



# Тема уроку

## Алфавіт мови Pascal. Структура Pascal-програми.

```
ca Turbo Pascal
File Edit Search Run Compile Debug Tools Options
11.PAS
Program Celkoz;
Uses Crt;
Var
  f : text;
  Bals : array [1..25] of integer;
  St : array [1..25] of string;
  X,Stu : string;
  i,j,b,g,n : integer;
  c,ch : char;
Begin
  <$I->
  Assign(f,'b:\11\discipln.dat');
  Reset(f);
  <$I+>
  ClrScr;
  j:=1;
  While not SeekEof(f) do
  Begin
    Readln(f,Stu);
    St[j]:=Stu;
    j:=j+1;
  End
  1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+
```

# *План уроку*

- 1. Мова програмування PASCAL: історичні відомості .**
- 2. Алфавіт мови PASCAL.**
- 3. Структура PASCAL програми.**
- 4. Величини. Типи величин.**
- 5. Практична робота (картки).**
- 6. Підсумки уроку та домашнє завдання.**

# Історія

У 1970 р. була створена мова програмування TURBO PASCAL, названа так на честь видатного математика **Блеза Паскаля**.

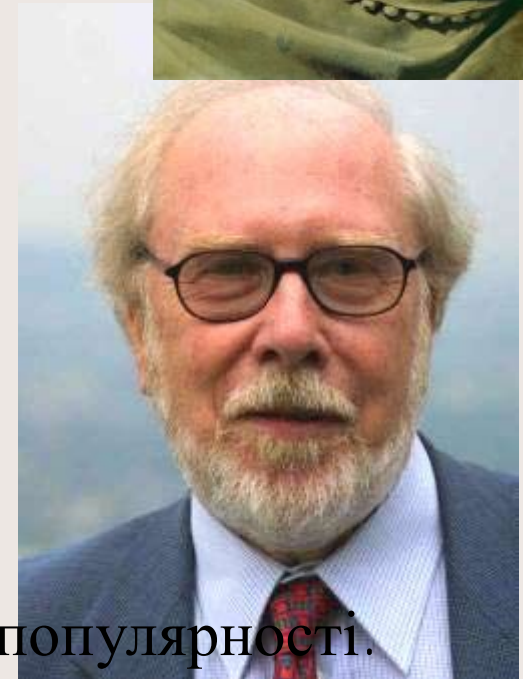


Автором мови був **Ніклаус Вірт** - професор та директор Інституту інформатики Швейцарської вищої політехнічної школи.

Мова PASCAL була дуже зручною для навчання студентів та для створення досить складних програм.

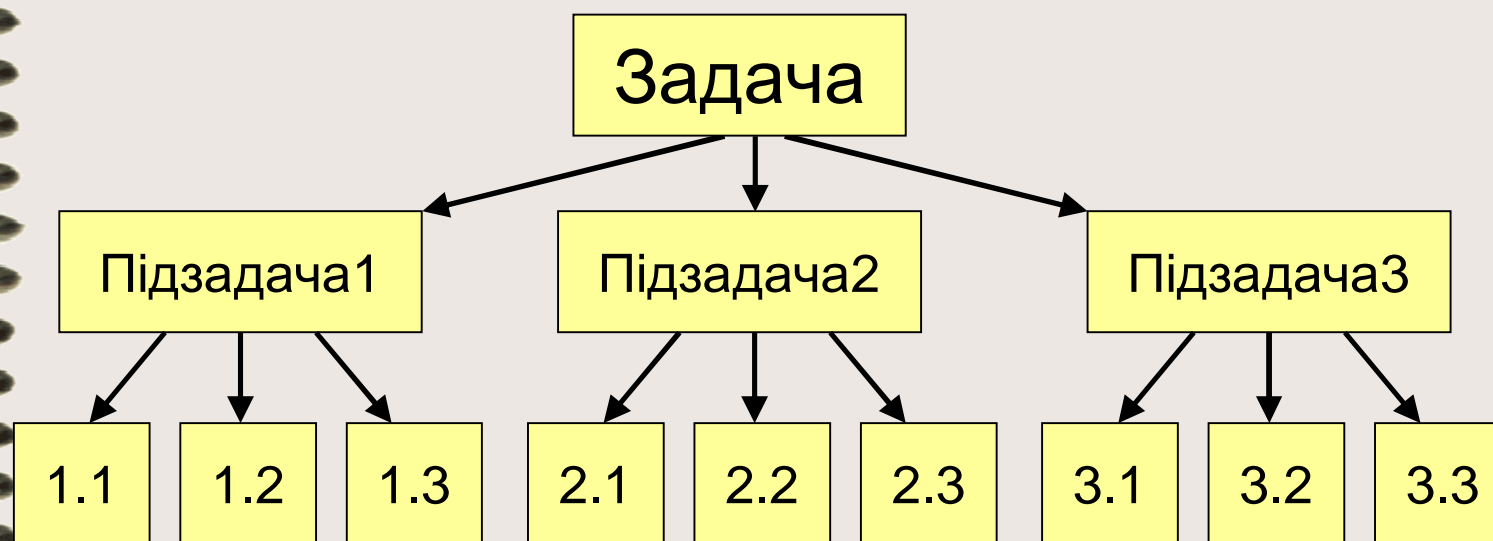
Дуже швидко PASCAL набула великої популярності.

