\frac{\sin(x/y)}{2x^2}$	-2,77	-1,48

Приклад виконання завдання. Знайти площу трикутника, використовуючи формулу Герона $S = p(h-a)(p-b)(p-c)$, де a, b, c - сторони трикутника, $p = (a + b + c)/2$ – напівпериметр.

```

Program PPP (Input, Output);
(* Програма обчислення площі трикутника за формулою Герона *)
Var
a, b, c, p, s: real;
Begin
(* Введення даних a,b,c *)
readln (a,b,c);
(* Друкування даних a,b,c *)
write (a,b,c);
(* Значення напівпериметру *)
p := (a + b + c)/2.0;
(* Значення площі трикутника *)
s:= sqrt (p*(p-a)*(p-b)*(p-c));
(* Друкування значення площі *)
write (' площа трикутника дорівнює =', S);
End.

```

Зміст звіту

1. Текст програми на мові Паскаль використання операторів вводу – виводу з табл.1.7.
2. Текст програми на мові Паскаль і результати розрахунків функції з табл.1.6.
3. Текст програми на мові Паскаль і результати розрахунків з табл.1.9.

Контрольні запитання

1. З яких базових синтаксичних розділів складається програма? Що таке глобальний та локальний блок?
2. Стисло опишіть поняття мітки та розділ описання міток.
3. Опишіть розділи описання констант типів даних та змінних.

4. Які правила запису розділу операторів? Що таке коментар? Де, як і коли використовується коментар у програмі?
5. Які процедури вводу - виводу Ви знаєте? Стисло опишіть їх формат.
6. Що таке формат виводу? Як він використовується?
7. Які правила виводу виражень речовинного типу, цілочислених виражень та виражень рядкового, булевського і символьного типу?

РОЗДІЛ 2. ОПЕРАТОРИ БЕЗУМОВНОГО ТА УМОВНОГО ПЕРЕХОДУ

Мета роботи

Мета роботи – вивчення і використання при написанні програм операторів безумовного та умовного переходу та їх застосування у конкретних прикладах.

Методичні вказівки

Оператор безумовного переходу. Означає «перейти до» і використовується у випадках, коли після виконання деякого оператора треба виконати не наступний по порядку, а якийсь інший, відзначений міткою оператор. Область дії мітки є тільки той блок, у якому вона описана. Через те перехід за безумовним оператором дозволяється у межах блока мітки.

Формат: `goto <мітка>;`

Приклад: . . .

Label M1,M2;

. . .

M1:

goto M2;

...

M2:

C := A + B;

...

Рекомендовано вживати цей оператор як можна ріже [2].

Оператор умовного переходу. Забезпечує виконання або невиконання деякого оператора, групи операторів або блока в залежності від завданих умов. Паскаль налічує два умовних оператора – if та case.

Оператор if найбільш частіше використовується у програмах. Він може приймати одну з таких форм:

1. if <умова> then <оператор 1> else <оператор 2>;
2. if <умова> then <оператор>.

Ключові слова if, then, else відповідально означають «якщо», «тоді», «інакше». Умова – це вираження булевського типу. Воно може бути простим або складним. Складні умови утворюють за допомогою логічних операцій and, or, not та інші. Результат вираження завжди має булевський тип. При цьому у першому випадку, якщо значення вираження True (істина), тоді виконується <оператор 1>, якщо False (невірно) – <оператор 2>. У другому варіанті формату, якщо результат True, виконується <оператор>, якщо False – оператор, що слідує за оператором if.

Наприклад:

if A > B then S:= A + B else S:= A - B;

if I = J then goto M100;

Оператор if може входити до складу другого оператора if. У такому випадку говорять про вкладеність операторів.

Приклад:

```
if I >= J then begin
```

```
s:= x + y;
```

```
writeln (s);
```

```
end
```

```
else begin
```

```
s:= x*y;
```

```
writeln (s);
```

```
end;
```

Оператор case є узагальненням оператора if. Він дозволяє зробити вибір з вираження, що має назву селектор. Більш докладно з роботою цього оператора можна ознайомитись у підручниках [1-4].

Порядок виконання контрольного завдання

Для вказаного викладачем варіанту завдання наданого у табл.2.1 написати програму з використання умовного оператора if і одержати результати її виконання.

