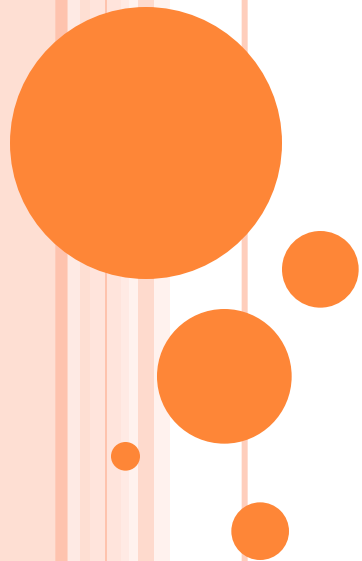


***ПЕРШЕ ЗНАЙОМСТВО З  
МОВОЮ  
ПРОГРАМУВАННЯ  
ПАСКАЛЬ***



**Уся НАША ГІДНІСТЬ МІСТИТЬСЯ В  
ДУМЦІ. НЕ ПРОСТІР І НЕ ЧАС, ЯКІ МИ НЕ  
МОЖЕМО ЗАПОВНИТИ, ПІДНІМАЮТЬ НАС,  
А САМА НАША ДУМКА.**

**Тому НАВЧАЙМОСЯ ПРАВИЛЬНО  
МИСЛИТИ...**

***БЛЕЗ ПАСКАЛЬ***

# СТВОРЕННЯ МОВИ ПРОГРАМУВАННЯ ПАСКАЛЬ

Мову програмування *Паскаль* для навчання студентів основам програмування створив у 1968-1971 р.р. швейцарський учений *Ніклаус Вірт* з кафедри інформатики Стенфордського університету.

Таку назву вона отримала на честь відомого французького математика, фізика та філософа *Блеза Паскаля*.



# ПЕРЕВАГИ МОВИ ПРОГРАМУВАННЯ ПАСКАЛЬ

- **Простий синтаксис мови** – програми на *Паскалі* легко читати та редагувати, оскільки вони містять невелику кількість базових понять.
- **Досить низькі апаратні та системні вимоги** як для роботи самого середовища програмування, так і для програм, записаних мовою *Паскаль*.
- **Універсальність мови**, тобто можливість використовувати цю мову програмування для вирішення задач різних класів та сфер діяльності.
- **Підтримка структурного та об'єктно-орієнтованого програмування.**



# *АЛФАВІТ МОВИ ПАСКАЛЬ*

*Алфавіт* – це сукупність символів, що дозволені до використання при записі програми певною мовою програмування.

*Алфавіт мови Паскаль складається з*

- великих та малих літер латинського алфавіту та знаку підкреслення;
- десяткових та шістнадцяткових цифр;
- спеціальних символів:



# СПЕЦІАЛЬНІ СИМВОЛИ МОВИ ПАСКАЛЬ

Сим-вол	Назва	Сим-вол	Назва	Сим-вол	Назва	
+	плюс	( )	дужки	'	апостроф	
-	мінус	{ }		фігурні	#	номер
*	множення	[ ]		квадратні	@	комерційне at
/	ділення	.	крапка	^	тильда	
=	дорівнює	,	кома	\$	знак грошової одиниці	
>	більше	;	крапка з комою		пропуск	
<	менше	:	двокрапка			

Комбінації спеціальних символів можуть утворювати **складені символи**. Наприклад, <= (менше або дорівнює), := (присвоєння). Такі символи Паскаль сприймає як один символ.

**Роздільник (пропуск) між елементами таких символів неприпустимий.** З символів алфавіту мови Паскаль конструюють **слова**, які відділяють одне від одного **розділовими символами**.



