
Практична робота № 18 «Побудова алгоритмів з використанням процедур і функцій».

Мета.

Навчальна. вчитися будувати алгоритми з використанням процедур і функцій та реалізовувати їх у вигляді програм.

Розвиваюча. Розвивати логічне мислення та алгоритмічне мислення, навички створення програм, самостійність, вміння аналізувати та застосовувати набуті знання до практичних завдань.

Виховна. Виховувати наполегливість, естетичність у оформленні, зібраність, уважність, вміння раціонально використовувати час.

Тип уроку. Засвоєння нових вмінь і навичок.

Матеріали для роботи з учнями:

[Підпрограми](#)

План

Актуалізація опорних знань.

ТБ.

Практична робота.

Завдання для практичної роботи.

Підсумки уроку.

Домашнє завдання.

Пам'ятка для учня!

Пригадайте правила техніки безпеки при роботі з ПК.

Через кожні 15 хв. виконуйте вправи для очей та для зняття м'язової втоми.

Хід уроку

1. Організаційний момент.

2. Перевірка домашнього завдання.

Наявність.

Питання.

3. Актуалізація опорних знань.

У мові Паскаль є два види підпрограм: підпрограми-процедури і підпрограми-функції. Спочатку розглянемо процедури.

Концепція структурного програмування полягає в тому, що будь-яку складну чи громіздку задачу потрібно звести до послідовності простіших підзадач, записати підпрограми для їхнього розв'язування і створити головну програму, яка має складатися з викликів цих підпрограм.

Різновидом підпрограми є процедура, де розв'язується допоміжна задача. Процедура нагадує звичайну програму, але починається не зі слова `program`, а зі слова `procedure` і закінчується не крапкою, а крапкою з комою:

```
procedure <Назва процедури>;  
  <Розділи описів і оголошень процедури> {Оголошені тут  
  змінні називаються локальними}  
  begin  
    <Розділ команд процедури>  
  end;
```

Назву процедури дає користувач. Процедур у програмі може бути декілька. Їх записують у розділі описів головної програми. Головна програма має такий вигляд:

```
program <Назва головної програми>;  
  <Розділ описів і оголошень>;  
    {Оголошені тут змінні називаються глобальними}  
  <Процедура 1>;  
  <Процедура 2>;  
  begin  
    <Розділ команд, який містить:  
      виклик процедури 1;  
      виклик процедури 2;>  
  end.
```

Розділ команд головної програми складається головно з викликів процедур. Виклик процедури — це команда, яка збігається з назвою процедури.

Локальна змінна діє лише в процедурі, в якій вона описана. Змінні, описані в головній програмі, називаються **глобальними**. Вони діють у всіх процедурах.

Через глобальні змінні процедура отримує вхідні дані з головної програми, а також повертає їй результати обчислень.

Приклад.1

Дано сторони a і b прямокутника. Обчислити його периметр і площу, використавши для цього процедуру,

```
program Main2;  
  var a, b, p, s : real;  
  procedure Rectangular(a, b : real; var p, s : real);  
    begin  
      p:= 2*(a + b);  
      s:= a * b  
    end;  
begin  
  write ('Введіть a і b ');  
  readln(a,b);  
  Rectangular;  
  writeln(p:6:2, s:6:2)  
end.
```

Тут a , b , p , s — глобальні змінні. Починає виконуватися розділ команд головної програми, а не процедура, хоча вона записана вище. Користувач вводить дані з клавіатури, натискає на клавішу вводу, і лише тоді виконується команда *Rectangular* — керування передається процедурі, де обчислюються периметр і площа. Коли процедура виконана, керування передається наступній команді головної програми *writeln*, яка виводить результати на екран.

Формальні та фактичні параметри.

У випадку складних задач, які потребують використання багатьох процедур і залучення до їх складання декількох виконавців, спосіб обміну даними між розділом команд і процедурою через глобальні параметри є неефективним.