Таблиця 2.1 – Варіанти завдання

Варіант	Умови завдання	
1	2	3
1	$f = \begin{cases} ab - y, & y < 0 \\ (ay + b^2) / 2, & y = 0 \\ ay + 1, & y > 0 \end{cases}$	$a = 0,2;$ $b = 0,01$
2	$f = \begin{cases} c - ba^2, & a > 0 \\ a - (b/2 + 1)^2, & a = 0 \\ a - 1, & a < 0 \end{cases}$	$b = 1,2;$ $c = 1,71$
3	$f = \begin{cases} 0,5y - 1, & y = 1 \\ (1 - y^2) / a, & y > 1 \\ (1 + y^2), & y < 1 \end{cases}$	$a = 18,12$

4	$f = \begin{cases} 1-x^2y, & y < 3 \\ 1+x^2y, & y = 3 \\ 1-2xy, & y > 3 \end{cases}$	$x = 0,812$
5	$f = \begin{cases} 1-y^2, & y = 1 \\ (1+y^2)/a, & y < 1 \\ 1+3.5y, & y > 1 \end{cases}$	$a = 0,412$
6	$f = \begin{cases} (1-x^3)/2, & x < 0 \\ (1+2x)/(a-2), & x > 0 \\ 1, & x = 0 \end{cases}$	$a = 6,31$
7	$f = \begin{cases} 1-x^2, & x = 0 \\ (1+xb)/a, & x < 0 \\ (1+x)/0.5, & x > 0 \end{cases}$	$a = 7,2;$ $b = 0,01$
8	$f = \begin{cases} 3,1a-x^2, & x = 1 \\ 5,2b-x, & x > 1 \\ x-1, & x < 1 \end{cases}$	$a = 5,2;$ $b = 1,6$
1	2	3
9	$f = \begin{cases} 3.1x^2+b, & x > 3 \\ (x-1)/2, & x < 3 \\ (x+b)^2, & x = 3 \end{cases}$	$b = 3,21$
10	$f = \begin{cases} (x^3-1)/(a-1), & x = 5 \\ (x+1)^2, & x > 5 \\ (x-1)^2, & x < 5 \end{cases}$	$a = 62,1$
11	$f = \begin{cases} (1-x^2)/(c+1), & x > 0 \\ (1-x)^2b, & x < 0 \\ (1-x)c^2, & x = 0 \end{cases}$	$b = 1,2;$ $c = 6,62$
12	$f = \begin{cases} (1+x)^2b, & x = 10 \\ b-x^2, & x < 10 \\ b+x^2, & x > 10 \end{cases}$	$b = 6,531$
13	$f = \begin{cases} y^2-1, & y > 1 \\ \ln(ab), & y = 1 \\ ((y^2+1)b)/(a-b), & y < 1 \end{cases}$	$a = 0,2;$ $b = 0,3$
14	$f = \begin{cases} x^2+b, & b < 0 \\ 0,5x-b, & b > 0 \\ yx+b, & b = 0 \end{cases}$	$x = 3,12;$ $y = 0,5$
15	$f = \begin{cases} x^2+1, & x > 1 \\ ax+b, & x = 1 \\ x+(b/a), & x > 1 \end{cases}$	$a = 0,01;$ $b = 5,1$

16	$f = \begin{cases} (1+x)^2 / b, & x=1 \\ b+x^2, & x<1 \\ b/x^2, & x>1 \end{cases}$	$b = 12$
17	$f = \begin{cases} (a+x)^2 / a, & x=3 \\ ab+x^2, & x<3 \\ a^2b/x^2, & x>3 \end{cases}$	$a = 2,4$ $b = 3,6$
18	$f = \begin{cases} (x^3+1)/(a+1), & x=4 \\ (x-1)^2, & x>4 \\ (2x+1)^2, & x<4 \end{cases}$	$a = 7,2$
19	$f = \begin{cases} x^2+b^2, & b<0 \\ 0,5x+b, & b>0 \\ yx, & b=0 \end{cases}$	$x = 1,12;$ $y = 1,5$
20	$f = a^2 + b^2, x \leq 0$ $f = (a^2 + b^2) / x^2, x > 0$	$a = 1,41$ $b = 3,1$

Приклад виконання завдання. Знайти значення функції Y , якщо при $x < 0$
 $Y = 2x - 1, 0 \leq x < 1$ $Y = 5x - 5, x \geq 1$ $Y = \sin(x) + 1$.

Program ННН (Input, Output);

Var x, y : real;

Begin

Writeln ('Ввести аргумент x');

Readln (x);

If x < 0 then Y := 2*x - 1 else

Begin

if x >= 1 then Y := sin(x) + 1

else Y := 5*x - 5

end;

Writeln ('Значення Y дорівнює ', Y:12:5);

End.

Зміст звіту

1. Текст програми на мові Паскаль для варіанту завдання.
2. Результат розрахунків з поясненням.

Контрольні запитання

1. Що таке оператор безумовного переходу? В яких випадках він використовується?
2. Які Ви знаєте умовні оператори? Наведіть формат оператора if.
3. У яких випадках корисно застосування оператора case?
4. Що таке умова? Який тип умов та які операції відношення використовуються при запису умови?
5. Як Ви розумієте поняття вкладеності умовних операторів? Наведіть приклади.

