



# **ЛЕКЦІЯ № 7**

**Базові засоби розробки і оформлення програм  
на мові Pascal.**



- 
- 
- 1. Загальні відомості про мови високого рівня.**
  - 2. Початкові відомості про мову Pascal.**
  - 3. Загальні положення програмування на алгоритмічній мові високого рівня Pascal.**

# Ніклаус Вірт. Професор, розробник мови Паскаль.



Дуже важливо твердо засвоїти правила запису алгоритма рішення задачі на Turbo Pascal(Pascal ABC і надалі просто Pascal) з виконанням правил його синтаксису.

## 1. Загальні відомості про мови високого рівня.

Програмування є процес написання алгоритму рішення задачі мовою зрозумілою ЕОМ.

При цьому для АМВР, як і для будь-якої іншої мови повинні бути визначені:

- символіка, тобто припустимі символи в мові;
- синтаксис, тобто правила утворення конструкцій мови із символів;
- семантика, тобто значеннєвий зміст одержуваних конструкцій.

Символіка, тобто набір символів мови Pascal є підмножиною набору символів коду ASCII і значною мірою збігається із символікою, прийнятої в системі програмування АЛГОРИТМ.

Він (набір) включає:

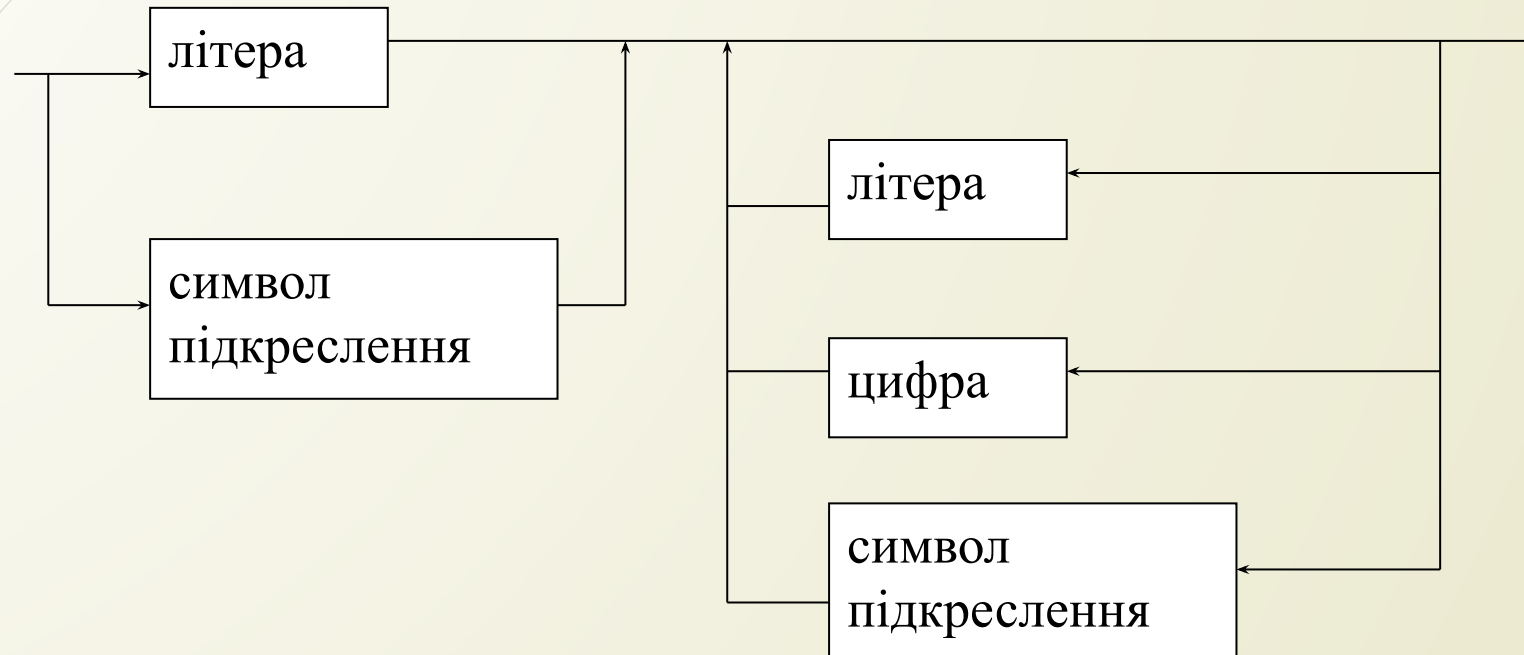
- прописні і малі літери латинського алфавіту (A-Z, a-z, 26 букв) а також символ підкреслення, що використовується нарівні з буквами, коди (65 - 90), (97-122), 95 відповідно;





Ці символи застосовуються для побудови ЛЕКSEM, тобто мінімальних значимих одиниць тексту програми.