Універсальним способом обміну даними; між процедурами і головною програмою є використання списків формальних і фактичних параметрів. Список формальних параметрів записують у круглих дужках після назви процедури:

procedure <Назва> (Список формальних параметрів>)

У списку формальних параметрів змінні оголошують із зазначенням типу, а перед змінними кожного типу, які повертатимуть результати обчислень у головну програму, має бути слово `var`. Наприклад, розглянемо задачу про обчислення периметра і площі прямокутника і розв'яжемо її за допомогою процедури з параметрами:

```
procedure Rectangular(a, b : real; var p, s : real );  
begin  
    p := 2 * (a + b);  
    s := a * b  
end;
```

Команда виклику процедури має такий загальний вигляд:

<Назва> (Список фактичних параметрів>)

На місці фактичних параметрів, які передають дані в процедуру, можуть бути сталі, змінні або вирази. На місці фактичних параметрів, які отримують з процедури результати, можуть бути лише змінні.

Між формальними і фактичними параметрами має бути така відповідність: кількість параметрів у списках, їхні типи і послідовність мають бути однаковими. Назви відповідних формальних і фактичних параметрів можуть не збігатися.

Наприклад, процедуру `Rectangular` можна викликати командою

```
Rectangular(3, 4, p, s);
```

Отримаємо такий результат:

```
p = 14.0, s = 12.0.
```

Дія команди виклику процедури. Активізується процедура із заданою назвою (`Rectangular`). Значення фактичних параметрів (вхідні дані 3 і 4) надаються відповідним формальним параметрам (`a` і `b`) процедури. Виконується процедура, де обчислюються периметр (`p`) і площа (`s`). Результати виконання процедури присвоюються фактичним параметрам (`p` і `s`) у команді виклику. Після цього виконуватиметься наступна команда, що йде за командою виклику.

Підпрограма-функція.

Якщо серед стандартних функцій (таких як `sin(x)`, `cos(x)`, `ln(x)`, `exp(x)`, `abs(x)` тощо) немає потрібної, то користувач може сконструювати нестандартну функцію за допомогою підпрограми-функції, яку належить описати у розділі описів головної програми (там, де і процедури) так:

function <Назва>(<список формальних параметрів>) : <тип функції>;

<Розділ описів функції>

begin

<Розділ команд, серед яких є така: Назва := вираз>

end

У функції параметри такі ж, як у процедури: параметри- аргументи (параметри-значення) і параметри-результати (параметри-змінні). Перед останніми пишуть службове слово var.

Особливість функцій Назва функції є іменем змінної. Назву функції надає користувач. Тип функції описують у заголовку.

Підпрограма-функція, як і всі стандартні функції (sin(x), cos(x) тощо), повертає в точку її виклику головний результат

— значення функції. Цей результат присвоюється назві функції.

Зверніть увагу, як утворюється результат: тіло функції має містити таку команду присвоєння:

<назва функції> := <вираз для отримання результату>

Обчислення функцій Звертання до функції (виклик функції) виконують за допомогою вказівника функції, а саме:

<Назва>(<список фактичних параметрів>)

Вказівник функції, наприклад Gryvni(25), можна використовувати лише у виразах так, як sin(x), cos(2*pi) тощо.

Дія вказівника. Серед описів відшуковується функція з відповідною назвою, наприклад, Gryvni. Значення фактичних параметрів (сталих, змінних, виразів) вказівника передаються формальним параметрам функції. Виконується обчислення у тілі функції. Головний результат через назву функції повертається у головну програму (кажуть: присвоюється вказівнику). Інші результати, якщо вони є, присвоюються параметрам-результатам.

Приклад.2.

Скільки гривень можна отримати в банку в обмін на 25 доларів, якщо курс становить 25,05 грн/дол.?

Побудуємо функцію, яка переводить долари в гривні:

function Gryvni(Dollars : integer) : real;

begin

*Gryvni := 25.05 * Dollars*
end;

Тепер у головній програмі у будь-якому виразі можна записати звертання до цієї функції — Gryvni(25), наприклад, це зробимо у команді виведення:

writeln('Сума в гривнях така:', Gryvni(25)).

Після виконання відповідної програми на екрані отримаємо:

Сума в гривнях така: 626.25

4. Техніка безпеки під час роботи на ПК.

[Інструктаж з ТБ при роботі з ПК та в комп'ютерному класі.](#)

5. Практична робота № 18 «Побудова алгоритмів з використанням процедур і функцій».