РОЗДІЛ 3. СТРУКТУРОВАНІ ТИПИ ДАНИХ. ПРОГРАМУВАННЯ АЛГОРИТМІВ З ІТЕРАЦІЙНИМ ЦИКЛОМ

Мета роботи

Мета роботи – вивчення структурованого типу даних масив та операцій над елементами масивів. Також розглядаються оператори циклів (повторів), їх застосування, правопис та використання при роботі над масивами даних.

Методичні вказівки

Масив – це структурований тип даних, що складається з фіксованого числа елементів одного типу. Можна використовувати будь-який раніш описаний тип, завдяки чому можуть існувати масиви записів, масиви рядків, масиви масивів та інше. Елементами масиву можуть бути дані будь-якого типу. Тип елементів масиву має назву базового. Число елементів фіксується при описанні та у процесі виконання програми не змінюється. Доступ до кожного елемента виконується завдяки індексуванню елементів масиву. Індекс масиву -

вираження любого скалярного типу, тільки не речовинного. Тип індексу визначає межі зміни значення індексу. Для описання масиву призначено словосполучення `array of`.

Формат:

Type

<ім'я типу> = array [тип індексу] of <тип компонент>;

Var

<ідентифікатор,...> : <ім'я типу>;

Масив може бути описано без представлення типа у підрозділі описання типів даних:

Var

<ідентифікатор,...>: array [тип індексу] of <тип компонент>;

Наприклад:

Var

A: array [1..120] of real;

K: array [1..20] of integer;

Якщо у якості базового масиву використовується інший масив, тоді одержану структуру називають багатомірним масивом. Наприклад:

Var

Matr: array [1..8,1..4] of real;

Коли у такий формі завдання один індекс, тоді масив зветься одномірним, якщо два – двомірним і так далі. Для описання масиву можна використовувати раніш визначені константи:

Const

I = 5; J = 10;

Var

Massiv : array [1..I,1..J] of real;

Для виконання дій над масивами як одним цілим оперують тільки з ідентифікатором масиву без указу індексу у квадратних дужках. Будь-який елемент масиву можна обробити вказав ім'я масиву та індекс елемента у квадратних дужках. При роботі з двомірними масивами треба вказати два індекси. Більш докладніше про роботу з масивами можна прочитати у підручнику [1-3]. Індексовані змінні можна використовувати як прості змінні, але найбільш зручно їх застосування у операторах циклу (повтору).

Цикл – це послідовність операторів, що може виконуватися більш одного разу. Якщо кількість повторів відома раніше, то використовується оператор `for`, якщо кількість повторів невідома, то застосовують оператори `repeat` або `while`.

Оператор циклу `for` складається із заголовка та тіла циклу. Він може бути поданий у двох форматах:

1. `for <параметр циклу> := <S1> to <S2> do <оператор>;`
2. `for <параметр циклу> := <S1> downto <S2> do <оператор>;`

де *S1* та *S2* – вираження, що визначають відповідно початкове та кінцеве значення параметра циклу. `for ... do` – заголовок циклу, `<оператор>` – тіло циклу. Тіло циклу може бути простим або складним оператором. Оператор `for` забезпечує виконання тіла циклу доки не будуть перебрані усі значення параметра циклу від початкового до кінцевого. Параметр циклу, його початкове та кінцеве значення, повинні належати до одного типу даних. При цьому припускається будь-який скалярний тип, тільки не речовинний. Якщо застосовуються типи `integer`, `byte` та інтервальний, то значення параметра циклу послідовно збільшується (при `for ... to`) або зменшується (при `for ... downto`) на одиницю на кожному повторі.

Наприклад:

`for I:= 1 to 20 do write (I:3);`

for I:= 20 downto 1 do write (I:3).

Результат виконання роботи операторів циклу і печатання значення I від 1 до 20 у першому випадку та від 20 до 1 у другому. У операторі for після do може знаходитися складний оператор. При цьому у його тілі не можуть знаходитися оператори, що змінюють значення параметра циклу. Після нормального завершення оператора for значення параметра циклу дорівнює кінцевому значенню, якщо оператор for не виконувався, то значення параметра циклу не визначено. У тілі циклу можуть знаходитися інші оператори for. Тоді такі внутрішні цикли мають назву вкладені.

Наприклад:

for I:= 1 to 20 do

for J:= 1 to 20 do

A [I,J] := 0; (обнулення елементів матриці A).

У операторі for Паскаль не допускає зміни параметра циклу на величину, відмінну від одиниці. Однак цей недолік компенсується використанням будь-якого кроку при організації циклів у операторах repeat та while .

Оператор повтору repeat складається із заголовка (repeat), тіла та умов закінчення (until).

Формат: repeat

<оператор;>

<оператор;>

...

<оператор>

until <умова>;

Умова – вираження булевського типу. При написанні умови припускаються булевські операції та операції відношення. Оператори, що знаходяться між словами repeat та until, являють собою тіло циклу. Спочатку

виконується тіло циклу, потім перевіряється умова виходу з циклу. Якщо результат булевського вираження False, тіло циклу активізується ще раз, якщо True – вихід з циклу.

