

# **DEUST IOSI FORMATION CONTINUE**

## **PROGRAMMATION STRUCTUREE**

### **Eléments PASCAL**



## LE LANGAGE PASCAL

### A – LES DECLARATIONS

Les variables sont utilisées pour stocker (mémoriser) une valeur réutilisable. C'est une zone mémoire qui porte un nom choisi par le programmeur. Ce nom doit commencer par une lettre avec un maximum de 255 caractères (non autorisés espace . ! @ & \$ #)

Toutes les variables doivent être préalablement déclarées avant d'être utilisées dans le programme, c'est-à-dire qu'on leur affecte un type.

Les variables sont déclarées au tout début du programme avec la syntaxe VAR :

VAR nom\_de\_la\_variable : type ;

Les principaux types de variables sont :

- <b>Shortint</b> :	entier	-128 à +128	1 octet
- <b>Integer</b> :	entier	-32768 à 32767	2 octets
- <b>Longint</b> :	entier	-2 147 483 648 à 2 147 483 647	4 octets
- <b>Real</b> :	réel	2,9E-39 à 1,7E+38	6 octets
- <b>Double</b> :	réel	5,0E-324 à 1,7E+308	8 octets
- <b>Byte</b> :	entier	de 0 à 255	1 octet
- <b>Word</b> :	entier	de 0 à 65535	2 octets
- <b>Boolean</b> :	booléen	True ou False	1 octet
- <b>String</b> :	chaîne de caractères		256 octets

## B – STRUCTURE D'UN PROGRAMME PASCAL

```

Program nom de programme ;
Uses unités utilisées ;
Const déclaration de constantes ;
Var déclaration de variables ;
BEGIN { programme principal }
    ...
    Instructions
    ...
END.
  
```

Chaque instruction se termine par un point-virgule  
 La dernière instruction du programme est un **END** suivi d'un **point**.

## C – LES INSTRUCTIONS

ALGORITHMIQUE	VISUAL BASIC
<u>Saisie de l'information</u> ENTRER (variable)	<b>Read</b> (variable); /* sans retour à la ligne */ Ou <b>Readln</b> (variable); /* avec retour à la ligne */
<u>Affectation</u> Variable ← valeur	Variable := valeur
<u>Restitution de l'information</u> AFFICHER (variable)  AFFICHER ("texte", variable)	<b>Write</b> (variable); /* sans retour à la ligne */ ou <b>Writeln</b> (variable) ; /* avec retour à la ligne */  <b>Write</b> ('texte', variable); ou <b>Writeln</b> ('texte', variable);
<u>Structure alternative</u> <i>Une seule instruction dans le SI</i> <u>SI</u> condition <u>Alors</u> Instruction1 <u>Sinon</u> Instruction2 <u>Fsi</u> <i>Plusieurs instructions dans le SI</i> <u>SI</u> condition <u>Alors</u> Instruction1 InstructionN <u>Sinon</u> Instruction2 InstructionM <u>Fsi</u>	<b>If</b> condition <b>Then</b> Instruction1 <b>Else</b> Instruction2;  <b>If</b> condition <b>Then</b> <b>Begin</b> Instruction1; InstructionN; <b>End</b> <b>Else</b> <b>Begin</b> Instruction2; InstructionM; <b>End;</b>

ALGORITHMIQUE	VISUAL BASIC
<b><u>Structure conditionnelle</u></b> <u>SI</u> condition <u>Alors</u> Instruction <u>Fsi</u>	<b>If</b> condition <b>Then</b> Instruction;  <i>Même remarque que la structure alternative lorsqu'il y a plusieurs instructions (BEGIN ...END;)</i>