Хід роботи

- Вивчити можливості мови програмування Pascal використання функцій та процедур опрацювання рядкових величин:
- склеювання рядкових величин;
- визначення довжини рядкової величини;
- копіювання;
- видалення та вставка.
- Вибрати варіант завдання (за погодженням з учителем).
- Скласти математичну модель, алгоритм розв'язування задачі та реалізовувати їх у вигляді програм.
- Завантажити середовище програмування Паскаль.
- Ввести з клавіатури текст програми, складеної на основі блок-схеми.
- Програма має обов'язково містити команди:
- введення даних, які є відомими в задачі;
- виведення на екран даних, які є результатом розв'язку задачі.
- Зберегти у файлі текст введеної програми.
- Назва файлу має нести інформацію про номер практичної роботи та прізвище її виконавця. Файли програм мають зберігатися в особистій папці на диску.
- Виконати запуск створеної програми на виконання.

- у випадку появи повідомлень про помилки уважно перевірити текст вашої програми і виправити помилки. Якщо помилку не вдається виправити самотійно, потрібно скористатися допомогою вчителя;
- зробити «зрозумілим» інтерфейс програми;
- доповнити програму командами виведення з текстовими підказками. Між величинами, які виводяться на екран, має бути просторове розділення. Вибрати найзручніший формат для даних, що виводяться.
- Одержати результат виконання доповненої програми. Переписати у звіт текст робочої програми.
- Повторити запуск програми кілька разів, міняючи кожен раз вхідні дані. Переписати у звіт результати виконання програми.
- Зберегти у файлі остаточний текст програми. Ім'я файла та адресу його розміщення записати у звіт.
- Зробити висновки щодо виконаної роботи.

6. Завдання до практичної роботи.

Рівень А

Дано дійсні числа a, b, c, d . Обчислити:

$$y = (p(a) + p(b) + p(c) + p(d))/4,$$

де $p(x) = 4x^4 + 3x^3 + 2x^2 + x + 0.5$.

Рівень Б

I. За даними дійсними числами a, b обчислити:

$$u = f(0.5, a) + f(a + b, a - b),$$

$$\text{де } f(x, y) = \frac{x^2 + xy - y^2}{1 + x + y} + \frac{x - y}{x^2 + y^2 + 2}.$$

II. Дано дійсні числа u та v . Визначити значення:

$$z = f(u + v, v - u) + f(1.7, v),$$

$$\text{де } f(x, y) = \frac{x + y}{x^2 + xy + y^2} + \frac{x}{1 + y^2}.$$

Рівень В

I. Дано дійсні числа x, y та z . Обчислити:

$$u = \frac{\max(x, y) + \max(x + x, xz)}{\max(0.5x + z)}$$

II. Дано дійсні числа a, b, c . Обчислити:

$$z = \frac{\max(a, a + b) + \max(a, b + c)}{1 + \max(a + bc, b - c)}$$

7. Підсумки уроку.

1. Для чого створюють підпрограми?
2. Що таке процедура і для чого її створюють?
3. У чому суть концепції структурного програмування?
4. Який вигляд має процедура?
5. Який загальний вигляд має головна програма?
6. Який вигляд має команда виклику процедури?
7. Хто дає назву процедурі?
8. Де записують процедури?
9. Де описуються і діють глобальні змінні?
10. Де описуються і діють локальні змінні?
11. Яке призначення глобальних і локальних змінних?
12. Що таке фактичні й формальні параметри?
13. Як відбувається обмін даними між фактичними і формальними параметрами?
14. Чи повинні збігатися назви відповідних фактичних і формальних параметрів?
15. У якому випадку фактичний параметр може бути сталою або виразом?
16. Чи може формальний параметр бути сталою або виразом?
17. Який вигляд має команда виклику процедури?
18. Скільки результатів може повертати процедура?
19. Яке призначення функцій?
20. Яка різниця між стандартними і нестандартними функціями?
21. Хто складає і надає назви нестандартним функціям?
22. Скільки значень повертає підпрограма-функція?
23. Які є правила оформлення функції?
24. Що таке вказівник функції?
25. Як транслятор обчислює нестандартну функцію?
26. Чи може бути функція без параметрів? (відповідь: так, дані передаються у функцію через глобальні змінні).

8. Вправи та задачі.