Оператор repeat має три характерні особливості:

1. виконується не менш одного разу;
2. тіло циклу виконується, доки умова дорівнює False;
3. у тілі може знаходитися довільне число операторів без операторних дужок
begin ... end.

Приклад застосування оператора repeat:

```
Program Rep;  
Uses Crt;  
Var I, S : Integer;  
Begin  
I:= 0; S:= 0;  
repeat  
S:= S+I;  
I:= I+2;  
until (I>10);  
writeln (' Сума парних чисел дорівнює ', S)  
End.
```

Оператор while аналогічне оператору repeat, але перевірка умови виконується у ньому на початку оператора.

Формат: while <умова> do <тіло циклу>.

Умова – булівське вираження, тіло циклу - простий або складний оператор. Поперед кожного виконання тіла циклу обчислюється значення вираження умови. Якщо результат True, тіло циклу виконується, якщо False –

вихід з циклу та перехід до першого після while оператора. Як у операторі repeat програміст повинен сам здійснювати умови виходу з циклу. Оператор while, як і другі оператори циклу, може бути вкладеним. Приклад застосування оператора while:

```
I :=1;
while I <=100 do
begin
  B [I]:= I;
  I :=I + 1;
end;
```

Порядок виконання контрольного завдання

На алгоритмічній мові Паскаль для вказаного викладачем варіанту завдання з табл.3.1 написати текст програми із застосуванням операторів for, repeat, while (три програми).

Таблиця 3.1 – Дані для виконання завдання

Варіант	Обчислити значення за формулою	Варіант	Обчислити значення за формулою
1	2	3	4
1	$f = \sum_{i=1}^3 a_i - \sum_{i=1}^4 b_i$	11	$f = \sum_{i=1}^8 (a_i + b_i)^2$
2	$f = x + bx \cdot \sin x$ для $0 \leq x \leq 1$ з кроком 0,1	12	$f = \left(\sum_{i=1}^4 x_i y_i \right)^2$
1	2	3	4
3	$f = \sqrt{\sum_{i=1}^5 (y_i + x_i)}$	13	$f = \sum_{i=1}^4 y_i \sum_{j=1}^4 x_j$
4	$f = (x_i + y_i) / P_i$, де $i = 1, 2, \dots, 10$.	14	$f = \prod_{i=1}^4 (x_i + 0.5 y_i^2)$
5	$f = \sum_{i=1}^8 a_i + \sum_{j=1}^8 b_j$	15	$f = \sum_{i=5}^6 (\cos x_i \cdot \sin y_i)^2$

6	$f = \sum_{i=1}^4 \sum_{j=1}^4 a_{ij} + bx$	16	$f = \left(\sum_{i=1}^5 z_i \right)^{5/3}$
7	$f = \sum_{i=1}^3 \sqrt{\cos x_i \cdot \sin x_i}$	17	$f = \left(\sum_{i=3}^5 x_i \right)^2$
8	$f = \prod_{i=1}^4 \sqrt{(2x_i + y_i^2)}$	18	$f = \sqrt{\sum_{i=1}^4 \sum_{j=1}^4 a_{ij}}$
9	$f = \sqrt{\sum_{i=1}^3 y_i \sum_{j=1}^2 x_j}$	19	$f = x + \sqrt{bx \cdot \cos x}$ для $0 \leq x \leq 1$ з кроком 0,1
10	$f = \left(\sum_{i=1}^4 x_i y_i \right)^{3/2}$	20	$f = \prod_{i=1}^3 \sqrt[3]{x_i^2}$

Приклад виконання завдання: обчислити значення функції $Y = \sum_{i=1}^{20} a_i \cdot b_i$.

Program Abc;

Uses Crt;

Var Y: real;

a, b: array [1..20] of real;

i: integer;

Begin

(* введення даних - масивів а та в *)

for i:= 1 to 20 do read (a[i]);

for i:= 1 to 20 do read (b[i]);

(* обчислення значення Y *)

Y:= 0;

for i:=1 to 20 do Y:= Y + a[i]*b[i];

(* i:= 1; *)

(* repeat *)

(* Y:= Y + a[i]*b[i]; *)

(* i:=i + 1; *)

(* until i<21; *)

(* i:= 1; *)

(* while i <=20 do *)

(* begin *)

```
(*   Y: = Y + a[i]*b[i];   *)  
(*   i := i + 1;           *)  
(*   end;                 *)  
writeln (' Значення суми Y', Y:12:5);  
End.
```

Зміст звіту

Тексти програм на мові Паскаль з розрахунками для заданих викладачем вхідних даних.