*Ви вивчаєте її не для того, щоб стати програмістами, а щоб привести розум до порядку, навчитися логічно мислити, приймати рішення та робити в житті правильний вибір.*



**1970** – Ніклаус Вірт (Швейцарія)

- мова для навчання студентів
- розробка програм “зверху-вниз”



- різноманітні структури даних (масиви, структури, множини)

# Програма

---

**Програма** – це

- алгоритм, записаний на будь-якій мові програмування
- набір команд для комп'ютера

**Команда** – це опис дій, які повинен виконати комп'ютер.

- звідки отримати вихідні дані?
- що потрібно з ними зробити?

## З чого складається програма?

---

**program** *<ім'я програми>* ;

**const** ... ; *{константи}*

**var** ... ; *{змінні}*

*{ процедури і функції }*

**begin**

... *{основна програма}*

**end.**

коментарі у фігурних дужках не  
опрацьовуються

## З чого складається програма?

---

**Константа** – постійна величина, яка має ім'я.

**Змінна** – змінна величина, яка має ім'я (комірка пам'яті).

**Процедура** – додатковий алгоритм, який описує деякі дії (малювання кола).

**Функція** – додатковий алгоритм, для виконання обчислень (обчислення квадратного кореня, **sin**).

# Алфавіт мови

Алфавіт мови програмування – це набір символів, які можуть бути використані при складанні програми.

Програміст повинен знати, які символи “розуміє” дана мова програмування, щоб без помилок написати програму.





# Основні поняття мови

До основних понять мови програмування належать:  
оператори, ідентифікатори, числа, символи, рядки.

**Оператор** – це самостійна одиниця мови, яка описує зміст відповідного етапу алгоритмічного процесу. Алгоритм задається послідовністю операторів.

**Ідентифікатори або імена** використовуються для позначення змінних величин.



# Ідентифікатори

## Правила запису ідентифікаторів

Ідентифікатори позначаються тільки з букви або знаку підкреслення

Між двома ідентифікаторами повинен бути проміжок

Ідентифікатор не повинен співпадати за написанням ні з одним зі службових слів

Ідентифікатор може складатися з букв, цифр і знаку підкреслення

Максимальна довжина ідентифікатора 127 символів

Великі і малі букви в ідентифікаторах не розрізняються

# Імена програм, констант, змінних

---

## Імена можуть містити

- латинські букви (A-Z)

великі і маленькі букви не розрізняються

- цифри

ім'я не може починатися з цифри

- знак підкреслення \_

## Імена НЕ можуть містити

- українські букви
- пропуски
- дужки, знаки +, =, !, ? та ін.

## Які імена правильні?

AXby R&B 4Wheel Вася "PesBarbos"

TU154 [QuQu] \_ABBA A+B

# Константи

---

`const`

`i2 = 45; { ціле число }`

`pi = 3.14; { дійсне число }`

ціла і дробова частина відділяються крапкою

`qq = 'Вася'; { рядок символів }`

можна використовувати українські букви!

`L = True; { логічна величина }`

Може приймати два значення:

- True (істина, "так")
- False (хибність, "ні")

# Змінні

---

**Змінна** – це величина, яка має ім'я, тип і значення.

Значення змінної величини під час виконання програми може змінюватися.