1. Що виконує така процедура:

```
procedure Mytext;  
begin  
    writeln('З Новим роком!')  
end;
```

2. Запишіть результат виконання програми

```
program New; var i : integer;  
procedure One;  
begin  
    writeln('111111')  
end;  
procedure Two;  
begin  
    writelnf 222222')  
end;  
begin  
    One;  
    Two;  
    One;  
    for i:=1 to 3 do Two  
end.
```

3. Нарисуйте, що виводить на екран така процедура:

```
procedure T;  
var i : integer;  
begin  
    for i := 1 to 5 do writeln('—':i)  
end;
```

4. Що виконує така програма:

```
program Rysunok;  
procedure Stars(n: integer);  
    var i:integer;  
begin  
        for i := 1 to n do write('*'); writeln  
end;  
begin  
    Stars(5);  
    Stars(10);  
    Stars(15)  
end.
```

5. Що робить така програма:

```

program Sumy; var k,s : integer;
  procedure Suma(n: integer; var s: integer);
    var i : integer;
  begin
    s := 0;
    for i:=1 to n do s := s+ i
  end;
begin
  suma(5,s);
  writeln(s);
  read(k);
  suma(k,s);
  writeln(s);
end.

```

6. В одній процедурі передбачте виведення на екран свого прізвища та імені, в інших двох — виведення на екран даних про двох ваших друзів. Дайте процедурам назви і складіть головну програму, в якій першу процедуру викличіть один раз, другу — п'ять разів, а третю — сім.
7. Обчисліть довжини п'яти кіл і площі відповідних п'яти кругів, якщо їхні радіуси набувають значень 2, 4, 6, 8, 10. Використайте: а) процедуру для обчислення довжини кола і процедуру для обчислення площі круга; б) одну процедуру.
8. Обчисліть площі поверхонь і об'єми трьох повітряних куль ($S = 4\pi r^2$, $V = (4/3)\pi r^3$) за відомими радіусами r , використавши процедуру для обчислення площі й об'єму. Радіуси куль відповідно дорівнюють 5, 10, 15 м.
9. Оформіть алгоритм розв'язування квадратного рівняння у вигляді процедури. У головній програмі передбачте введення коефіцієнтів і двічі викличіть процедуру для розв'язування квадратного рівняння
10. Складіть процедуру з параметрами, яка отримує три цілі числа, а повертає їхню суму, добуток і середнє арифметичне. Складіть відповідну головну програму.
11. Складіть програму з використанням процедури з параметрами для обчислення периметра і площі двох прямокутників.
12. Складіть процедуру, яка отримує значення аргумента, а повертає значення трьох функцій: $y = \sin x$, $2 - \cos x$, $v = \tan x$.
13. Складіть програму, де вводиться ціле число і викликаються три процедури обчислення сум чисел від 1 до заданого: а) всіх чисел; б) лише парних; в) лише непарних. Виведіть результати.

14. Складіть процедуру для визначення мір, якщо 1 дюйм = 2,54 см = 1/12 фута, а також головну програму для побудови 10 рядків таблиці цих мір для перших десяти дюймів. Додаткове завдання: запишіть модуль з такою процедурою.
15. Складіть програму розв'язування квадратного рівняння, використовуючи процедуру з параметрами.
16. Складіть функцію для обчислення площі круга за заданим радіусом.
17. Складіть функцію і запишіть програму для переведення гривень у долари за курсом НБУ.
18. Складіть програму для побудови таблиці мір, використовуючи функцію, якщо: а) 1 морська миля = 1852 м; б) 1 кабельт = 183 м; в) 1 ярд = 0,9144 м; г) 1 фут = 0,3048 м.
19. Складіть програму табулювання функції $y = \sin^2 x$ на деякому проміжку з деяким кроком, використайте підпрограму-функцію для її обчислення.
20. Складіть програму для обчислення медіан у трикутнику, заданому координатами вершин.
21. Складіть програму, щоб визначити, скількома способами можна призначити будь-яких трьох учнів у вашому класі черговими (застосуйте формулу для C_n^m зі значеннями $m=3$, n — кількість учнів у класі, використайте функцію для обчислення факторіалів).
22. Складіть програму, щоб визначити, скількома способами можна обрати п'ятьох учнів у вашому класі для участі в змаганнях.

9. Домашнє завдання.

Повторити "Підпрограми"

Виконати завдання 13, 20.