

Langages de programmation

1- Programme informatique

Un programme informatique est une suite d'instructions exécutable par l'ordinateur.

2- Langages de programmation

2-1- Définition

Un langage de programmation est un ensemble de caractères (alphabets, signes), un vocabulaire et un ensemble de règles syntaxiques qu'il faut respecter pour réaliser un programme.

2-2- Exemple de langages de programmation

Pascal, Langage C, Delphi, Visual C++, Java, Visual basic,....

Remarque

La construction d'un programme informatique nécessite le passage par les étapes suivantes :

Problème → **Enoncé** → **algorithme** → **Programme**

3-Langage PASCAL

Langage pascal est un Langage structuré qu'on utilise souvent pour débiter la programmation.

4- Structure d'un programme en langage Pascal

Un programme PASCAL est composé de trois parties : l'entête, partie déclarative et le corps.

```
Program nom ;                               {En-tête}
Unités ;
Déclarations des variables, constantes; {Partie déclarative}
Begin
Instructions 1 ;                             }
..... ;                                   {Corps de programme}
Instructions n ;
End.
```

A- L'en-tête du programme

Program suivi du nom du programme

Uses suivi du nom de l'unité

Une unité en PASCAL est une fonction prête à l'emploi

Exemple

```
Program somme ;
Uses Crt ; (Permet la gestion de l'écran en mode texte)
```

B- la partie déclarative

1- Les variables et constantes

Les variables sont destinées à contenir des informations modifiables

Les constantes sont destinées à recevoir des informations non modifiables

Syntaxe

Variable	Constante
Var nom : type ;	Const nom = valeur ;

Exemple

```
Var note : real ;  
Var nom : string ;  
Const Pi = 3.14 ;
```

2- Différents types de variables

Désignation	Description	Place en mémoire
Integer	entier	2 octets
Real	réel	6 octets
Char	caractère	1 octet
String	chaîne de caractères	256 octets
Boolean	valeurs logiques	1 octet

Remarque

- Toute variable utilisée dans un programme doit être déclarée
- Les noms (identificateurs) des constantes et des variables, doivent être des noms simples sans : accent, signes de ponctuation, espace et chiffre au début.
- Les identificateurs doivent impérativement être différents de mots réservés du langage Pascal (**program, begin, if, then, const, case, else, var, end**.....)

C- Le corps de programme

Cette partie contient toutes les instructions nécessaires à l'exécution du programme.

1-Instructions d'entrée/ sortie et l'affectation

Instructions	Désignations
Write	Permet d'écrire (afficher) à l'écran
Writeln	Permet d'écrire à l'écran et de faire passer le curseur à la ligne suivante.
Read	Permet de lire (entrer) les valeurs tapées au clavier et les interprètes dans le format de la variable.
Readln	Joue le même rôle de Read et faire passer le curseur à la ligne suivante.
:=	L'affectation permet d'attribuer une valeur à une variable

Exemple

```
Write('C'est un exemple') ;  
Read(Nom) ;  
P :=10/2 ;  
Writeln('le produit de a et b est : ',P) ;
```

2-Opérateur et Expression

Opérateurs arithmétiques	Opérateurs sur les chars et string	Opérateurs relationnels et logiques
+ : addition - : soustraction * : multiplication / : division Mod : Reste de la division entière Div : Division entière	+ permet de concaténer deux chaînes	> , = , < , >= , <= , <> AND OR NOT

Table de vérité des opérateurs logiques

<i>A</i>	<i>B</i>	<i>A AND B</i>	<i>A OR B</i>	<i>NOT A</i>
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Exemple

A := 3+8

B := 40 Mod 7

C := (A<0) and (b>0)

3-Structures conditionnelles

Structure conditionnelle simple
if condition then Instruction ;
Structure conditionnelle alternative
if Condition then Instruction1 else Instruction2 ;
Structure imbriquée
if condition1 then Instruction1 else if condition 2 then Instruction2 else Instruction3 ;

Structure à choix multiple	
Case Variable ou Expression of Val 1 : Instruction 1 ; Val 2 : Instruction2 ; Val n : Instruction n ; End ;	Case Variable ou Expression of Val 1 : Instruction 1 ; Val 2 : Instruction 2 ; Val n : Instruction n ; Else Autres Instructions ; End ;

4 – Fonctions

Syntaxe	Fonction
Sin(x)	sinus
Cos(x)	cosinus
Abs(x)	valeur absolue
Sqrt(x)	racine carré
Sqr(x)	carré

5-Transcription d'un algorithme

Exemple1

Algorithme	Programme
Algorithme Bonjour ; Variable A : entier ; Variable N : Chaîne de caractères ; Début Ecrire (' Entrez votre nom: ') ; Lire (N) ; Ecrire (' Entrez votre âge: ') ; Lire (A) ; Ecrire (' Bonjour ', N, ' tu as ', A, ' ans ') ; Fin.	Program Bonjour ; Uses crt ; Var A : Integer ; Var N : String ; Begin Clrscr ; Write (' Entrez votre nom: ') ; Readln (N) ; Write (' Entrez votre âge:') ; Readln (A) ; Writeln (' Bonjour ', N, ' tu as ', A, ' ans ') ; Readln ; End.

Exemple2

Algorithme	Programme
Algorithme Nature_nombre ; Variable n : Réel ; Début Ecrire ('Entrez un nombre : ') ; Lire(n) ; Si n > 0 Alors Ecrire ('Ce nombre est positif') ; Sinon Ecrire ('Ce nombre est négatif') ; Fin si Fin.	Program Nature_nombre ; Var n : Real ; Begin Writeln ('Entrez un nombre : ') ; Read (n) ; if (n > 0) then Writeln ('Ce nombre est positif') else Writeln ('Ce nombre est négatif') ; readln ; end.

Exemple3

Algorithme	Programme
Algorithme Nom_chiffre Variable n : entier ; Début Ecrire ('Entrez un chiffre entre 0 et 5 : '); Lire (n) ; Cas n vaut 0 : Ecrire (' Zéro') ; 1 : Ecrire ('Un') ; 2 : Ecrire ('Deux') ; 3 : Ecrire ('Trois') ; 4 : Ecrire ('Quatre') ; 5 : Ecrire ('Cinq') ; Sinon Ecrire (' Erreur de saisie ') ; Fin cas Fin.	Program Nom_chiffre ; Var n : integer ; Begin Write ('Entrez un chiffre entre 0 et 5 : '); Readln (n) ; Case n of 0 : Write(' Zéro') ; 1 : Write ('Un') ; 2 : Write ('Deux') ; 3 : Write ('Trois') ; 4 : Write ('Quatre') ; 5 : Write ('Cinq') ; else Write (' Erreur de saisie ') ; End ; Readln ; End.

Remarque :

- **Readln** avant le « **End.** » du programme principal, attend que la touche **Entrée** soit frappée pour que le programme se termine
- **Write** ou **writeln** (**A** : n) : affiche le contenu de **A** sur une largeur d'au moins **n** colonnes
- **Write** ou **writeln** (**A** : n :k) : affiche le contenu de **A** sur une largeur d'au moins **n** colonnes avec **k** chiffres après la virgules.
- **Clrscr** de l'unité **crt** permet d'effacer l'écran.