## Типи змінних:

- integer { цілі }
- real { дійсні }
- char { один символ }
- string { рядок }
- boolean { логічні }

## Оголошення змінних (виділення пам'яті):

```
var a, b: integer;  
    Q: real;  
    s1, s2: string;
```

# Як змінюється значення змінної?

---

**Оператор** – це команда мови програмування високого рівня.

**Оператор присвоєння** служить для зміни значення змінної.

**Приклад:**

```
program qq;
```

```
var a, b: integer;
```

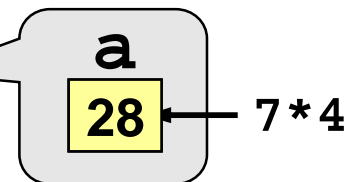
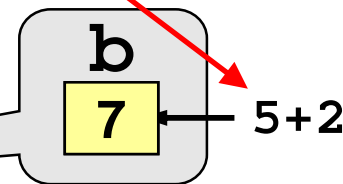
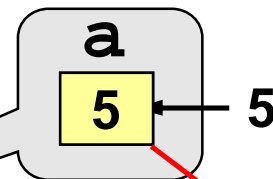
```
begin
```

```
  a := 5;
```

```
  b := a + 2;
```

```
  a := (a + 2) * (b - 3);
```

```
end.
```



# Оператор присвоєння

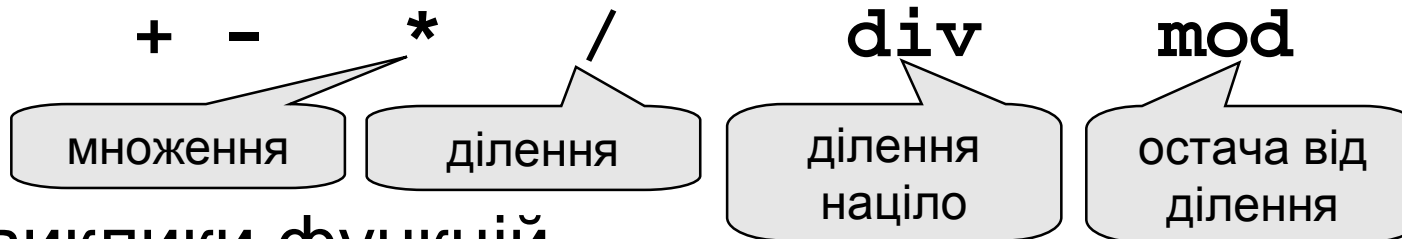
---

## Загальна структура:

*<ім'я змінної> ::= <вираз>;*

## Арифметичні вирази можуть містити

- константи
- імена змінних
- знаки арифметичних дій:



- виклики функцій
- круглі дужки ( )

# Які оператори неправильні?

---

```
program qq;  
var a, b: integer;  
    x, y: real;  
begin  
    a := 5;  
    10 := x;  
    y := 7,8;  
    b := 2.5;  
    x := 2*(a + y);  
    a := b + x;  
end.
```

ім'я змінної повинно  
знаходитися зліва від знака :=

ціла і дробова частина  
відділяються **крапкою**

неможливо записати дійсне  
значення в цілу змінну



# Ручна прокрутка програми

---

```
program qq;  
var a, b: integer;  
begin  
  a := 5;  
  b := a + 2;  
  a := (a + 2)*(b - 3);  
  b := a div 5;  
  a := a mod b;  
  a := a + 1;  
  b := (a + 14) mod 7;  
end.
```

a	b
?	?
5	
	7
28	
	5
3	
4	
	4

# Порядок виконання операцій

---

- обчислення виразів у дужках
- множення, ділення, `div`, `mod` зліва направо
- додавання і віднімання зліва направо