Ідентифікатори (імена) утворюються за наступною схемою



Ідентифікатори виступають як імена констант, типів, змінних, масивів, процедур, модулів, функцій, програм і полів у записах.

Розходжень між словами, записаними прописними і малими літерами не робиться.

Варто мати на увазі ту обставину, що не можна як ідентифікатори використовувати службові (зарезервовані) слова.

<i>and</i>	<i>goto</i>	<i>program</i>
<i>asm</i>	<i>if</i>	<i>record</i>
<i>array</i>	<i>implementation</i>	<i>repeat</i>
<i>begin</i>	<i>in</i>	<i>set</i>
<i>case</i>	<i>inherited</i>	<i>shl</i>
<i>const</i>	<i>inline</i>	<i>shr</i>
<i>constructor</i>	<i>interface</i>	<i>string</i>
<i>destructor</i>	<i>label</i>	<i>then</i>
<i>div</i>	<i>library</i>	<i>to</i>
<i>do</i>	<i>mod</i>	<i>type</i>
<i>downto</i>	<i>nil</i>	<i>unit</i>

АМВР дозволяє представляти програму на деякій проміжній мові між людиною і внутрішньою мовою ЕОМ, яку вони розуміють.

Для перетворення програми з АМВР на внутрішню мову ЕОМ застосовуються спеціальні програмні засоби, іменовані *трансляторами*.

Транслятори бувають двох типів: *інтерпретатори* і *компілятори*.

*Інтерпретатори* обробляють оператори при запуску програми на виконання по мірі їхнього виконання.

*Компілятори* обробляють всі оператори програми і формують готовий програмний модуль до виконання при виконанні режиму компіляції.

## 2. Початкові відомості про мову Pascal.

АМВР Turbo Pascal оперує з типізованими структурами даних. Тому дані, що програміст припускає використовувати в програмі, обов'язково повинні бути оголошені завчасно, тобто віднесені до конкретного типу. Це робиться в розділі описів. Варто строго виконувати такі вимоги мови: змінна чи інший представник даних повинні бути описані тільки один раз, тобто не можна той самий об'єкт віднести до різних типів.

Turbo Pascal припускає досить широку палітру типів, у тому числі і користувальницький тип, тобто такий, котрий уводить (придумує) для себе сам програміст. Важливо, щоб це робилося з дотриманням прийнятих для мови правил.

Сьогодні ми назвемо основні (базові) типи даних, щоб пояснити суть процесу опису.



## Група цілих типів

Назва типу	Ідентифікатор	Діапазон представлення чисел	Розмір пам'яті
Коротке ціле	Shortint	-128..127	1 байт
Ціле зі знаком	Integer	-32768..32767	2 байти
Довге ціле зі знаком	Longint	-2147483648.. 2147483647	4 байти
Коротке ціле без знака	Byte	0..255	1 байт
Ціле без знака	Word	0..65535	2 байт

## Група дійсних типів

<i>Назва типу</i>	<i>Идентифікатор</i>	<i>Діапазон представлення чисел</i>	<i>Значущі цифри мантиси</i>	<i>Розмір пам'яті</i>
Дійсьне одинарної точності	Single	від $1.5 \cdot 10^{-45}$ до $3.4 \cdot 10^{38}$	7..8	4 байти
Дійсьне	Real	від $2.9 \cdot 10^{-39}$ до $1.7 \cdot 10^{38}$	11..12	6 байтів
Дійсьне подвійної точності	Double	від $5.0 \cdot 10^{-4932}$ до $1.7 \cdot 10^{4932}$	15..16	8 байтів
Дійсьне підвищеної точності	Extended	від $3.4 \cdot 10^{-45}$ до $1.1 \cdot 10^{38}$	19..20	10 байтів
Ціле у фор-маті дійсного	Comp	від $-2^{63} + 1$ до $2^{63} - 1$ чи приблизно від $-9.2 \cdot 10^{18}$ до $9.2 \cdot 10^{18}$	19..20	8 байтів



## Група булевських типів

*Boolean , ByteBool, WordBool, LongBool.*

### СИМВОЛЬНИЙ ТИП

*Char*

### РЯДКОВИЙ ТИП

*String*

### ТЕКСТОВИЙ ТИП

*Text*

### ТИП МАСИВ

*Array [Поч.знач. індексу .. кінцеве. Знач. індексу] of < тип елемента масиву >*

### 3. Загальні положення програмування на алгоритмічній мові високого рівня (АМВ) Pascal. Неформальна схема програми на мові Pascal.