# СЛОВА МОВИ ПРОГРАМУВАННЯ ПАСКАЛЬ

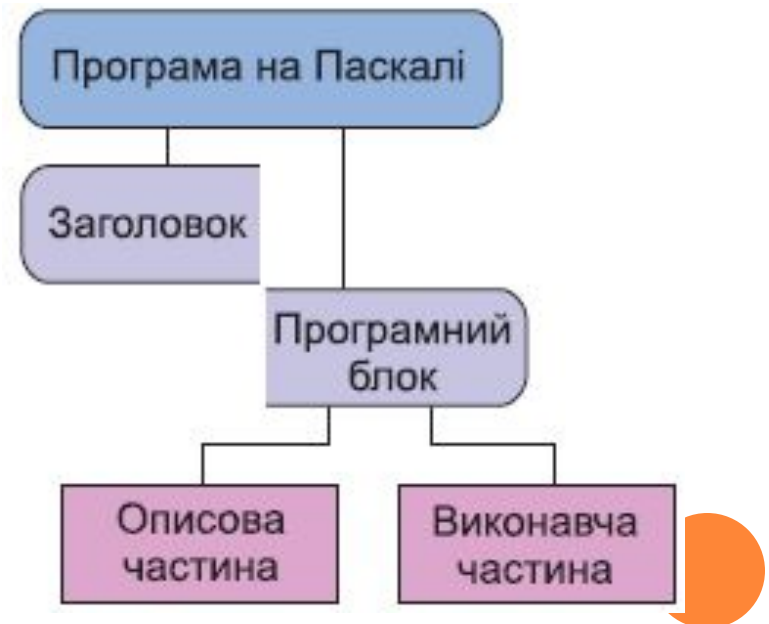
- ▣ *Зарезервовані (службові)* – слова, що є складовою частиною мови. Вони мають заздалегідь визначене написання та завжди однаковий зміст.
- ▣ *Стандартні* – слова, призначені для заздалегідь визначених розробниками мови типів даних, сталих величин, певних процедур або функцій.
- ▣ *Ідентифікатори (імена величин)* – слова, що визначає користувач для імен сталих та змінних величин, власних процедур або функцій.



# СТРУКТУРА ПРОГРАМИ МОВОЮ ПАСКАЛЬ

НАМ	Паскаль
<u>Програма</u> <назва програми>; <u>Початок</u> <тіло програми> <u>Кінець</u> .	<u>Program</u> <назва програми>; <u>Begin</u> <тіло програми> <u>End</u> .

Program <назва програми>;  
    Uses <список імен бібліотечних модулів,  
        що використані у програмі>;  
    Const <описання сталих величин>;  
    Type <описання типів користувача>;  
    Var <описання змінних величин>;  
        <описання процедур та функцій  
користувача>;  
    Begin  
        <тіло програми>  
    End.





# ПЕРША ПРОГРАМА МОВОЮ ПАСКАЛЬ

Коли навчальна експедиція Амазонкою закінчувалася, Айбоша на згадку про подорож вирішив зробити спільну фотографію всіх нових друзів, які зібралися разом на великій галявині.

У перші  $x$  рядів встали маленькі мавпочки по  $y$  тварин у кожному ряді. За ними стояли  $a$  рядів травоїдних та гризунів, по  $b$  тварин у кожному ряду. За тваринами зручно розмістилися учені та члени екіпажу теплоходу «Дельфін». Їх було стільки, скільки мавпочок, травоїдних та гризунів разом. За фотографуванням з-за дерев спостерігало  $z$  хижаків, які не бажали налякати друзів, тому вони поки що не наважувалися вийти із схованки.

Скільки всього нових знайомих зібралося фотографуватися та спостерігати за цим процесом на галявині?

# ВИЗНАЧЕННЯ ЗМІННИХ

$x, a$  – кількість рядів на фотографії;

$y, b, z$  – кількість тварин у рядах та хижаків за деревами.

Змінимо імена цих змінних (аргументів), враховуючи **правила створювання імен**, які діють у Паскалі:

- Ім'я сталої або змінної величини **починається** тільки з літери (або знаку підкреслення) і **може містити** літери, цифри та знак підкреслення.
- Між двома ідентифікаторами (іменами) обов'язково має бути хоча б один розділовий символ.
- Максимальна довжина імені становить 127 символів, але враховуються лише перші 63 з них.
- Великі та маленькі літери компілятор Паскаля не розрізняє.

# ВИЗНАЧЕННЯ ЗМІННИХ

*zagal* - результат, у цій змінній будемо зберігати загальну кількість друзів, що фотографуються та спостерігають за цим на галявині.

Крім цих змінних величин доцільно ввести такі:

*mapra* – проміжна змінна, призначена зберігати кількість мавпочок на галявині;

*trav\_griz* – проміжна змінна, призначена для збереження загальної кількості трав'яних та гризунів.



# Типи даних мови Паскаль

Скалярні  
(прості)

+

Структуровані  
(складені)

Стандартні  
типи

Типи  
користувача

Цілі (Integer)

Дійсний (Real)

Літерний (Char)

Логічний (Boolean)

Масиви

Множини

Файли

Записи

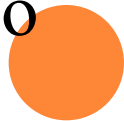
Об'єкти

Посилання



Назва цілочисельного типу	Діапазон значень	Обсяг пам'яті (байт)
Коротке ціле без знаку <i>Byte</i>	0...255	1
Коротке ціле із знаком <i>ShortInt</i>	-128...127	1
Ціле без знаку <i>Word</i>	0...65535	2
Ціле із знаком <i>Integer</i>	-32768...32767	2
Довге ціле із знаком <i>LongInt</i>	-2147483648...2147483647	4