Контрольні запитання

1. Що таке масив? Які типи масивів Ви знаєте? Як описують масиви?
2. Дії над масивами. Дії над елементами масивів.
3. Наведіть використання процедур вводу - виводу масивів.
4. Що таке цикл? Які оператори циклу (повтору) Ви знаєте?
5. Якій формат та принцип роботи оператора for? Наведіть приклади?
6. Якій формат та принцип роботи оператора repeat? Наведіть приклади?
4. Якій формат та принцип роботи оператора while? Наведіть приклади?

РОЗДІЛ 4. ЗАГАЛЬНІ СВІДОМОСТІ ПРО ПРОЦЕДУРИ ТА ФУНКЦІЇ, БАЗОВІ ПОЛОЖЕННЯ ПРОГРАМУВАННЯ, СКЛАДЕННЯ ПРОГРАМ З ВИКОРИСТАННЯМ ПРОЦЕДУР І ФУНКЦІЙ

Мета роботи

Мета роботи – розглядання та вивчення таких понять мови Паскаль як процедури та функції, їх визначення, описання, застосування і правила користування.

Методичні вказівки

Для поліпшення написання програм, їх структури та виконання була запропонована концепція підпрограм. Підпрограмою зветься програмна одиниця, що має ім'я, звісно якому вона може бути викликана з інших частин програми. У Паскалі це процедури та функції. У загальному випадку підпрограма має ту ж структуру, що і програма.

У Паскалі визначені вбудовані (стандартні) процедури та функції і процедури та функції користувача. Перші, це є частина мови і можуть викликатися за ім'ям без попереднього визначення у розділі описань блоку. Головні стандартні бібліотечні модулі, вбудовані процедури та функції, скалярні процедури та функції, функції перетворювання типів, спеціальні процедури та функції докладно розглянуті у підручниках [1-3]. Далі будуть розглянуті процедури та функції користувача.

Процедура – це незалежна поіменована частина програми, що призначена для виконання заданих дій. Вона містить у себе заголовок та тіло. За своєю структурою можна розглядати як мініатюрну програму. Після описання процедури її можна викликати за ім'ям з послідуєчих частин програми. Коли процедура виконає свою роботу, програма буде продовжена з оператора, що слідує безпосередньо за оператором виклику процедури. Використання імені процедури у програмі зветься оператором процедури або викликом процедури. Ім'я процедури не може знаходитись у вираженні як операнд.

Функція аналогічна процедурі, але вона має дві різниці:

- функція передає до місця виклику скалярне значення (результат своєї роботи);
- ім'я функції може входити до вираження як операнд.

Процедура користувача уявляє собою іменовану групу операторів, що реалізують визначену частину загальної задачі, та при необхідності може бути викликана для виконання за ім'ям з будь-якої позиції розділу операторів. Заголовок містить в себе зарезервоване слово Procedure, ідентифікатор (ім'я) процедури та необов'язковий замкнутій у круглих дужках список формальних параметрів з указуванням типу кожного з параметрів.

Формат процедури:

Procedure <ім'я процедури> {<параметри>;

<розділ описань>

begin

<розділ операторів>

end;

Наприклад:

Procedure DDD (A: real);

Procedure abc;

Для звернення до процедури використовують оператор виклику процедури. Він містить в себе: ідентифікатор (ім'я) процедури та список фактичних параметрів, відокремлених один від одного комами і замкнених у круглі дужки. Список параметрів може бути відсутній, якщо процедурою не передається ніяких значень.

Формат: <ідентифікатор процедури> {(фактичні параметри)};

Наприклад: SSS (a1,a2,a3);

Параметри забезпечують механізм заміни, що дозволяє виконати процедуру з різними початковими даними. Між фактичними параметрами у операторі виклику процедури і формальними у заголовку описання процедури встановлюється взаємно однозначна відповідність у наслідок їх перебору зліва направо. Кількість та тип формальних параметрів повинні відповідати кількості та типу фактичних параметрів. Якщо процедура повертає до програми деякі значення, відповідні змінні повинні бути описані як параметри-змінні з використання слова Var.

Функція, визначена користувачем, складається із заголовка та тіла функції. Заголовок містить у себе зарезервоване слово Function, ідентифікатор (ім'я) функції, замкнуте у круглі дужки, необов'язковий список формальних параметрів і тип функцією значення, що повертається.

Формат функції:

Function <ім'я функції> {<параметри>}: <тип результату>;

<розділ описань>

begin

<розділ операторів>

end;

Наприклад:

Function KKK (x,y,i: integer): real;

Function Ret : Boolean;

У розділу оператори повинно знаходитися не менш одного оператора, що присвоює ідентифікатору функції значення. Якщо таких присвоєвань декілька, тоді результатом виконання є значення останнього оператора присвоєвання.

Звернення до функції здійснюється за ім'ям з необов'язковим списком аргументів. Кожний аргумент повинен відповідати формальним параметрам, вказаним у заголовку, та мати той же тип.

