

Module: Informatique 21. Définition:

Le langage Pascal a été créé en 1969 à l'école polytechnique de ZURICH par N. WIRTH. Il a été conçu pour permettre d'enseigner la programmation comme une science. Ce langage est à la base d'une nouvelle méthode de programmation : la programmation structurée, et c'est pour cette raison que ce langage a eu un tel succès dans l'enseignement.

2. Les Types de base:

Type	Désignation	Taille	Valeurs
INTEGER	Entier(Z)	4 bytes	-2147483648..2147483647
REAL	Réel(R)	8 bytes	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$
CHAR	Caractère	1 byte	{'a'..'z','A'..'Z','0'..'9',' ','(',')','+', '-', '/', '*',}
BOOLEAN	Logique	1 byte	{True, False}

3. Les Operateursa. Operateurs arithmétiques (Opérations sur INTEGER et REAL)

Operateur	Opération	Type Opérande	Type Résultat	Exemple
+	Addition	INTEGER, REAL	INTEGER, REAL	X + Y
-	Soustraction	INTEGER, REAL	INTEGER, REAL	X - Y
*	Multiplication	INTEGER, REAL	INTEGER, REAL	X * Y
/	Division réel	INTEGER, REAL	REAL	X / Y
DIV	Division entière	INTEGER	INTEGER	X DIV Y
MOD	Modulo (reste de la division entière)	INTEGER	INTEGER	Y MOD Y
+ (unaire)	Signe positif	INTEGER, REAL	INTEGER, REAL	+ X
- (unaire)	signe négative	INTEGER, REAL	INTEGER, REAL	-X

Attention:

- Les expressions arithmétiques s'écrivent sous forme linéaire en langage pascal et en utilisant les opérateurs au dessus.
- Langage pascal évalue les expressions arithmétiques de gauche vers l'adroite en respectant les priorités comme suite:

Opérateur	Priorité	Rem
*, /, DIV, MOD	1	Haute
+, -	2	Bas

Rem on utilise les parenthèses () pour redéfinir la priorité.

Voici quelque Exemple: Soit A, B, C et D des variables de type INTEGER ou REAL

1. $\frac{A+B}{C \times D}$ s'écrit en pascal: (A + B) / (C * D)

Si on veut que pascal calcule la division entière (A + B) DIV (C * D)

2. $\frac{A+B}{C-D}$ s'écrit en pascal: $(A + B) / (C - D)$ OU $(A + B) \text{ DIV } (C - D)$
3. $\frac{A \times B}{C}$ s'écrit en pascal: $(A * B) / C$ ou $A * B / C$

Les parenthèses ne sont pas obligatoire ici (3.) mais en (2. et 1.) sont obligatoire

Exercice

1. Ecrire les expressions suivantes sous formes linéaires:

Soit A, B, C trois variables

$$-A + B \times C, A + \frac{B}{C}, \frac{A+B}{C}, \frac{A}{C} \times B, \frac{A}{C \times B}, \frac{-A+B}{C}$$

2. Soit X, Y, Z, N quatre variables, Donnez pour chaque expression le type possible de chaque opérande.

- a. $X+Y*Z$ b. $X \text{ DIV } 3$ c. $Y \text{ DIV } Z$ d. $X \text{ MOD } Y$
 e. $X \text{ DIV } (Y \text{ MOD } N)$ f. $-X+Y / N$ g. $X-Y \text{ DIV } N$

Evaluez chaque expression

Pour: $X = 2.5$ $Y = 4.5$ $Z = 2$ $N = 1.5$

Puis pour $X = 10$ $Y = 6$ $Z = 3$ $N = 4$

Solution:

1. $-A + B * C, A + B / C, (A+B)/C, A/C*B, A/(C*B), (-A+B)/C$
 2.