Так само, як і в *Навчальній Алгоритмічній Мові*, в *Паскалі* визначені:

- **арифметичні операції** (+, -, \*, /), які застосовні для змінних цілого та дійсного типів);
  - **операції відношення** (порівняння) (=, <, >, <=, >=, < >), які порівнюють значення двох величин (операндів) і визначають *істинним* чи *хибним* є значення всього виразу;
  - **логічні операції** (логічні та (*and*), або (*or*), ні (*not*)), результатом яких є логічне значення *true* або *false*.
- 

# ДЕТАЛІЗАЦІЯ БЛОКІВ ПРОГРАМИ

Для визначення кількості мавпочок, травоїдних та гризунів потрібно кількість рядів, якими вишукувалися тварини, помножити на кількість тварин у кожному ряді:

$mavpa := x * y$ ; {визначення кількості мавпочок}

$trav\_griz := a * b$ ; {визначення кількості травоїдних та гризунів}

Для визначення загальної кількості друзів необхідно додати кількість мавпочок, травоїдних та гризунів, вчених та членів екіпажу (для визначення цього значення ми повинні додати попередньо визначені значення кількості мавпочок та травоїдних і гризунів), хижаків та 1 (щоб не забути додати до загальної кількості друзів самого Айбошу):

$zagal := mavpa + trav\_griz + (mavpa + trav\_griz) + z + 1$ ;



Program foto;

Var  $x, a, y, b, z, mavpa, trav\_griz, zagal$ : byte ;

Begin

<введення значень аргументів>;

$mavpa := x * y$ ; {кількість мавпочок}

$trav\_griz := a * b$ ; {кількість травоїдних та гризунів}

$zagal := mavpa + trav\_griz + (mavpa + trav\_griz) + z + 1$ ;  
{загальна кількість}

<виведення отриманого результату>

End.





# ПРОЦЕДУРИ ВВЕДЕННЯ ТА ВИВЕДЕННЯ

Read (<імена змінних, що відокремлені одне від одного комою>);

Write (<імена змінних, що відокремлені одне від одного комою>);

Використовуючи процедуру Write, можна вивести на екран значення дійсного числа у зручному вигляді.

Для цього використовують *спеціальний формат виведення*, аналогічно до того, як ми це робили у *НАМ*:

Write (<*ім'я змінної*>:*n*:*m*);

де *n* – кількість позицій для виведення на екран дійсного числа (знак «-» та десяткова кома входять у це число),

*m* – кількість позицій (з *n* можливих) для виведення цифр після десяткової коми.



При використанні такого формату виведення дійсних чисел потрібно враховувати дію наступних *правил*:

- *Якщо вказана кількість позицій буде більшою*, ніж необхідно, то перед цілою частиною числа виведуться пропуски.
- *Якщо кількість позицій для цілої частини буде замалою*, то компілятор автоматично додасть при виведенні необхідну кількість позицій.
- *Якщо кількість позицій для дробової частини буде недостатньою*, відбудеться виведення заокругленого значення дробової частини.



Program foto;

Var *x, a, y, b, z, mavpa, trav\_griz, zagal*: byte ;

Begin

write ('Введіть кількість рядів мавпочок =>');

readln (*x*);

write ('Введіть кількість мавпочок у ряді =>');

readln (*y*);

write ('Введіть кількість рядів травоїдних та гризунів =>');

readln (*a*);

write ('Введіть кількість травоїдних та гризунів у ряді =>');

readln (*b*);

write ('Введіть кількість хижаків =>');

readln (*z*);

*mavpa* := *x* \* *y*; {кількість мавпочок}

*trav\_griz* := *a* \* *b*; {кількість травоїдних та гризунів}

*zagal* := *mavpa* + *trav\_griz* + (*mavpa* + *trav\_griz*) + *z* + 1; {загальна кількість}

write ('Загальна кількість друзів, що фотографуються та спостерігають за цим', *zagal*)

End.




*При записі програми мовою Паскаль потрібно виконувати наступні правила:*

- Програма, записана мовою Паскаль, повинна мати вищеописану структуру. Вона складається із рядків довільної довжини, що містять довільну кількість операторів. Максимальна довжина рядка становить 127 символів. Але для того, щоб текст програми можна було легко читати та виправляти, радимо вам використовувати правило: *«В одному рядку – один оператор»*.
- Крапка з комою розділяє оператори програми, тому відсутність її між ними компілятор сприймає як помилку.