Формат:

<ідентифікатор функції> {(фактичні параметри)};

Параметри можуть мати будь-який тип. Існують три типи параметрів: значення, змінна та не типізована змінна. Група параметрів, поперед якими відсутнє зарезервоване слово Var і за якими слідує тип, мають назву параметрами-значення. Формальний параметр-значення обробляється, як локальна у відношенні до процедури або функції змінна. Зміни у формальних параметрах-значеннях не впливають на значення відповідальних фактичних параметрів. У якості параметрів-значень заборонено файловий тип. Фактичний параметр повинен мати тип, сумісний за присвоєнням з типом формального параметра-значення. Якщо у якості параметра використовується змінна рядкового типу, тоді формальний параметр буде мати довжину 255 байт.

Група параметрів поперед яких слідує ключове слово Var і за якими слідує тип, мають назву параметри-змінні. Наприклад: Procedure FFF(Var X, Y:integer), де X, Y – параметри-змінні. Ці параметри використовуються у тому випадку, якщо значення повинно бути передано з процедури у блок, що викликає. Правила користування формальними та фактичними параметрами теж самі, що і в загалі.

Група параметрів, при описанні яких використовується зарезервоване слово Var, але відсутній тип, мають назву не типізованих змінних. Це означає, що параметр передається у вигляді указника на область пам'яті, яка містить в себе його істинне значення. Використання параметрів доведено у наступних прикладах [1-4].

```
Program DemoParam;
Uses Crt;
Var A, B: byte;
K, N: integer;
I1,I2,J1,J2 : integer;
(*                                     *)
Procedure WW1;
begin
(* Процедура використовує глобальні змінні; після *)
(* виконання процедури їх значення можуть змінитися *)
A:= A + 4;
(* Зміна значення глобальної змінної A *)
end;
(*                                     *)
Procedure WW2 (Var R,T : integer);
(* Процедура передає до основної програми значення R,T;*)
(* якщо не вказати Var, то значення не будуть передаватися *)
begin
R:= 125; T:= 136
```

end;

Procedure WW3 (R1, R2 : integer;

Var X1, X2: integer);

(* R1,R2 - приймає, X1,X2 - передає *)

begin

X1:= R1 + R2;

X2:= R1*R2

end;

Begin

A:= 9;

writeln (A:5);

WW1;

writeln (A:5);

WW2 (K, N);

(* У K та N (відповідально R та T) містяться результати *)

(* роботи процедури WW2 *)

writeln (K:5, N:5);

I1:= 2;

I2:= 5;

WW3 (I1, I2, J1, J2);

(* Значення I1, I2 (відповідально R1, R2) передаються до *)

(* процедури; значення J1,J2 (відповідально X1,X2) приймаються з процедури *)

writeln (J1:5, J2:5);

End.

Для вірного визначення області дії ідентифікаторів при використанні у програмі процедур та функцій необхідно виконувати слідуєчі правила:

- кожний ідентифікатор треба описати поперед тим, як він буде використовуватися;

- область дії ідентифікатора являє собою блок, у якому його описано;
- усі ідентифікатори у блоку повинні не повторюватися;
- один й той ж ідентифікатор може мати різне значення у кожному окремому блоці;
- якщо ідентифікатор підпрограми користувача має теж ім'я, що і стандартна процедура або функція, то остання недоступна в межах області дії підпрограми, декларованої користувачем, інакше стандартна функція ігнорується, а виконується підпрограма користувача.

З рекурсивними підпрограмами, попереднім описанням підпрограм, процедурами керування програмою можна ознайомитись у підручниках [1 -4].

Порядок виконання роботи

1. На алгоритмічній мові Паскаль для вказаного викладачем варіанту завдання з табл.4.1 написати текст програми із застосуванням підпрограм – функції для обчислення функції $z = g(x, y)$.
2. На алгоритмічній мові Паскаль для вказаного викладачем варіанту завдання з табл.4.2 написати текст програми із застосуванням підпрограм – процедури і одержати результати. У завданні a_i і b_i елементи масивів.

Таблиця 4.1 – Дані для виконання завдання 1

Варіант	$f(x, y, z)$	$z = g(x, y)$	a	b
1	2	3	4	5
1	$ x + zy - z^2 / \sqrt{8 + \cos^2 y}$	$2xy / (x + \cos y)$	1,12	0,87
2	$\sqrt{\frac{y}{x^2 + 1}} + e^{-z}$	$\sqrt{x + 2y} \cdot \sin x^2$	6,55	2,78
3	$\frac{x + x^2 + \cos x}{y(y - z) + \ln x}$	$\frac{\sqrt{15y}}{y + \operatorname{ctgx}}$	1,88	3,42
4	$2^{-x} \sqrt{y + \sqrt[4]{z}}$	$\ln(x^4 + y)$	-0,85	1,25
1	2	3	4	5