Expression	X	Y	Z	N	Résultat	(1)	(2)
$X+Y*Z$	INTEGER REAL	INTEGER REAL	INTEGER REAL		INTEGER REAL	11.5	28
$X \text{ DIV } 3$	INTEGER				INTEGER	Erreur	3
$Y \text{ DIV } Z$	INTEGER		INTEGER		INTEGER	Erreur	2
$X \text{ MOD } Y$	INTEGER	INTEGER			INTEGER	Erreur	1
$X \text{ DIV } (Y \text{ MOD } N)$	INTEGER	INTEGER		INTEGER	INTEGER	Erreur	5
$-X+Y / N$	INTEGER REAL	INTEGER REAL		INTEGER REAL	REAL	0.5	-8.5
$X-Y \text{ DIV } N$	INTEGER REAL	INTEGER		INTEGER	INTEGER REAL	Erreur	9

b. Operateurs logiques (Opérations Sur le type BOOLEAN)

Operateur	Opération	types Opérandes	Type de Résultat
NOT	Négation	BOOLEAN	BOOLEAN
AND	Conjonction	BOOLEAN	BOOLEAN
OR	Disjonction	BOOLEAN	BOOLEAN
XOR	Disjonction Exclusive	BOOLEAN	BOOLEAN

Soit A et B deux variables de type BOOLEAN

A	B	A and B	A or B	A xor B
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE

A	NOT A
TRUE	FALSE
FALSE	TRUE

Attention:

- Les expressions logiques s'écrivent en pascal sous forme linéaire, et en utilisant les opérateurs au dessus.
- Langage pascal évalue les expressions logiques de gauche vers la droite en respectant les priorités comme suite:

Opérateur	Priorité	Rem
NOT	1	Haute
AND	2	
OR, XOR	3	Bas

- Soit A, B, C et D variables de type BOOLEAN
Calculer le résultat de chaque expression sachant que:
A = TRUE, B = FALSE, C = TRUE, D = False
 - Not A and B
 - A and not B
 - Not (A or B)
 - Not (A and B) or C and D
 - C xor A and B
 - A and B and C and D

c. Opérateurs Relationnels.

Opérateurs	Opération	Exemple
<	Inferieur	X<Y
<=	Inferieur ou égale	X<=Y
>	Supérieur	X>Y
>=	Supérieur ou égale	X>=Y
=	Egalité	X=Y
<>	Différent de (≠)	X<>Y

Rem

- Pour X=10 et Y= 5
X>Y
X<Y

est True
est False

(X>8) and (Y<7)

est True

➤ True > False

4. Le Format d'un programme pascal:

```

Program Nom_programme;
    Partie déclarations: Déclaration des Nouveaux types.
                        Déclaration des Constantes.
                        Déclaration des Variables.
                        .....
Begin

    Partie Instructions

End.

```

Règles

- Nom_programme : - Proposer par le programmeur.
- Ne doit pas être un mot clé.
 - Suite des lettres et des chiffres qui commence par une lettre ne contient aucun caractère spécial sauf le _
 - De préférence être significatif.
 - Pas de déférence entre majuscule et minuscule.

a. Partie déclaration:

1. Déclaration des nouveaux types

Format

```
TYPE
```

```
    Nom_Type = Type de base;
```

Exemple

```
TYPE
```

```
    Naturel = 0.. 2147483647;
```

```
    Entier = Integer;
```

Explication : Maintenant en peut utiliser Naturel pour définie des variables de ce type, même chose pour le Type Entier

Règles

- Nom_Type: - Proposer par le programmeur.
- Ne doit pas être un mot clé.
 - Suite des lettres et des chiffres qui commence par une lettre ne contient aucun caractère spécial sauf le _
 - De préférence être significatif.
 - Pas de déférence entre majuscule et minuscule.

2. Déclaration des constantes:

Une Constante est une case mémoire non modifiable

Format

```
CONST
```

```
    Nom_Constante = Valeur;
```

Exemple

CONST

PI=3.14;g=10;Message='Bonjour';

Explication : Maintenant on peut utiliser Pi, g et Message comme des constantes pour calculer nos résultats

Règles

Nom_Constante: - Proposer par le programmeur.

- Ne doit pas être un mot clé.

- Suite des lettres et des chiffres qui commence par une lettre ne contient aucun caractère spécial sauf le _

- De préférence être significatif.

- Pas de différence entre majuscule et minuscule.