2 3 5 4 1 7 8 6 9

**z := (5\*a\*c+3\*(c-d))/a\*(b-c)/ b;**

$$z = \frac{5ac + 3(c-d)}{ab(b-c)}$$

$$x = \frac{a^2 + 5c^2 - d(a+b)}{(c+d)(d-2a)}$$

2 6 3 4 7 5 1 12 8 11 10 9

**x := (a\*a+5\*c\*c-d\*(a+b))/((c+d)\*(d-2\*a));**

# ВЕЛИЧИНИ

**ЗМІННІ**

**СТАЛІ**  
(КОНСТАНТИ)

**ОСНОВНІ  
ХАРАКТЕРИСТИКИ**

**ІМ'Я**  
(ідентифікатор)

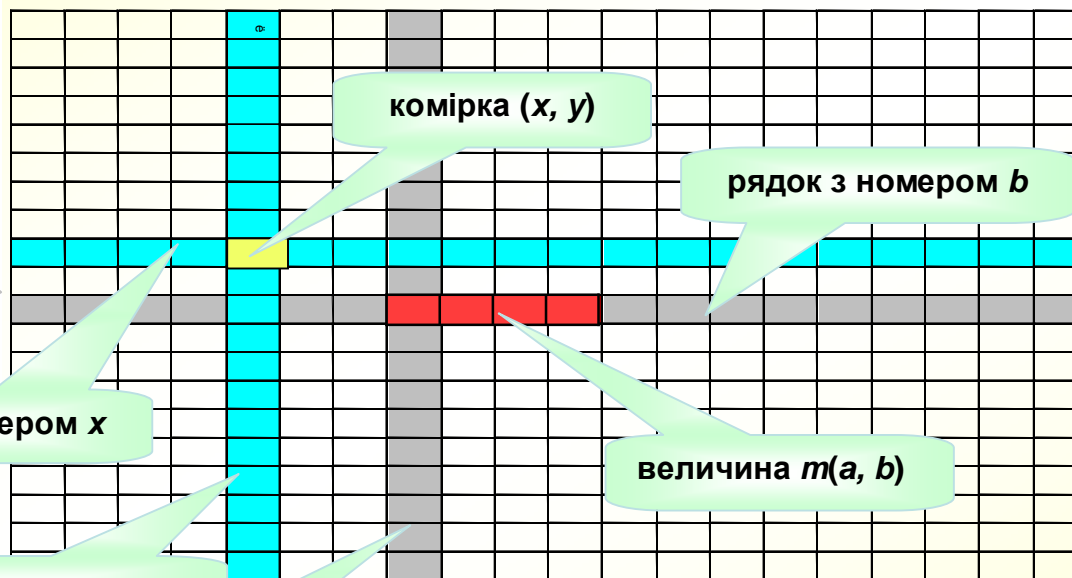
**ЗНАЧЕННЯ**

**ТИП**

# ЗМІСТ ПОНЯТТЯ ВЕЛИЧИНИ В ІНФОРМАТИЦІ

## МОДЕЛЬ ПАМ'ЯТІ КОМП'ЮТЕРА

(носії пам'яті – диск розбито на доріжки і сектори, на перетині яких розміщені комірки пам'яті. Номери доріжки  $x$  і сектора  $y$  – це координати комірки пам'яті, що знаходиться на їх перетині. Пам'ять зручно уявити, як прямокутник, що складається з однакових прямокутних клітинок – комірок  $(x,y)$ , де  $x$  – номер рядка, а  $y$  – стопчика, що містить комірку.



рядок з номером  $x$

доріжка з номером

стопчик з номером  $u$

стопчик з номером  $a$

комірка пам'яті  $(x, y)$

сектор з номером

величина  $m(a, b)$

## ВЕЛИЧИНА

займає одну або кілька сусідніх комірок пам'яті. Вона має ім'я (ідентифікатор), тип і значення. Координати величини  $m(a,b)$  – це номери рядка і стопчика її першої зліва комірки пам'яті.

Для різних типів величин відводить ся відповідна кількість сусідніх комірок пам'яті. Величини в програмах оголошують і описують. **Оголошити** величину – це вказати її ім'я, **описати** – вказати її тип.

<i>тип</i>	<i>допустимі значення</i>	<i>формат</i>
<b>SHORTINT</b>	<b>-128 .. 127</b>	<b>1 байт зі знаком</b>
<b>INTEGER</b>	<b>-32768.. 32767</b>	<b>2 байти зі знаком</b>
<b>LONGINT</b>	<b>-2147483648 .. 2147483647</b>	<b>4 байти зі знаком</b>
<b>BYTE</b>	<b>0 .. 255</b>	<b>1 байт без знака</b>
<b>WORD</b>	<b>0 .. 65535</b>	<b>2 байти без знака</b>

<i>тип</i>	<i>допустимі значення</i>	<i>точність</i>	<i>формат</i>
<b>REAL</b>	<b><math>2.9 \cdot 10^{-39} .. 1.7 \cdot 10^{+38}</math></b>	<b>11 — 12 знаків</b>	<b>6 байт</b>
<b>SINGL</b>	<b><math>1.5 \cdot 10^{-45} .. 3.4 \cdot 10^{+38}</math></b>	<b>7 — 8 знаків</b>	<b>4 байти</b>
<b>DOUBLE</b>	<b><math>5.0 \cdot 10^{-324} .. 1.7 \cdot 10^{-308}</math></b>	<b>15 — 16 знаків</b>	<b>8 байт</b>
<b>EXTENDED</b>	<b><math>3.4 \cdot 10^{-4932} .. 1.1 \cdot 10^{4932}</math></b>	<b>19 — 20 знаків</b>	<b>10 байт</b>
<b>COMP</b>	<b><math>- 9.2 \cdot 10^{+18} .. 9.2 \cdot 10^{+18}</math></b>	<b>19 — 20 знаків</b>	<b>8 байт</b>