5	$\frac{1 + \cos^2(x + z)}{ x^3 - 2 \ln \sqrt{y} }$	$\frac{\sqrt{x} \sin^2 y}{x + e^y}$	1,54	3,26
6	$\frac{y + \sqrt{y^2 + x^2}}{(z + x) e^x}$	$\frac{\sin(x / y)}{2x^2}$	-2,77	-1,48
7	$\frac{y \ln x - zx^3}{1 + tg^2(x^2)}$	$\frac{arctgx}{x + y}$	0,47	-0,67
8	$\sqrt{ y } e^{-(x+y)} + \cos(z^3)$	$\frac{x\sqrt{x^3 + y}}{x - 1}$	1,52	0,75
9	$e^{x-1} + \arcsin(y^2 / x)$	$2 \sin(\pi x + y)$	3,24	-1,54
10	$\ln(\sqrt{x} + \sqrt{y}) \cdot z^3$	$\frac{\cos^2 \pi(2 + x)}{4 - y^2 \sqrt{x}}$	3,44	4,82
11	$\cos x + 2tg^2(z / y)$	$\sqrt{x} \sin y$	0,24	-1,78
12	$4y^2 \ln xz $	$\ln(\sqrt[4]{4x^4 + 4y^4})$	3,24	-1,74
13	$\frac{z^3}{x + y^3 / (x + z^2)}$	$\frac{ y + 8x }{\sin x + tgy}$	-0,72	-1,42
14	$e^{-x} (tg^2 z)^y$	$2xy / (x + \cos y)$	-4,24	0,57
15	$\frac{x \ln y - zy^3}{1 + tg^2(y^2)}$	$\frac{x\sqrt{x^2 + y^2}}{x - 1}$	2,67	3,42
16	$\frac{x + \sqrt{y^2 + x^2}}{(z + y) e^x}$	$\frac{\cos(x / y)}{2x^2}$	0,34	0,17
17	$\frac{1 + \sin^2(y + z)}{ y^3 - 2 \ln \sqrt{x} }$	$\frac{\sqrt{y} \sin^2 x}{y + e^x}$	3,26	1,54
18	$\frac{y + y^2 + \cos x}{x(x - z) + \ln y}$	$\frac{\sqrt{15x}}{x + ctgy}$	3,42	1,88
19	$4x^2 \ln yz $	$\ln(\sqrt[4]{4x^4 + 4y^4})$	-1,74	3,24
20	$\frac{\sqrt{x^2 + y^2}}{(z + y) e^y}$	$\frac{\sin(x / y)}{2x^2}$	-2,77	-1,48

Таблиця 4.2 – Дані для виконання завдання 2

Варі- ант	Функція	<i>n</i>	<i>m</i>	Варі - ант	Функція	<i>n</i>	<i>m</i>
--------------	---------	----------	----------	------------------	---------	----------	----------

1	$\sum_{i=1}^n \cos^2(a_i) - \sum_{i=1}^m \cos(2a_i)$	4	8	11	$0.5\sqrt{\sum_{i=1}^n (a_i+1)^2} + 2\sqrt{\sum_{i=1}^m (a_i-1)^2}$	8	5
2	$1.2\sum_{i=1}^n a_i + 5.4\sum_{i=1}^m b_i$	5	8	12	$4.2\sum_{i=1}^n a_i^2 + 1.3\sum_{i=1}^m b_i^2$	4	7
3	$\prod_{i=1}^n a_i / n + \prod_{i=1}^m b_i / m$	3	5	13	$0.011\prod_{i=1}^n a_i + 0.041\prod_{i=1}^m b_i$	7	9
4	$0.21\sum_{i=1}^n \ln a_i + 1.34\sum_{i=1}^m \ln b_i$	4	3	14	$0.42\sum_{i=1}^n \ln a_i + 2.67\sum_{i=1}^m \ln b_i$	5	6
5	$\prod_{i=1}^n a_i / i + \prod_{i=1}^m b_i / i$	6	4	15	$\prod_{i=1}^n a_i^2 / i + \prod_{i=1}^m b_i^2 / i$	3	4
6	$3.8\sum_{i=1}^n a_i^2 + 0.28\sum_{i=1}^m (a_i^2 - 1)$	3	5	16	$0.8\sum_{i=1}^n (a_i^2 + 2) + 0.8\sum_{i=1}^m (a_i^2 - 2)$	4	3
7	$\sqrt{\sum_{i=1}^n (a_i+1)^2} + \sqrt{\sum_{i=1}^m (b_i-1)^2}$	4	6	17	$\sum_{i=1}^n \sin^2(2a_i) - \sum_{i=1}^m \sin(2b_i)$	5	7
8	$0.1\prod_{i=1}^n (a_i-1) + 1.41\prod_{i=1}^m (b_i-1)$	8	3	18	$\prod_{i=1}^n \sqrt{a_i / i} + \prod_{i=1}^m \sqrt{b_i / i}$	7	3
9	$1.2\sum_{i=1}^n (a_i+1) + \sum_{i=1}^m (b_i-1)$	6	3	19	$\sqrt{\sum_{i=1}^n (a_i-1)^2} + 2\sqrt{\sum_{i=1}^m a_i^3}$	4	5
10	$1.2\sum_{i=1}^n (a_i^2+1) + \sum_{i=1}^m (a_i^2-2)$	4	7	20	$\prod_{i=1}^n a_i^2 / (i+1) + \prod_{i=1}^m b_i^2 / i$	5	3

