

A spiral-bound notebook with a light-colored, textured cover and a silver metal spiral binding on the left side. The notebook is open to a page with a faint grid pattern. The text is centered on the page.

# Структура програми у ВР 7.0

Синтаксично програма складається з необов'язкового *заголовка* і *програмного блоку*, який у свою чергу будується з двох частин:

- описової;
- виконавчої.

**ЗАГОЛОВОК**

*Program* <ім'я>;

## ОПИСОВИЙ РОЗДІЛ

*Uses <ім'я1, ім'я2>; {список використаних бібліотечних модулів}*

**Приклад: USES CRT, GRAPH;**

*Label <ім'я1, ім'я2>; {розділ опису міток}*

**Приклад: Label 5, A20, 172;**

**GOTO A20;**

*Const < ідентифікатор > = < значення >;  
{розділ оголошення констант}*

**Приклад:** Const digit=1000;  
e=2.71828;  
pidpys='Kravchenko';

*Type < ідентифікатор > = < опис типу >;  
{розділ опису типів даних заданих  
програмістом}*

**Приклад:** Vector=Array[1..10] of real;  
Color=(red,green,blue);

*Var*

*< ідентифікатор > : < тип >;  
{ розділ опису змінних }*

**Приклад:** *Var i,j: integer;  
s, q: real;  
key: char;*

**! Потрібно перерахувати імена усіх змінних, які використ. у програмі;**

*{ розділ опису підпрограм }*

**Procedure**

**function**

Підпрограма

var опис локальних змінних;

Begin

Команди підпрограми

end;

## РОЗДІЛ ОСНОВНОЇ ПРОГРАМИ

*Begin*

*< тіло основної програми >;*

*End.*

# Оператори ВВОДУ-ВИВODУ



**Введення даних** - це передача інформації від зовнішнього носія в оперативну пам'ять для обробки.

**Виведення** - це зворотній процес, коли дані передаються після обробки з оперативної пам'яті на зовнішній носій.

У мові Паскаль стандартним засобом спілкування людини з ЕОМ є консоль, що складається з таких пристроїв, як клавіатура та екран монітора.

## Формат введення:

***Read*** ( $x_1, x_2, \dots, x_n$ );

де  $x_1, x_2, \dots, x_n$  - змінні допустимих типів даних,

***Readln*** ( $x_1, x_2, \dots, x_n$ );

дані зчитуються рядками,

Значення введених даних повинні строго відповідати типам цих даних вказаних у розділі VAR, інакше компілятор виводить на екран повідомлення про помилку.

***Readln;*** чекає натиснення *Enter*,  
зручно писати перед *End*.

## Формат виведення:

**Write** (Y1, Y2,... Yn) - виведення на монітор;  
де Y1, Y2,... Yn - дані, що виводяться,

**Writeln** (Y1, Y2,... Yn)  
дані виводяться рядками,

**Приклад:**

```
Writeln('Y1=', Y1)
```

# Виведення значення цілої величини I

<i>Значення I</i>	<i>Вираз</i>	<i>Результат</i>
134	write(I:6);	____134
1	write(I:10);	_____1
312	write(I+I:7);	_____624

# Виведення значення дійсної величини **R**

<b>Значення R</b>	<i>Вираз</i>	<i>Результат</i>
511.04	write(R:8:4);	511.0400
-46.78	write(R:7:2);	_ -46.78
-46.78	write(R:9:4);	_ -46.7800

# **Керування порядком обчислень**

Кожна програма складається  
з певного набору конструкцій

---

- проходження
- розгалуження
- цикл

# ЛІНІЙНА СТРУКТУРА

Містить оператори трьох типів:

- GOTO <мітка>;
- оператор виклику процедур;
- оператор присвоєння :=



## оператор присвоєння :=

<ім'я змінної>:=<вираз>;

Тип змінної і тип виразу повинні збігатись  
(ціле автоматично перетворюється у дійсне)

### Заокруглення

a2:=**round**(b-c+sin(x));

# ЛОГІЧНА СТРУКТУРА

(умовні оператори розгалуження)

*if* та *case*.

*Перша або повна форма:*

**if** < умова > **then**

**begin**

<серія 1>;

