



ТЕМА УРОКУ

**Поняття про підпрограми.
Складання та реалізація
програм, що містять
звернення до підпрограм.**





План уроку

- 1. Поняття про підпрограми. Види підпрограм.**
- 2. Структура та описання підпрограм-процедур.**
- 3. Формальні та фактичні параметри. Приклади підпрограм-процедур.**
- 4. Локальні та глобальні змінні. Процедури функції.**
- 5. Виконання тестів і складання програм.**
- 6. Тестова перевірка ТЕМА 3_5 – оцінювання.**
- 7. Підсумки уроку та домашнє завдання.**



Мета уроку:

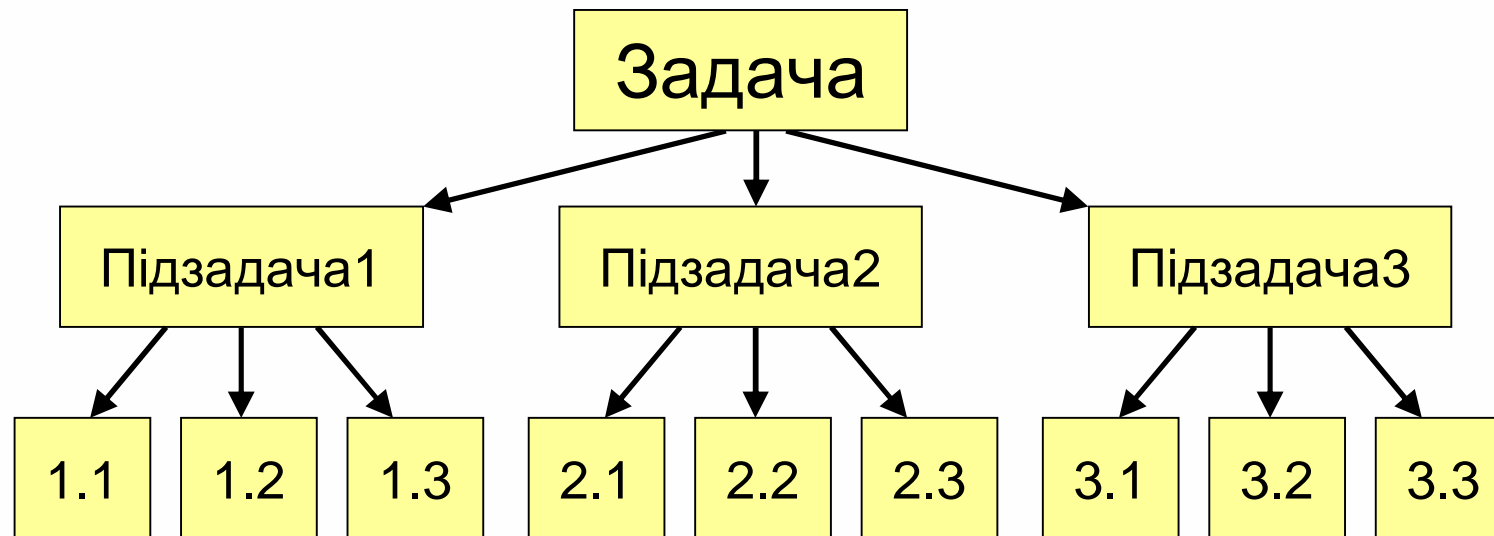
1. Оволодіти знаннями, вміннями, навичками в створенні та реалізації програм із зверненням до підпрограми;
2. Розвивати логічне мислення, вміння та навички в програмуванні;
3. Формувати культуру естетичного оформлення програми на мові програмування Pascal.



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Поняття про підпрограми.

При розв'язанні складної задачі рекомендується складати алгоритм, а потім і програму від більш загального плану до докладного.



Використання підпрограм має такі переваги:

- полегшення розробки великих і складних програм;
- економія пам'яті: пам'ять для змінних виділяється тільки на час роботи підпрограми.



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Поняття про підпрограми.

Підпрограма – це іменована, логічно закінчена група операторів, яку можна викликати для виконання будь-яку кількість разів із різних місць програми.

Підпрограми мають структуру, аналогічну до структури головної програми на Паскалі.

Кожна підпрограма – це міні-програма в програмі. В ній діють всі закони, що притаманні будь-якій програмі.



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Поняття про підпрограми.

У Паскалі розрізняють два види підпрограм: **функції та процедури.**

Процедура - це незалежна іменована частина програми, яку можна викликати з будь-якого місця програми для виконання певних дій.

Щоб до процедури можна було звернутися з основної програми, її слід описати.

Функції – це підпрограма особливого характеру, результатом роботи яких є певне значення подібне до змінної значення.