# ЦІЛІ ТИПИ

<i>тип</i>	<i>допустимі значення</i>	<i>формат</i>
<b>SHORTINT</b>	<b>-128 .. 127</b>	<b>1 байт зі знаком</b>
<b>INTEGER</b>	<b>-32768.. 32767</b>	<b>2 байти зі знаком</b>
<b>LONGINT</b>	<b>-2147483648 .. 2147483647</b>	<b>4 байти зі знаком</b>
<b>BYTE</b>	<b>0 .. 255</b>	<b>1 байт без знака</b>
<b>WORD</b>	<b>0 .. 65535</b>	<b>2 байти без знака</b>

# ДРОБОВІ ТИПИ

<i>тип</i>	<i>допустимі значення</i>	<i>точність</i>	<i>формат</i>
<b>REAL</b>	<b><math>2.9 \cdot 10^{-39} .. 1.7 \cdot 10^{+38}</math></b>	<b>11—12 знаків</b>	<b>6 байт</b>
<b>SINGL</b>	<b><math>1.5 \cdot 10^{-45} .. 3.4 \cdot 10^{+38}</math></b>	<b>7—8 знаків</b>	<b>4 байти</b>
<b>DOUBLE</b>	<b><math>5.0 \cdot 10^{-324} .. 1.7 \cdot 10^{-308}</math></b>	<b>15—16 знаків</b>	<b>8 байт</b>
<b>EXTENDED</b>	<b><math>3.4 \cdot 10^{-4932} .. 1.1 \cdot 10^{4932}</math></b>	<b>19—20 знаків</b>	<b>10 байт</b>
<b>COMP</b>	<b><math>-9.2 \cdot 10^{+18} .. 9.2 \cdot 10^{+18}</math></b>	<b>19—20 знаків</b>	<b>8 байт</b>

**const**  $n=5$ ;

ТИП ЕЛЕМЕНТІВ МАСИВА - ДІАПАЗОН

**M : array**[1..n] **of** 0..9 = (8,2,1,0,7);

**type** **Mas** = **array**[1..n, 1..n] **of** **real**;

**var** **a, b, c, max** : **real**;

**st** : **string**; **d** : **byte**;

**Mas1** : **array**[1..n, 1..n] **of** **integer**;

**Mas2** : **Mas**;

ТАК ОГОЛОШУЮТЬСЯ КІЛЬКА  
ОДНОТИПНИХ ВЕЛИЧИН

ВЕЛИЧИНА **Mas2** МАЄ ОГОЛОШЕНИЙ  
ТИП  
**array**[1..n, 1..n] **of** **real**)

## АЛФАВІТ мови програмування PASCAL

### ЛІТЕРИ ЛАТИНСЬКОГО АЛФАВІТУ

(великі та малі) **A - Z, a - z**

### АРАБСЬКІ ЦИФРИ

**0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

### СПЕЦІАЛЬНІ СИМВОЛИ

**@, #, \$, &** та інші

### ЗНАКИ ОПЕРАЦІЙ

**-** (віднімання)                      **+** (додавання)  
**\*** (множення)                        **/** (ділення)  
**:=** (присвоювання)

### ЗНАКИ ВІДНОШЕНЬ

**<** (менше)            **<=** (не більше)    **>** (більше)  
**>=** (не менше)    **=** (дорівнює)    **<>** (не дорівнює)  
**in** (є елементом множини)

### РОЗДІЛОВІ ЗНАКИ

**.** – для відокремлення цілої частини від дробової;  
**,** – для відокремлення елементів списків;  
**;** – для відокремлення вказівок;  
**(\* \*)** або **{ }** – для запису коментарів;  
**( )** – відкриваюча і закриваюча дужки;  
**..** – задання діапазону;  
**[ ]** – для звернення до елементів масиву;  
**\$** – шістнадцятковий код;  
**^** – управляючий символ

Важливо вірно розуміти поняття “Алфавіт мови програмування Турбо Паскаль”, його зв’язок із загальним поняттям “алфавіт” та роллю в цій мові програмування. До цього нас зобов’язує велика формалізація в описі мови програмування.