## D – LES AUTRES INSTRUCTIONS

ALGORITHMIQUE	VISUAL BASIC
<b><u>Structure de choix</u></b> <u>Suivant</u> <variable ou expression> <u>Faire</u> <valeur1> : Action 1 <valeur2> : Action 2 <valeur3> : Action 3 .... <u>Sinon</u> Action N <u>Fin Suivant</u>	<b>Case</b> <variable ou expression> <b>of</b> <valeur1> : Action 1; <valeur2> : Action 2; <valeur3> : Action 3; .... <b>Else</b> : Action N; <b>End;</b>
<b><u>Structure itérative</u></b> <u>Tant Que</u> <condition> <u>Faire</u> Séquence d'actions <u>Fin Tant Que</u>	<b>While</b> <condition> <b>do</b> Instruction;  <i>Même remarque que la structure alternative lorsqu'il y a plusieurs instructions (BEGIN ...END;)</i>
<b><u>Structure répétitive</u></b> <u>Répéter</u> Séquence d'actions <u>Jusqu'à</u> <condition>	<b>Repeat</b> Séquence d'actions; <b>Until</b> <condition>;
<b><u>Structure POUR</u></b> <u>Pour</u> <identificateur> = initial à final Séquence d'actions <u>Fin Pour</u> >	<b>For</b> <identificateur> := initial <b>To</b> final <b>Do</b> Instruction;  <i>Même remarque que la structure alternative lorsqu'il y a plusieurs instructions (BEGIN ...END;)</i>

## E – LES FONCTIONS INTERNES

Une fonction interne renvoie dans leur nom un résultat qui peut être utilisé dans les applications

Les fonctions les plus utilisées sont :

- **Uppcase (caractère)** : convertit un caractère en majuscules
- **Chr (nombre)** : renvoi le caractère dont le code ASCII est précisé dans la parenthèse
- **Odd (nombre)** : retourne TRUE si le nombre est pair, False dans le cas contraire
- **Int (nombre)** : renvoie la partie entière d'un nombre sans arrondis
- **Round (nombre)** : renvoie la partie entière d'un nombre avec arrondis
- **Randomize** : initialise le générateur de nombres aléatoires avec un nombre aléatoire
- **Random(nombre)** : retourne un nombre aléatoire (entier) compris entre 0 et nombre
- **Getdate(a,m,j,jj)** : renvoie la date du système dans les variable **a** (pour l'année), **m** (pour le mois), **j** (pour le jour) et **jj** (pour le numéro du jour) déclarées en type WORD. Le numéro du jour est une valeur comprise entre 0 et 6 (la valeur 0 correspond à dimanche, etc...).  
**Pour utiliser la date, il est indispensable d'utiliser l'unité DOS (instruction Uses DOS; en début de programme)**
- **Gettime(h,m,s,c)** : renvoie l'heure du système dans les variables **h** (pour l'heure), **m** (pour les minutes), **s** pour les secondes) et **c** (pour les centièmes de seconde) déclarées en type WORD.  
**Pour utiliser l'heure, il est indispensable d'utiliser l'unité DOS (instruction Uses DOS; en début de programme)**
- **GotoXY(x,y)** : permet de positionner le curseur à la position voulue dans l'écran ou dans une fenêtre Window  
x et y sont respectivement le numéro de colonne et le numéro de ligne (axes des abscisses et des ordonnées).  
**Remarque : La résolution de l'écran texte, en TurboPascal, est de 80 colonnes par 25 lignes et de 16 couleurs.**

- **TextBackground(x)** : choix de la couleur de fond pour le texte qui sera tapé.  
x est le code de la couleur (compris entre 0 et 15) ; il est tout à fait possible de mettre le nom de la couleur à la place de x.

**Pour utiliser les couleurs appliquées au texte, il est indispensable d'utiliser l'unité GRAPH (instruction Uses graph; en début de programme)**

#### Code des couleurs

Code	Nom	Couleur
0	Black	noir
1	Blue	bleu
2	Green	vert foncé
3	Cyan	cyan foncé
4	Red	rouge
5	Magenta	mauve foncé
6	Brown	marron
7	LightGray	gris clair
8	DarkGray	gris foncé
9	LightBlue	bleu flou
10	LightGreen	vert clair
11	LightCyan	cyan clair
12	LightRed	rouge clair
13	LightMagenta	mauve clair
14	Yellow	jaune
15	White	blanc

- **TextColor(x)** : choix d'une couleur pour le texte qui sera affiché.  
ou
- **TextColor(x+blink)** : Choix d'une couleur pour le texte qui sera affiché en mode clignotant.

**Pour utiliser les couleurs appliquées au texte, il est indispensable d'utiliser l'unité GRAPH (instruction Uses graph; en début de programme)**

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.