3. Déclaration des variables

Une Variable est une case mémoire modifiable

Format

VAR

Nom_ Variable : Nom_Type;

Nom_Type	{	Integer
		Real
		Char
		Boolean
		Nouveau Type définie par le programmeur

Exemple1

VAR

X, Y, Z : Real ; au lieu X: Real; Y: Real; Z: Real;

N, p, k : integer;

Exemple2

TYPE

Naturel = 0.. 2147483647;

Entier = Integer;

VAR

n : Naturel ;

a , b: Entier;

Trouve : boolean;

Règles

Nom_ Variable: - Proposer par le programmeur.

- Ne doit pas être un mot clé.

- Suite des lettres et des chiffres qui commence par une lettre ne contient aucun caractère spécial sauf le _

- De préférence être significatif.

- Pas de déférence entre majuscule et minuscule.

b. Partie Instructions: Contient les instructions.

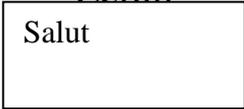
1. L'instruction: Write(LN)

Rôle: Afficher sur l'écran.

Format:

Write (LN) (Constante)	dans ce cas affiche sa valeur
	Variable);	dans ce cas affiche sa dernière valeur
	Valeur		dans ce cas affiche la valeur
	Expression		dans ce cas calcule et affiche le résultat

Exemples

Programme	Exécution	Commentaire
Program exemple1; Begin End.	Ecran 	Ce programme ne fait rien (aucun résultat)
Program exemple2; Const PI=3.14; g=10; Message='Bonjour'; Begin End.	Ecran 	Ce programme ne fait rien (La déclaration n'est pas un rôle)
Program exemple3; Const PI=3.14; g=10; Message='Bonjour'; Begin Write (Message); End.	Ecran 	Ce programme Affiche bonjour
Program exemple4_1; Begin Write ('Salut'); End.	Ecran 	Ce programme Affiche Salut
Program exemple4_2; Begin Write ('Salut'); Write ('Mohamed'); End.	Ecran 	Ce programme Affiche Salut et Mohamed
Program exemple4_3; Begin Write ('Salut','Mohamed'); End.	Meme Resultat	Même rôle

<pre>Program exemple4_5; Begin Write ('Salut',' ','Mohamed'); End.</pre>	<p>Ecran</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">Salut Mohamed</div>	Même Rôle
<pre>Program exemple4; Const PI=3.14; g=10; Begin Writeln (15); Writeln (10*15/2); Writeln (2*g*Pi); End.</pre>	<p>Ecran</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">15 75 62.8</div>	Ce programme affiche la valeur 15 puis affiche l'expression $\frac{10 \times 15}{2}$, puis l'expression $2\pi g$ pour $g=10$
<pre>Program exemple4; PI=3.14; g=10; Begin Writeln (15, 10*15/2, 2*g*Pi); End.</pre>	<p>Ecran</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">15 75 62.8</div>	Même Rôle
<p>Rem: Write: écrire → le résultat suivant sera afficher en même ligne. Writeln: écrire et retourner à la ligne → le résultat suivant sera afficher à ligne suivante.</p>		

2. L'instruction: Read(LN)

Rôle: Introduire les données depuis le clavier.

Format:

Readln (Variables);

Read

Règles

Readln(a);

- Cette instruction désigne que le programme demande une valeur pour la variable a
- A doit être déclaré à la clause VAR.
- En peut écrire Readln (a, b); au lieu Readln(a);Readln(b);

Exemple

Soit le programme suivant:

```
Program exemple1;
```

```
Var a: Real;
```

```
Begin
```

```
  Readln (A);
```

```
  Write ('Resultat=', A*A)
```

```
End.
```

Exécuter ce programme pour: 5, 10, 15
 Quel est le rôle de ce programme.

Solution:

Pour: 5

A
5

Resultat =25

Ecran

Pour: 10

A
10

Resultat =100

Ecran

Pour: 15

A
15

Resultat =225

Ecran

Ce programme calcule le carré d'un nombre réel.

3. L'instruction := (Affectation)

Rôle: affecte une valeur à une variable.