Алфавіт асоціюється із мовою. У будь-якій мові є свій алфавіт - це допустимий набір символів, з допомогою яких записуються слова та речення конкретної мови.

Під поняттям “український алфавіт” звичайно розуміється набір символів (літер): “А”, “а”, “Б”, “б”, ..., “Я”, “я”. Вже звідси очевидне протиріччя, адже вважається, що в українській мові 33 літери, насправді ж їх 66. Але в дійсності, як легко переконатись, написати довільний текст українською мовою неможливо без розділових знаків, арабських цифр, врешті пропусків між словами. Тому у повному розумінні поняття “український алфавіт” суттєво відрізняється від його побутового трактування, на що у середній школі не звертається увага.

Тому, при ознайомленні з поняттям “Алфавіт мови програмування Турбо Паскаль” слід наголосити на цих нюансах, звернути увагу на його формальність.



## ЗАПЕРЕВОВАНІ СЛОВА В TURBO PASCAL 7

1	<b>ABSOLUTE</b>	21	<b>FUNCTION</b>	41	<b>PROGRAM</b>
2	<b>AND</b>	22	<b>GOTO</b>	42	<b>PUBLIC</b>
3	<b>ARRAY</b>	23	<b>IF</b>	43	<b>RECORD</b>
4	<b>ASM</b>	24	<b>IMPLEMENTATION</b>	44	<b>REPEAT</b>
5	<b>ASSEMBLER</b>	25	<b>IN</b>	45	<b>SET</b>
6	<b>BEGIN</b>	26	<b>INHERITED</b>	46	<b>SHL</b>
7	<b>CASE</b>	27	<b>INLINE</b>	47	<b>SHR</b>
8	<b>CONST</b>	28	<b>INTERFACE</b>	48	<b>STRING</b>
9	<b>CONSTRUCTOR</b>	29	<b>INTERRUPT</b>	49	<b>THEN</b>
10	<b>DESTRUCTOR</b>	30	<b>LABEL</b>	50	<b>TO</b>
11	<b>DIV</b>	31	<b>MOD</b>	51	<b>TYPE</b>
12	<b>DO</b>	32	<b>NEAR</b>	52	<b>UNIT</b>
13	<b>DOWNTO</b>	33	<b>NIL</b>	53	<b>UNTIL</b>
14	<b>ELSE</b>	34	<b>NOT</b>	54	<b>USES</b>
15	<b>END</b>	35	<b>OBJECT</b>	55	<b>VAR</b>
16	<b>EXTERNAL</b>	36	<b>OF</b>	56	<b>VIRTUA</b>
17	<b>FAR</b>	37	<b>OR</b>	57	<b>WHILE</b>
18	<b>FILE</b>	38	<b>PACKED</b>	58	<b>WITH</b>
19	<b>FOR</b>	39	<b>PRIVATE</b>	59	<b>XOR</b>
20	<b>FORWARD</b>	40	<b>PROCEDURE</b>		

**НЕ МОЖНА ВЖИВАТИ В ЯКОСТІ ІМЕН ВЕЛИЧИН, КОНСТАНТ, ПРОЦЕДУР, ФУНКЦІЙ**

## СПИСОК МАТЕМАТИЧНИХ ФУНКЦІЙ ТА ПРОЦЕДУР в Turbo Pascal 7

НАЗВА	РЕЗУЛЬТАТ ЗАСТОСУВАННЯ
<b>ABS(X)</b>	Модуль числа X
<b>COS(X)</b>	Косинус X (X – в радіа - нах)
<b>SIN(X)</b>	Синус X (X – в радіанах)
<b>ARCTAN(X)</b>	Арктангунс X (X – в раді - анах)
<b>EXP(X)</b>	$e^x$
<b>LN(X)</b>	Натуральний логарифм X
<b>PI</b>	Число $\pi$
<b>SQR(X)</b>	Квадрат числа X
<b>SQRT(X)</b>	Арифметичний квадратний корінь з X