Неформальну структуру програми можна представити у такому вигляді:

```

    {I. Заголовок програми}
program <пробіл> <Ім'я _програми>;

    {II. Розділ указівок використуваних модулів }
uses      Список_ Використовуваних_ Модулів;

    {III. Розділ описів }

    label
    const
    type
    var
    procedure
    function
    exports

    Опис_Міток;
    Опис_Констант;
    Опис_Типів;
    Опис_Змінних;

    Опис_Процедур_і_Функцій

    Опис_Експортованих_Імен;

    {IV. Розділ операторів (Операторний блок)}

    BEGIN

        Оператори

    END.
```

Деякі елементи наведеної схеми можуть бути відсутніми. Наприклад, може не бути міток, констант, типів. Однак варто пам'ятати, що програма мовою **Pascal** в обов'язковому порядку повинна починатися з розділу опису змінних (можливо і масивів), у якому оголошуються *всі* змінні, використовувані в програмі. Розділ опису змінних починається зі слова **VAR**. За розділом опису змінних порядкує розділ інструкцій. Розділ інструкцій починається зі слова **BEGIN** й закінчується словом **END**, за яким порядкує символ "крапка".

Наприкінці кожної інструкції ставиться символ "крапка з комою". Звичайно кожен рядок інструкції пишуть на окремому рядку.

Наприклад,

*VAR*

{ тут опис перемінні програми }

***begin***

{ тут виконувани інструкції програми }

***end.***



Для полегшення розуміння тексту програми, у текст можуть включатися *коментарі* - будь-який текст, заключений у фігурні дужки (або обмежений символами (\* й \*)), як правило такий текст, на відміну від іншої частини програми, для наочності виділяється іншим кольором і стилем {*наприклад, курсивом*}.

## **Наприклад,**

[*Цей текст є коментарем*] чи  
(\**Цей текст є коментарем*\*).

Коментарі звичайно розташовуються на окремому рядку чи наприкінці рядка тексту програми після інструкцій.

Коментар може займати кілька рядків тексту програми.

Кожен елемент програми (змінна, константа, оператор, мітка, масив і т.д. повинні мати ім'я, тобто ідентифікатор.

- ім'я ідентифікатора складається з допомогою латинських букв, арабських цифр і символ підкреслення;
- забороняється використовувати в якості ідентифікаторів зарезервовані слова;
- ідентифікатор не може починатися з цифри;
- ідентифікатор повинен починатися з букви чи символа підкреслення;
- ідентифікатор не може містити символ пробілу;

# Таблиця відповідності операторів мови Паскаль та блоків системи «Алгоритм».

№	УПБ	Символ в системі алгоритм	Тип блоку – оператор	Аналог мовою Паскаль
1	Нач	ТЕРМІНАТОР	Початок алгоритму або процедури	BEGIN
2	Кон	ТЕРМІНАТОР	Кінець алгоритму або процедури	END.
3	НОР	ПРОЦЕС	«Порожній блок»	Оператор, що не виконує жодних дій. Наприклад, BEGIN END; repeat until keypressed;
4	А..	ПРОЦЕС	Присвоювання (Assignment)	<ідентифікатор>:=<вираз>;
5	І..	ДАНІ	Введення даних (Input)	READ(ел1,ел2,...,елn); READLN(.....);
6	О..	ДАНІ	Виведення даних (Output)	WRITE(ел1,ел2,...,елn); WRITELN(.....);
7	Д..	РІШЕННЯ	Перевірка виконання умови (Decision)	If <лог. вираз> then <дія> else <дія>;
8	Ф..	ПОЧАТОК ЦИКЛУ	Початок циклу з параметром (For)	For <пар.циклу>:=<початкове значення> to <кінц.знач.> do<оператор>;
9	W..	ПОЧАТОК ЦИКЛУ	Початок циклу з передумовою (While)	WHILE <умова> DO <оператор>;
10	Р..	ПОЧАТОК ЦИКЛУ	Початок циклу з післяумовою (Repeat)	REPEAT <тіло циклу> UNTIL <умова>;
11	Е..	КІНЕЦЬ ЦИКЛУ	Кінець циклу з параметром чи передумовою (End)	
12	U..	КІНЕЦЬ ЦИКЛУ	Кінець циклу з післяумовою (Until)	
13	Р..	ПРОЦЕДУРА	Виклик процедури (Procedure)	PROCEDURE<ім'я>(<список параметрів >);

## **Заключення**

Як впливає із вище наведеного програма на мові Pascal повинна бути створеною з суворим дотриманням певних вимог, які є обов'язковими для користувача.

В зв'язку з цим вам належить ретельно вивчити названі правила і дотримуватися їх при створенні програм у подальшій роботі.

## **ДОМШНЄ ЗАВДАННЯ**