Приклад виконання завдання 1. Написати текст програми для обчислення функції

$$F(x) = 0.5 \sum_{i=1}^8 3x_i^2 / i + 1.4 \sum_{i=1}^5 x_i^2 / i - \sum_{i=1}^6 4x_i^2 / i,$$

де обчислення суми зробити у вигляді підпрограми-функції.

```

Program fx;
Uses Crt;
Var
x, f : real;
Function S (x, m: real, n: integer) : real;
Var
i: integer;
Begin
S:= 0;
```

```

for i:= 1 to n do
S:= S + sqr (x)/i*m;
End;
Begin
readln (x);
writeln ('x=', x);
f:= 0.5*S(x,8,3) + 1.4*S(x,5,1) - S(x,6,4);
writeln ('f=', f)
End.

```

Приклад виконання завдання 2. Написати програму для обчислення функції

$$F(x) = \frac{\prod_{i=1}^2 a_i}{\sqrt{\sum_{i=1}^2 a_i}} x^2 - \frac{\prod_{i=1}^4 a_i}{\sum_{i=1}^2 a_i} x + \frac{\prod_{i=1}^n a_i}{\ln\left(\sum_{i=1}^n a_i\right)}$$

де a - елемент масиву A , що складається з 16 елементів ($n = 16$). Обчислення суми та добутку зробити у вигляді підпрограми-процедури.

```

program ms;
Uses Crt;
type
mas = array[1..20] of real;
var
a:mas;
i,n:integer;
x, s1, s2, s3, p1, p2, p3,g:real;
procedure sumpr (a: mas; n:integer; var s, p: real);
var
i: integer;
begin

```

```

s :=0;
p:=1;
for i:=1 to n do begin
s := s + a[i];
p := p*a[i];
end;
end;
begin
readln (input, n, x);
for i:=1 to n do readln (input, a[i]);
sumpr (a,2,s1,p1);
sumpr (a,4,s2,p2);
sumpr (a,n,s3,p3);
f:=p1*x*x/sqrt (s1) - p2*x/s2+p3/ln(s3);
writeln (output, 'g=',g)
end.

```

Контрольні запитання

1. Що таке локальні та глобальні змінні? Які параметри Ви знаєте?
2. Чим відрізняється процедура-функція від підпрограми-процедури?
3. Яким чином здійснюється виклик функції та процедури?
4. Що таке рекурсивні підпрограми?
5. Назвіть правила визначення області дії ідентифікаторів.
6. Що таке формальні і фактичні параметри? Яка їх кількість та який тип вони мають?

Перелік джерел інформації

1. Бородич Ю. С. Паскаль для персональных компьютеров / Ю.С. Бородич, А.Н. Вальвачев, А.И. Кузьмич. – Мінськ.: БФ ГИТМП «НИКА», 1991. – 212 с.
2. Путятин Е.П. Турбо Паскаль в курсе высшей математики / Е.П. Путятин, Д.М. Смагин, В.П. Степанов – Харьков: Каравелла, 1997.– 352 с.
3. Алгоритмічна мова Паскаль / В.І.Кравченко, О.В. Бережной, В.В. Зоненко, 2007.– 119 с.
4. Основы програмування в середовищі Турбо Паскаль: Лекції та завдання до лабораторних робіт / І.І. Демків, П.І. Катенюк, І.В. Ключник.– Львів, 2003.– 202 с.

Навчальне видання

**Методичні вказівки
до виконання курсової роботи з дисципліни
«Інформатика»**

для студентів спеціальності 6.050202
«Автоматизація управління технологічними процесами»
денної та заочної форми навчання

Укладач ХАВІН Геннадій Львович

Відповідальний за випуск *Ю.В. Тимофієв*

Роботу до видання рекомендував *О.М. Шелковий*

Редактор *Л.А. Пустовойтова*

План 2013 р., поз. 132

Підписано до друку _____ Формат 60×84 1/16 Папір офсет. Riso-друк.

Гарнітура Times New Roman. Ум. друк, арк. . Наклад 100 прим. Зам №_____

Ціна договірна.

Видавничий центр НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.

Друкарня НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21