### ФУНКЦІЇ

НАЗВА	РЕЗУЛЬТАТ ЗАСТОСУВАННЯ
<b>TRUNC(X)</b>	Число X, заокруглене, від киданням цифр числа, що стоять після коми
<b>FRAC(X)</b>	Дробова частина X
<b>INT(X)</b>	Ціла частина X
<b>ROUND(X)</b>	Ціла частина X, одержана заокругленням за прави - лами математики
<b>RANDOM(X)</b>	Випадкове число з д іапа - зону 0 .. X
<b>ODD(X)</b>	TRUE, якщо X непарне
<b>INC(X,Y)</b>	Збільшення X на Y
<b>DEC(X,Y)</b>	Зменшення X на Y
<b>RANDOMIZE</b>	Забезпечує неспівпадання послідовностей випад - кових чисел, отриманих з допо-могою RANDOM(X)

### ПРОЦЕДУРИ

# Додавання двох чисел

---

**Задача.** Ввести два цілих числа і вивести на екран їх суму.

**Найпростіше розв'язання:**

```
program qq;  
var a, b, c: integer;  
begin  
    read ( a, b );  
    c := a + b;  
    writeln ( c );  
end.
```



# Оператор виведення

---

`write ( a );` { вивести значення  
змінної `a` }

`writeln ( a );` { вивести значення  
змінної `a` і перейти на  
новий рядок }

`writeln ( 'Привіт!' );` { виведення  
тексту }

`writeln ( 'Відповідь: ', c );`  
{виведення тексту і значення  
змінної `c`}

`writeln ( a, '+', b, '=', c );`

# Формати виведення

---

```
program qq;  
var i: integer;  
    x: real;  
begin  
    i := 15;  
    writeln ( '>', i, '<' );  
    writeln ( '>', i:5, '<' );  
    x := 12.345678;  
    writeln ( '>', x, '<' );  
    writeln ( '>', x:10, '<' );  
    writeln ( '>', x:7:2, '<' );  
end.
```

ВСЬОГО  
СИМВОЛІВ

ВСЬОГО  
СИМВОЛІВ

В дробовій  
частині

```
>15<  
>  15<  
>1.234568E+001<  
> 1.23E+001<  
>  12.35<
```

# ПОВНИЙ РОЗВ'ЯЗОК

---

```
program qq;  
var a, b, c: integer;  
begin  
    writeln( 'Ввести два цілих числа' );  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
end.
```