Відмінність від процедур полягає в тому, що результатом виконання функції є єдине значення! Це обчислюване значення надається ідентифікатор функції. Структура опису функції аналогічна опису процедурию



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Місце описання підпрограм

Всі підпрограми-процедури описуються перед командами основної програми.

```
Program ddd;  
Uses Crt;  
Var опис типів величин;
```

Заголовок основної програми

```
Підпрограма  
var опис локальних змінних;  
Begin  
    Команди підпрограми  
end;
```

Описання підпрограми

```
Begin  
    Команди програми  
end;
```

Тіло основної програми



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Структура процедури

Структура тіла процедури аналогічна структурі програми, тобто містить опис міток, констант, типів, змінних, а також розділ операторів:

```
Procedure < ім'я >;  
Const...;  
Var ...;  
Begin  
<оператор>;  
End;
```

Розглянемо як приклад
процедуру обчислення
дискримінанта
квадратного рівняння:

```
Procedure Discr ( A,B,C,D : real);  
Begin  
D:=B*B-4*A*C;  
End;
```



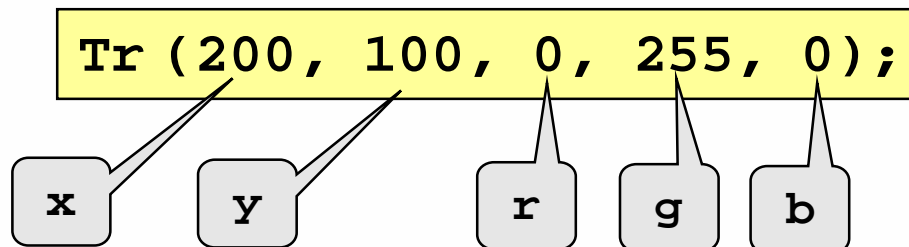

Процедури

Особливості:

- всі процедури розміщені **вище** основної програми
- в заголовку процедури перераховуються **формальні** параметри, вони позначаються іменами, оскільки можуть змінюватися

```
procedure Tr( x, y, r, g, b: integer);
```

- при виклику процедури в дужках вказуються **фактичні** параметри (числа або арифметичні вирази) **в тому ж порядку**





Процедури

Особливості:

- для кожного формального параметра після двокрапки вказується його тип

```
procedure A (x: real; y: integer; z: real);
```

- якщо однотипні параметри стоять поряд, їх перераховують через кому

```
procedure A (x, z: real; y, k, l: integer);
```

- всередині процедури параметри використовуються так само, як і змінні



Процедури

Особливості:

1. В якості фактичних параметрів (аргументів) можуть використовуватися не лише імена змінних або табличних величин, а й константи, арифметичні вирази.
2. Типи відповідних параметрів команди виклику і заголовка процедури мають збігатися.

Команда виклику процедури виконується в три етапи:

- 1) обчислення фактичних аргументів;
- 2) виконання алгоритму-процедури;
- 3) надання отриманих значень результатів алгоритму-процедури відповідним фактичним змінним.



ВИКОНАЙМО РАЗОМ!

Приклади процедур без параметрів

Приклад1: оформити у вигляді процедури програму привітання.

```
program HelloAll;
  procedure Hello(Name:string);
  begin
    Writeln('Привіт,',Name,'!');
    Writeln (Name,', як справи?');
    Writeln;
  end;
begin
  Hello('Катя');
  Hello('Андрій');
  Hello('Олена');
end.
```

Приклад2. Вивести на екран декілька рядків, які складатимуться з рисок і зірочок

```
program Main1;
  procedure Star; { це процедура Star }
  begin
    writeln ( ' ***** ' );
  end;      { кінець процедури Star }
  procedure Dash; {це процедура Dash }
  var i : integer ;
  begin
    for i := 1 to 20 do write ( ' - ');
    writeln
  end;      { кінець процедури Dash}
begin
  Dash;      { це виклик процедур }
  Star;
  Dash;
  Star
end.
```



ВИКОНАЙМО РАЗОМ!

Приклади процедур без параметрів

Приклад3: Дано сторони a, b прямокутника. Обчислити його периметр і площу, використавши для цього процедуру без параметрів.

```
program Main2;
var a,b,p,s:real;
  procedure Rectangular; { керування передається процедурі, де
                           обчислюється периметр і площа}
  begin
    p:=2*(a+b);
    s:=a*b;
  end;
begin
  writeln ('vvedite a,b');
  readln(a,b);
  rectangular;
  writeln (p,s);
end.
```



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Процедури з параметрами

Після заголовку процедури в круглих дужках можуть **вказуватись змінні** (з допомогою яких в процедуру передаються дані) і їх типи, які називаються *параметрами-значеннями*.

Змінні, які описанні в заголовку процедури є *формальними параметрами*.

Змінні або константи, які описанні у вказівці процедури є *фактичними параметрами*.

```
Program Perrymetr;  
Var C,D:integer;  
procedure PARAM(A,B:integer);  
Var S:integer;  
Begin  
S:=0;  
S:=A+B;  
Writeln ('s=',S);  
end;  
Begin {Основна програма}  
C:=10; {1-й спосіб}  
D:=100;  
PARAM(C,D);  
PARAM(100,100); {2-й спосіб}  
End.
```



Для того, щоб процедура змогла змінювати значення фактичних параметрів, потрібно змінити спосіб передачі параметрів в процедуру. Цей спосіб називається передачею по імені (змінні називаються параметрами-змінними).