Format:

Variable := valeur ;
 Constante;
 Variable;
 Expression;

Règles

- A:=5 on lit (A reçoit 5) affecte la valeur 5 a la variable a
- la variable doit être déclarée à la clause VAR
- Variable := valeur ;
 Valeur doit être de même type que la variable
 A: Integer; et A:=2.5 faux le juste A: real et A:=2.5;
- Ou A: char; et A:= 5 Faux aussi A: char et A:='5';
- Ou A: boolean; et A:='True'; faux le juste est A:=True;
- Variable:= Constante;
 La variable doit être de même type que la valeur de la constante

CONST

PI=3.14; g=10; Message='TRUE';

```
KHATAA=false;
Var
  A: integer; b: Boolean; x: real;
Avec cette déclaration:
A:= Pi;           Faux
A:= Message;     Faux
A:= g;           juste on peut
B:=Message;     Faux
B:= KHATAA      juste;
A:= A mod 2;     juste
X:= 2*Pi*A;     juste
A:= 2*PI*5;     Faux
X:=A;           Juste
B:=A>=5;        Juste
```

Exemple

Soit le programme suivant:

```
Program exemple1;
```

```
Var a, b: Real;
```

```
Begin
```

```
  Write ('Entre une valeur :'); Readln (A);
```

```
  B := a*a;
```

```
  Write ('Resultat=', b)
```

```
End.
```

Exécuter ce programme pour: 5

Ecran

A	B
5	25

Resultat =25

Exercices 1

Exercices 2

Exercices 3

Exercices 4

Exercices 5

Exercices 6

Exercices 7

Exercices 8

Exercices 9

4. L'instruction IF Then Else

Rôle: instructions liée à une condition (la condition est expression ou variable logique)

Format:

Format	Commentaire
If condition then InstructionV;	InstructionV; Sera exécuter sauf si la condition est TRUE
If condition then Begin InstructionV1; InstructionV2; End;	InstructionV1; InstructionV2; InstructionV3; ... Seront exécuter sauf si la condition est TRUE
If condition then InstructionV Else InstructionF;	<pre> graph LR C[Condition] -- True --> IV[InstructionV] C -- Faux --> IF[InstructionF] </pre>
If condition then Begin InstructionV1; InstructionV2; End Else InstructionF;	<pre> graph LR C[Condition] -- True --> IV[InstructionV1; InstructionV2; ..] C -- Faux --> IF[InstructionF] </pre>
If condition then InstructionV Else Begin InstructionF1; InstructionF2; End;	<pre> graph LR C[Condition] -- True --> IV[InstructionV;] C -- Faux --> IF[InstructionF1; InstructionF2; ...] </pre>
If condition then Begin Instruction1; Instruction2; End Else Begin Instruction1; Instruction2; End;	<pre> graph LR C[Condition] -- True --> IV[InstructionV1; InstructionV2; ..] C -- Faux --> IF[InstructionF1; InstructionF2; ...] </pre>

Exercices 10**Exercices 11****Exercices 12****Exercices 13****Exercices 14****Exercices 15****Exercices 16****Exercices 17****Exercices 18****Exercices 19**5. L'instruction For To DoRôle: instructions itérativesFormat:

Format	Commentaire
FOR variable := valeur1 To valeur2 Do instruction;	Variable:= valeur1; Instruction; Variable:= valeur1+1; Instruction; Variable:= valeur1+2; Instruction; Variable:= valeur2; Instruction;
FOR variable := valeur1 To valeur2 Do Begin Instruction1; Instruction2; ... InstructionN; End;	Variable:= valeur1; Instruction1; Instruction2; ... InstructionN; Variable:= valeur1+1; Instruction1; Instruction2; ... InstructionN; Variable:= valeur1+2; Instruction1; Instruction2; ... InstructionN; Variable:= valeur2; Instruction1; Instruction2; ... InstructionN;

Exemple

Ce programme affiche les nombres 1 2 3 4 5 6 ...10.