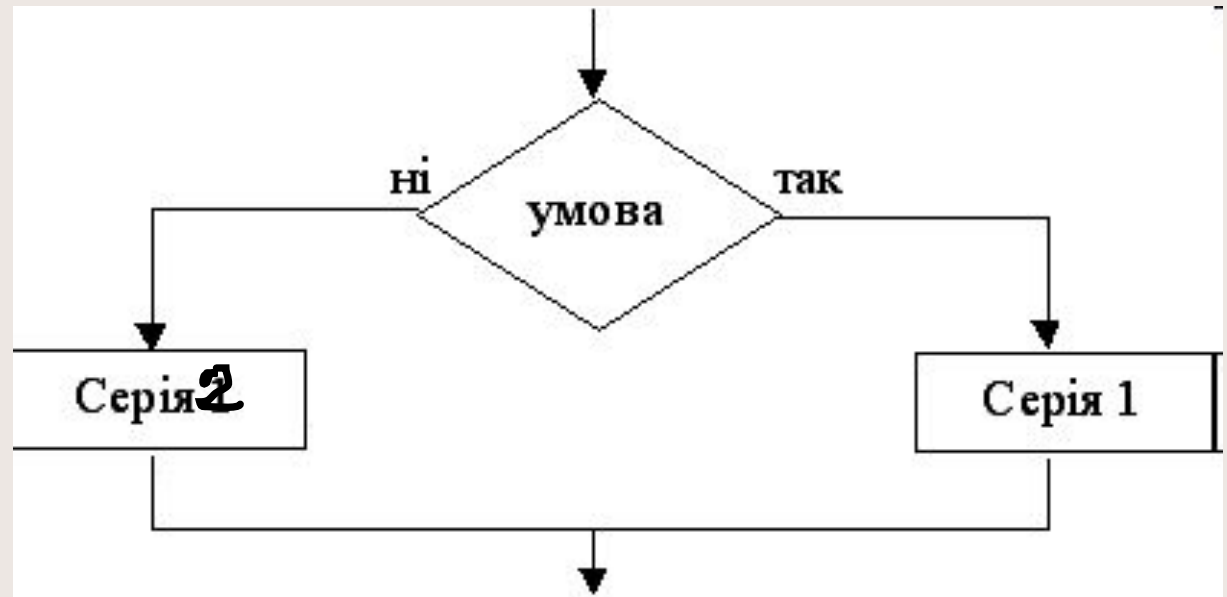
**end**

**else**

**begin**

<серія 2>;

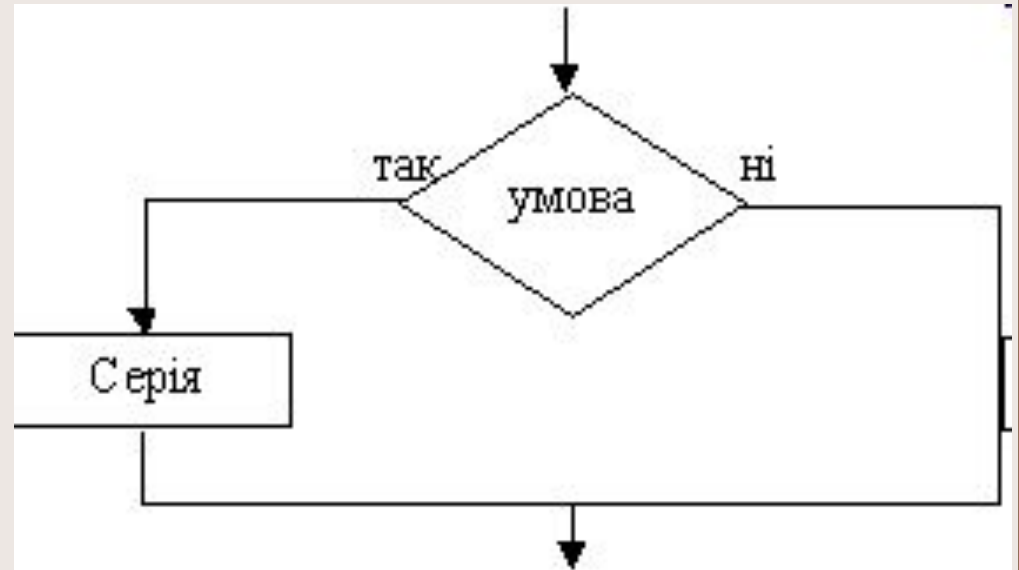
**end;**



**Перед else ; не ставлять**

## Друга або скорочена форма команди розгалуження:

```
if < умова > then  
begin  
<серія>;  
end;
```



**Логічний вираз (умова) може бути простий і складний**

## Вкладені умовні оператори

```
if умова1 then
    оператор1
else
    if умова2 then
        оператор2
    else
        оператор3
```

```
if умова1 then
    if умова2 then
        оператор
    else
        оператор
```

# Оператор вибору **CASE**

```
case <селектор> of  
  варіант : оператор;  
  ...  
  варіант : оператор;  
  [else оператор]  
end;
```

**Селектор** – змінна або вираз, який має довільний перелічуваний тип

## Приклад:

```
case ch of
```

```
  'A'..'Z', 'a'..'z' : WriteLn('Буква');
```

```
  '0'..'9' : WriteLn('Цифра');
```

```
  '+', '-', '*', '/' : WriteLn('Оператор');
```

```
  else WriteLn('Спеціальний символ')
```

```
end;
```