*При записі програми мовою Паскаль потрібно виконувати наступні правила:*

- У кінці програми ставиться не крапка з комою, а **крапка**.
  - Крапка з комою **не ставиться** після службового слова begin та перед службовим словом end, оскільки вони складають операторні дужки (так само, як у *НАМ* ми використовували операторні дужки початок-кінець).
  - У середині умовних операторів (операторах розгалуження) та операторів повторення (циклічних операторах) крапка з комою не ставиться. *Винятком* є випадок, коли інші оператори розміщені в операторних дужках begin-end, що містяться всередині умовного оператора чи оператора циклу.
- 

*При записі програми мовою Паскаль потрібно виконувати наступні правила:*

- У будь-якому місці програми можна використовувати **коментарі**, які не мають обмежень за довжиною та позначаються за допомогою фігурних дужок { }. У них варто зберігати пояснення до програми, що дозволить вам у разі потреби швидко відкоригувати написану програму.



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

1)  $18,9+27$

**$18.9+27$**



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

2)  $8,234+6 \cdot 7,54$

**$8.234+6*7.54$**





# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

3)  $\frac{1}{5} + \frac{1}{7}$

**1/5+1/7**



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

$$4) 14,37-5,25:3+12,2 \cdot (13,4-18,978)$$

$$14.37-5.25/3+12.2*(13.4-18.978)$$

# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

$$5) \quad \frac{2 + 3,5 \cdot 4,8}{4 : 3 - 5,21}$$

$$(2+3.5*4.8)/(4/3-5.21)$$

# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Запишіть вираз за правилами мови *Паскаль*

$$6) \quad \frac{3,14 + 2,76 : 2,12 - 5,9}{76,45 \cdot 4,89 : 2 + 3,6}$$

$$(3.14+2.76/2.12-5.9)/$$
$$(76.45*4.89/2+3.6)$$

# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

1) *write* (178.32:6:1);



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

2) *write* (-23.04:8:4);



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

3) *write* (17+4.335:2, '13+6.7');

**13+6,7**



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

4) *write* ('0.287+17.5', 2.9:6:3);

**0.287+17.5**





# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

5) *write* (10, 12:2:3, 345.2, -5);



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

б) *write* (-2:1, -2:2:1, -0.21:5:2);



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

7) *write* ('0.567:5:4',12, -4.8:3);

**0.567:5:4**



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що буде виведено на екран при виконанні оператора?

8) *write* (-78.9:8:5, '-67.3:4:1');

**-67.3:4:1**



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Що виведе на екран програма, якщо користувач введе такі значення аргументів?

```
Program Ex1;  
  Var x: integer;  
      y: real;
```

```
Begin  
  Write ('Введіть значення x =>');  
  Readln (x);  
  Write ('Введіть значення y =>');  
  Readln (y);  
  Write (x*y)  
End.
```

<u>№</u>	<u>x</u>	<u>y</u>
1	4	0.23
2	-6	3.5
3	2.8	5

1) **0,92**

2) **-21**

3) **14**

# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Знайдіть та поясніть помилки у наведених текстах програм:

```
Program Ex8_1;  
Const c=5;  
Begin  
    c:=c+sqr(c);  
    Write ('Нове значення c= ', c')  
End
```



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Знайдіть та поясніть помилки у наведених текстах програм:

```
Program Ex8_1;  
  Const c=5;  
Begin  
  c:=c+sqr(c);  
  Write ('Нове значення c= ', c)  
End
```



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Знайдіть та поясніть помилки у наведених текстах програм:

```
Program Ex8_2;  
  Var a, b, c: real;  
Begin  
  Write ('Введіть значення a =>');  
  Readln (a)  
  Write ('Введіть значення c =>');  
  Readln (c);  
  Write ((a*b+c)/2')  
End.
```



# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Знайдіть та поясніть помилки у наведених текстах програм:

```
Program Ex8_3;  
  Var x, y: real; a: word;  
Begin  
  Write ('Введіть значення x та y =>');  
  Readln (x, y);  
  Write ('Введіть значення a =>');  
  Readln (a);  
  a:=x+y*3,14+2.55;  
  Write ('Нові значення змінних,' x:2:4, y, a:3)  
End.
```

# НАВЧАЛЬНО-ТРЕНУВАЛЬНІ ЗАВДАННЯ

Обчисліть значення всіх змінних після виконання операторів:

№	Оператори	Значення змінних
1	$x:=1; x:=x*x+12;$	$x= 13$
2	$x:=0; y:=5; x:=x*y; y:=x;$	$x= 0 ;$ $y= 0$
3*	$a:=1.1; c:=2.2; c:=c-a; a:=a*2-3; c:=c+a/2;$	$a= -0,8;$ $c= 0,7$
4**	$a:=4; c:=5; a:= a+c/3-4; c:=(c-a)/a*c;$	$a= 1,67 ;$ $c= 0,40$