Program exemple;

Var

A : integer;

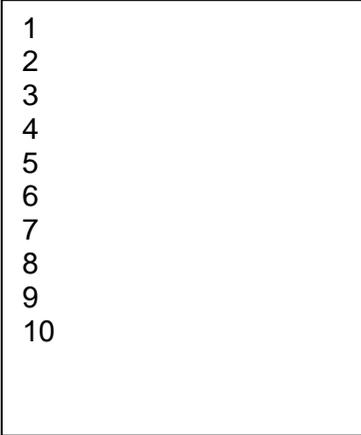
Begin

For i :=1 to 10 do writeln (a);

End.

Ecran

A
1
2
3
4
5
6
7
8
9
10

**Exercices 20****Exercices 21****Exercices 22****Exercices 23****Exercices 24****Exercices 25****Exercices 26****Exercices 27****Exercices 28****Exercices 29**

6. L'instruction WhileDoRôle: instructions itérativesFormat:

Format	Commentaire
While <u>Condition</u> do instruction;	Exécuter instruction jusqu'a <u>Condition</u> devient False
While <u>Condition</u> do Begin Instruction1; Instruction2; InstructionN; End;	Exécuter les instructions Instruction1; Instruction2; InstructionN; Jusqu'a <u>Condition</u> devient False

Condition est une expression logique ou variable de type boolean.Exemple

Ce programme affiche les nombres 1 2 3 4 5 6 ...10.

```

Program exemple;
Var
  A: integer;
Begin
  A:=1;
  While A<=10 do
  Begin
    Writeln (a);
    A:=A+1;
  End;
End.

```

A
1
2
3
4
5
6
7
8
9
10

Ecran

1
2
3
4
5
6
7
8
9
10

Exercices 30**Exercices 31****Exercices 32****Exercices 33****Exercices 34****Exercices 35**

7. L'instruction Repeat untilRôle: instructions itérativesFormat:

Format	Commentaire
Repeat Instruction1; Instruction2; InstructionN; Until <u>Condition</u> ;	Exécuter les instructions Instruction1; Instruction2; InstructionN; jusqu'a <u>Condition</u> devient True <u>Condition</u> est une expression logique.

Exemple

Ce programme affiche les nombres 1 2 3 4 5 6 ...10.

Program exemple;

Var

A: integer;

Begin

A:=1;

Repeat

Writeln (a);

A:=A+1;

Until A>10;

End.

Ecran

A
1
2
3
4
5
6
7
8
9
10

1
2
3
4
5
6
7
8
9
10

Exercices 36Exercices 37

Exercice

1. Ecrire un programme qui calcule le produit de deux nombres
2. Ecrire un programme qui calcule la somme de trois nombres
3. Ecrire un programme qui calcule la moyenne de trois notes
4. Ecrire un programme qui calcule le total à partir de la quantité et le prix
5. Reprendre l'exercice 4 et de calculer le total pour 2 articles
6. Ecrire un programme qui calcule le pourcentage d'un nombre (10%)
7. Ecrire un programme qui calcule la surface et le périmètre d'un cercle.
8. Ecrire un programme en Pascal qui permet de calculer et d'afficher la distance entre deux points dont les coordonnées sont données. Soit les points M (a, b) et N (c, d) ; la distance entre eux est donnée par la formule suivante : $d(M, N) = \sqrt{(a - c)^2 + (b - d)^2}$

Rem : \sqrt{x} en pascal s'écrit Sqrt(x)

9. Ecrire un programme en Pascal, qui convertit en Octets, Kilo binaire octets, Mega binaire octets et Giga binaire octets un nombre donnée en bits.
Sachant que 1 Octet = 8 bit ; 1 Kbi = 2^{10} Octet ; 1 Mbi = 2^{20} Octet ; 1 Gbi = 2^{30} Octet
10. Ecrire un programme qui détermine si un nombre est positif ou négative.
11. Ecrire un programme qui calcule la division de deux nombres.
12. Ecrire un programme qui introduit l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard.
Par exemple, si l'utilisateur tape 21 puis 32, le programme doit répondre :
"Dans une minute, il sera 21 heure(s) 33 minute".
NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.
13. Ecrire un programme qui détermine si un nombre est pair ou impair.
14. Ecrire un programme qui résoudre les équations de la forme $ax+b=0$.
15. Ecrire un programme qui, à partir du numéro d'un mois, permet d'afficher le nombre de jours qui lui correspond.