## Протокол:

Ввести два цілих числа

25 30

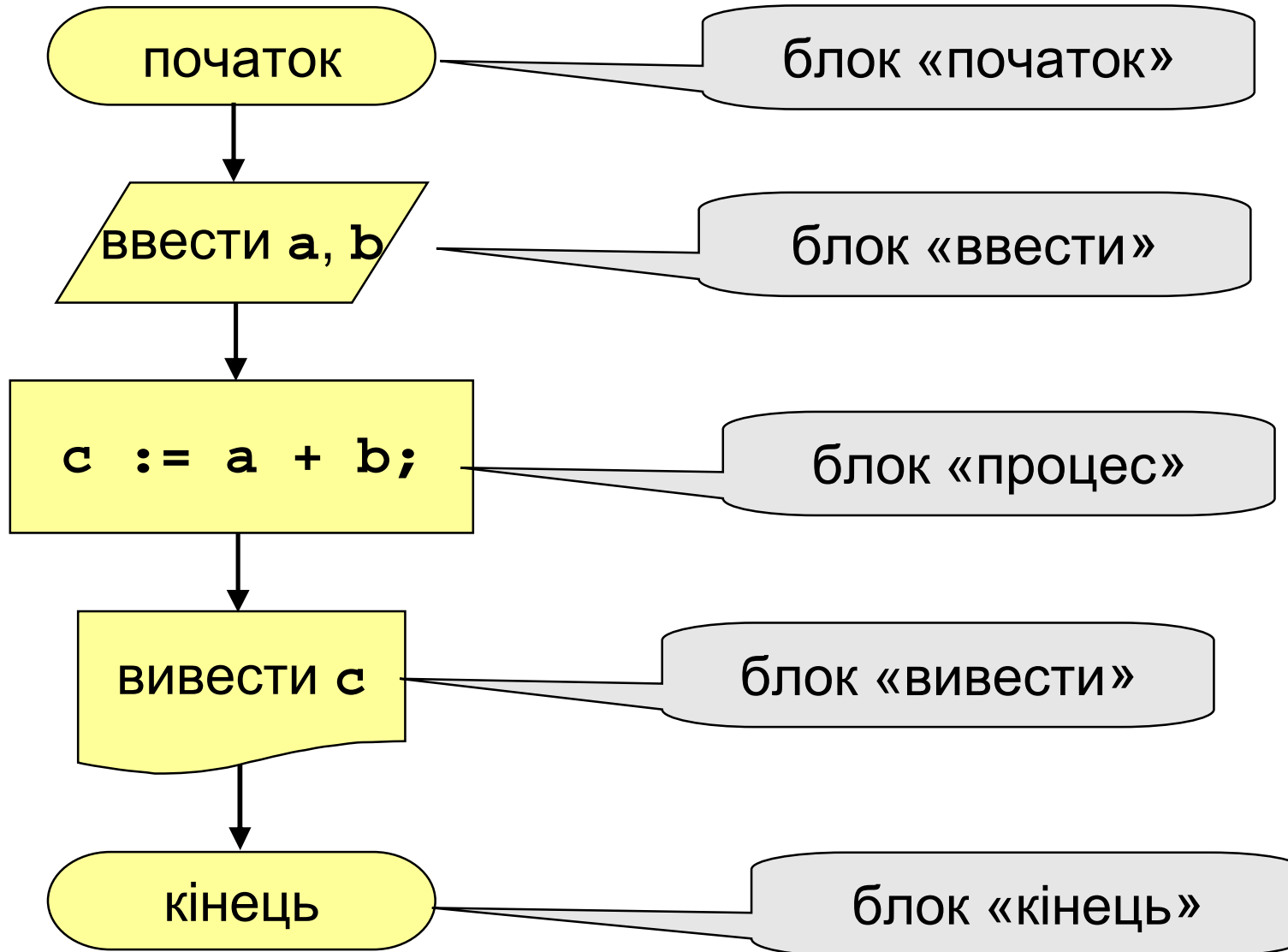
25+30=55

це виводить комп'ютер

це вводить користувач

# Блок-схема лінійного алгоритму

---





# Завдання

---

**"4":** Ввести три числа, знайти їх суму і добуток.

Приклад:

Ввести три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

**"5":** Ввести три числа, знайти їх суму, добуток і середнє арифметичне.

Приклад:

Ввести три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7)/3=5.33$$

# Структура Pascal програми

Програма складається із заголовка і блоку. Заголовок програми складається з службового слова **Program** та імені програми, після якого ставиться – ;

Блок програми може містити до 7 розділів, які записуються в такому порядку

**program** {ім'я програми} ;

**Users CRT** {розділ підключення модулів з системної бібліотеки} ;

**Label...** {розділ опису міток} ;

**Const...** {розділ опису констант} ;

**Type...** {розділ опису типів даних} ;

**Var...** {розділ опису змінних} ;

**Procedure** {розділ опису процедур і функцій} ;

**Begin**

...;{розділ вказівок – тіло програми}

**End.** **Модуль – це бібліотека констант, типів даних, змінних.**

Окремі розділи програми, крім розділу вказівок, можуть бути відсутніми. Між будь-якими двома вказівками ставиться крапка з комою, яка відокремлює їх одну від одної.

**Приклад 1:** скласти програму обчислення значення функції  $Y=(a+b-c)/(a-b)$  при  $a=6$ ,  $b=14$ ,  
 $c=4$

```
Program robota2;
```

```
Var
```

```
    Y:real;
```

```
    a,b,c:integer;
```

```
begin
```

```
    write('введіть значення a');
```

```
    readln(a);
```

```
    write('введіть значення b');
```

```
    readln(b);
```

```
    write('введіть значення c');
```

```
    readln(c);
```

```
    Y:=(a+b-c)/(a-b);
```

```
    writeln('значення функції Y=,'Y);
```

```
end.
```

Приклад 2: обчислити площу трикутника за формулою

Герона. Довжини сторін трикутника a, b, c.

```
Program robota2;  
  Var a,b,c:integer; p,S:real;  
  begin  
    write('введіть значення a');  
    readln(a);  
    write('введіть значення b');  
    readln(b);  
    write('введіть значення c');  
    readln(c);  
    p:=(a+b+c)/(2;  
    S:=sqrt(p*(p-a)*(p-b)*(p-  
c));  
    writeln('S=',S);  
  end
```

## Домашнє завдання:

- Вчити сторінки 61-73, п.12,13
- Підготуватися до тестової перевірки
- Підготуватися до практичної роботи

**Дякую за роботу на уроці!  
До нових зустрічей!!!**