Задача. Дано дві трійки чисел: A1, B1, C1, A2, B2, C2. Знайти значення сум:

$S1 = \min(A1, B1, C1) + \min(A2, B2, C2)$

$S2 = \max(A1, B1, C1) + \max(A2, B2, C2)$

Program Priklad;

Var A1, B1, C1, A2, B2, C2, MIN1, MAX1, MIN2, MAX2, S1, S2: real;

Procedure MinMax(A, B, C: real; var MIN, MAX: real);

Begin

MAX := A;

If MAX < B then MAX := B;

If MAX < C then MAX := C; end;

Begin

write('A1='); Readln (A1); write('B1='); Readln (B1); write('C1='); Readln (C1);

write('A2='); Readln (A2); write('B2='); Readln (B2); write('C2='); Readln (C2);

MinMax(A1, B1, C1, MIN1, MAX1);

MinMax(A2, B2, C2, MIN2, MAX2);

S1 := MIN1 + MIN2;

S2 := MAX1 + MAX2;

Writeln ('S1='); writeln ('S2=');

End.



ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Підпрограма-функція.

Підпрограма, яка має **єдиний результат**, може бути оформлена як функція. Опис підпрограми-функції має такий вигляд.

Перегляд

Функція викликається за допомогою вказівника. Вказівник – це ім'я функції, після якого в дужках перелічені фактичні параметри – аргументи функції. Вказівник має вигляд.

Перегляд





ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Function < Ім'я функції > (< входні дані >):< тип результату >;

{опис локальних змінних }

Begin

{Оператори}

Ім'я функції := результат;

End;





ЗРОЗУМІЙ, ЦЕ ПРОСТО!

Підпрограма-функція містить команду присвоєння:

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

Звертання до функції виконують за допомогою вказівника функції:

<Назва> (<Список фактичних параметрів>;



ПРИКЛАД ОПИСУ ФУНКЦІЇ

```
function Max2 ( x, y : real ) : real ;
```

ІМ'Я

ПАРАМЕТРИ

ТИП ФУНКЦІЇ

[блок опису локальних змінних]

```
begin
```

```
  if x >= y then
```

```
    Max2 := x
```

```
  else Max2 := y
```

```
end;
```

В описі функції є заголовок, блок опису локальних змінних та тіло функції.

В заголовку після *імені* в дужках описано *параметри*, а за дужками описано тип результату. *Ім'я результату співпадає з іменем функції*, тому в тілі функції воно (виділено синім кольором) використовується в якості результату.




Виклик функції

```
program Max3;  
var a, b, c, max : real;  
function Max2( x, y : real) : real;  
begin  
if x >= y then Max2:=x else Max2 := y  
end;  
begin  
Writeln('ввести a,b,c:');  
Read(a); Read(b); Read(c);  
max := Max2 (a, b) ;  
max := Max2 (max, c);  
Writeln(max :6:2);  
end.
```

Функція – це підпрограма, допоміжний алгоритм.

Має декілька (або один) формальних параметрів.

Суттєвою відмінністю її від процедури є те, що функція має лише один результат – ім'я функції.

 Параметри, описані в заголовку – формальні. У команді виклику функції їм відповідають фактичні параметри. Аргументи виділено синім, а результати червоним кольорами. Інший варіант команди виклику:

```
max := Max2 ( Max2 ( a, b ), c )
```



ВИКОНАЙ! В ТЕБЕ ВИЙДЕ!

1. Виберіть правильну відповідь. Функція мовою Pascal описується так:
- a) Function назва(список формальних параметрів) = вираз;
 - b) Function назва(список фактичних параметрів) = вираз;
 - c) Function назва(список формальних параметрів) : тип.

2. Функція викликається так:

Y:= назва;

Z:= назва(список фактичних параметрів);

Z:= назва(список формальних параметрів).

3. Повернення з процедури відбувається на...

Останній оператор програми;

На початок програми;

Оператор після виклику процедури.

4. Формальні і фактичні параметри мають співпадати...

За типами;

За розміром;

За назвою.



ВИКОНАЙ! В ТЕБЕ ВИЙДЕ!

5. Процедура в основній програмі викликається так:

- a) Назва(список формальних параметрів);
- b) Назва(список глобальних параметрів);
- c) Назва(список фактичних параметрів).

6. Функція має таку кількість результатів:

- a) Два;
- b) Багато;
- c) Один.

7. Обов'язковим оператором функції є...

- a) $y :=$ назва функції;
- b) Назва(список фактичних параметрів);
- c) Назва_функції: = вираз.

8. Виберіть правильну відповідь. Процедура з параметрами мовою Pascal описується так:

- a) Procedure назва(список формальних параметрів) : тип;
- b) Procedure назва(список фактичних параметрів) = вираз;
- c) Procedure назва(список формальних параметрів).



Виконання вправ

Картки №1, 5, 7



Домашнє завдання

1. Опрацювати конспект
2. Виконати картки №2, 4, 6