Exemple :

Si mois = 1, 3, 5, 7, 8, 10, 12

Alors le programme affiche : Le nombre de jours du mois ... est de 31 jours

Si mois = 4, 6, 9, 11

Alors le programme affiche : Le nombre de jours du mois ... est de 30 jours

Si mois = 2 et l'année est bissextile.

Alors le programme affiche : Le nombre de jours du mois 2 est de 29 jours

Si mois = 2 et Année et l'année n'est pas bissextiles

Alors le programme affiche : Le nombre de jours du mois 2 est de 28 jours

Rem : L'année est bissextile si elle est divisible par quatre. Toutefois, les années divisibles par 100 ne sont pas bissextiles.

16. Ecrire un programme qui calcule la racine d'un nombre avec test préalable pour savoir si le nombre est positif ou négatif (et affiche dans ce cas le message : le nombre est négatif.)
17. Ecrire un programme qui résoudre les équations de second degré.
18. Ecrire un programme qui, à partir du numéro d'un mois, permet d'afficher la saison.

Exemple :

Si mois = 6, 7, 8 Alors le programme affiche Eté

Si mois = 3, 4, 5 Alors le programme affiche Printemps

Si mois = 9, 10, 11 Alors le programme affiche Automne

Si mois = 12,1, 2 Alors le programme affiche Hiver.

19. Ecrire un programme qui, à partir de la saisie de deux réels et un opérateur affiche le résultat après exécution de l'opération choisie.

Exemple :

Si les entiers sont 14 et 2 et l'opérateur est '+' alors le résultat affiché est $14.00 + 2.00 = 16.00$

Pour les mêmes entiers et si l'opérateur est '/' alors le résultat affiché est $14.00 / 2.00 = 7.00$

20. Ecrire un programme qui affiche les diviseurs d'un Nombre N

21. Ecrire un programme qui calcule S tel que: $S = 1 + 2 + 3 + \dots + 100$

22. Ecrire un programme qui calcule S tel que: $S = 1 + 3 + 5 + \dots + N$

23. Ecrire un programme qui calcule S tel que: $S = 2 + 4 + 6 + \dots + N$

24. Ecrire un programme qui calcule S tel que: $S = 1 - 2 + 3 - 4 + \dots \pm N$ (N est strictement positif).

25. Ecrire un programme qui calcule S tel que: $S = -2 - 4 - 6 - \dots - N$.

26. Ecrire un programme qui calcule S tel que: $S = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots + \frac{1}{n^2}$

27. Ecrire un programme qui calcule S tel que: $S = \frac{1}{x} - \frac{2!}{x^2} + \frac{3!}{x^3} - \frac{4!}{x^4} + \dots + \frac{n!}{x^n}$

(N est un entier positif et X est un réel).

28. Ecrire un programme qui détermine et affiche la nième valeur U_n de la "suite de Fibonacci" définie comme suit : $U_1 = 1$ $U_2 = 1$ $U_n = U_{n-1} + U_{n-2}$ pour $n > 2$

29. Ecrire un programme en pascal qui permet de saisir un entier et afficher:

- "N est premier" si N est premier

- "N n'est pas premier" sinon

30. Ecrire un programme qui affiche les diviseurs d'un Nombre N.

31. Ecrire un programme qui calcule et affiche $S = 0.01 + 0.02 + 0.03 + 0.04 + 0.05 + \dots + 4$

32. Ecrire un programme qui détermine si un Nombre N est Premier ou non.

33. Ecrire un programme qui détermine le PGCD de deux Entiers.

34. Ecrire un programme qui détermine le PPCM de deux Entiers.

35. Ecrire un programme qui Affiche la représentation Binaire d'un Nombre décimal.

36. Ecrire un programme qui détermine le PGCD de deux Entiers.

37. Ecrire un programme qui détermine le PPCM de deux Entiers.