

I. Notion de variable et constante

Définition : Donnée

- ▶ Une **donnée** est une valeur introduite par l'utilisateur pendant l'exécution du programme :
 - ▶ directement (clavier, souris)
 - ▶ ou indirectement (fichier, base de données).

Définition : Constante

- ▶ Une **constante** est une valeur fixe utilisée par le programme.
Exemple : Pi.

Définition : Variable

- ▶ Une variable représente une valeur qui peut changer en cours d'exécution. Exemples :
 - ▶ L'inconnue dans une équation

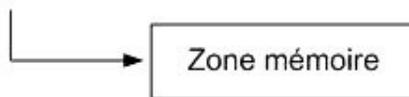
▶ 23

I. Notion de variable et constante

Représentation

- ▶ A un objet (donnée, variable ou constante) sont associées :
 - ▶ un **nom** (ou identificateur),
 - ▶ un **type** qui détermine l'ensemble des valeurs possibles,
 - ▶ une zone **mémoire** dont la taille dépend du type.

Identificateur



(la taille de la zone dépend du type)

▶ 24

I. Notion de variable et constante

Représentation

Une **variable** est un objet dont le contenu **peut être modifié** par une action. Les types les plus utilisées sont:

- ▶ Types standards
 - ▶ Entier
 - ▶ Réel
 - ▶ Booléen ou logique
 - ▶ Caractère
 - ▶ Chaîne
- ▶ Types non standard (non prédéfinis)
 - ▶ Énuméré
 - ▶ Intervalle

▶ 25

I. Notion de variable et constante

Représentation

- ▶ **ENTIER**: Pour représenter un ensemble **borné** des entiers relatifs (ex. 4, +13, -10).
- ▶ Les opérations utilisables sur les entiers sont :
 - ▶ les opérateurs arithmétiques classiques : + (addition), - (soustraction), * (produit) et / (division réelle)
 - ▶ la division entière, notée ÷ ou DIV : $n \text{ DIV } p$ donne la partie entière du quotient de la division entière de n par p
 - ▶ le modulo, (MOD) : $n \text{ MOD } p$ donne le reste de la division entière de n par p
 - ▶ les opérateurs de comparaison classiques : <, >, =, <=, >=, #

▶ 26

I. Notion de variable et constante

Représentation

► **ENTIER:**

Exemple de déclaration d'une constante de type entier et d'une variable de type entier:

Const

N = 100

Var

NB : **entier**

► 27

I. Notion de variable et constante

Représentation

- **REEL** : Pour représenter un sous ensemble **fini** des réels. Exemple : 1.5, -13.43, 7E3 (écrit en virgule flottante avec mantisse et exposant, $7 \cdot 10^3$)

Les opérations utilisables sur les réels sont :

- les opérations arithmétiques classiques : + (addition), - (soustraction), * (produit), / (division)
- les opérateurs de comparaison classiques : <, >, =, #, ...

► 28

I. Notion de variable et constante

Représentation

► **REEL** :

Exemple de déclaration d'une constante de type réel et d'une variable de type réel :

Const

```
PI = 3.14
```

Var

```
x : réel
```

► 29

I. Notion de variable et constante

Représentation

► **BOOLEEN**: est une variable de type logique (**booléenne**) et peut prendre deux valeurs **VRAI** ou **FAUX**.

Les opérations utilisables sur les booléens sont :

- les opérations logiques: **ET, OU, NON**
- les opérateurs de comparaison classiques : **<, >, =, #** (avec faux<vrai)

► 30

I. Notion de variable et constante

Représentation

- ▶ **BOOLEEN** : Tables de vérité

A	B	A ET B	A OU B
Vrai	Vrai	Vrai	Vrai
Vrai	Faux	Faux	Vrai
Faux	Vrai	Faux	Vrai
Faux	Faux	Faux	Faux

A	NON A
Vrai	Faux
Faux	Vrai

▶ 31

I. Notion de variable et constante

Représentation

- ▶ **BOOLEEN** :

Exemple de déclaration d'une constante et d'une variable de type booleen:

Const

 Trouve = vrai

Var

 Test : **booléen**

▶ 32

I. Notion de variable et constante

Représentation

- ▶ **CARACTERE:** domaine constitué des **caractères alphabétiques et numériques** et les caractères spéciaux. Une variable de ce type ne peut contenir qu'un **seul et unique caractère** (mis entre quotes).
- ▶ Les caractères sont ordonnés suivant l'ordre des codes machines considérés ASCII
- ▶ Les opérations élémentaires réalisables sont les comparaisons : <, >, =, ...

▶ 33

I. Notion de variable et constante

Représentation

- ▶ **CARACTERE :**
Exemple de déclaration d'une constante et d'une variable de type CARACTERE :

Const

```
Etoile = '*'
```

Var

```
Car: caractère
```

▶ 34

I. Notion de variable et constante

Représentation

- ▶ **CHAÎNE DE CARACTÈRE:** c'est un objet qui peut contenir plusieurs caractères de manière ordonnée.

Remarque : On note les littéraux de type **caractère** entre quote 'A'. On note les littéraux de type chaîne de caractères entre double quote : "bonjour"

Exemple

Const

```
Message = 'bonjour'
```

Var

```
Nom : chaîne[20]
```

▶ 35

I. Notion de variable et constante

Représentation

- ▶ Tous les objets manipulés par un algorithme doivent être clairement définis:

```
CONST PI=3.14
```

```
VAR a, b : ENTIER
    x, y : CARACTERE
    EXISTE : BOOLEEN
    C : CARACTERE
    NOM : CHAINE [30]
```

- ▶ Exemple : EXISTE ← VRAIE, C ← 'A',

▶ 36

I. Actions élémentaires

Affectation

- ▶ Définition
- ▶ L'affectation est l'opération qui consiste à attribuer à une variable la valeur d'une expression.
- ▶ Notation :
 - ▶ $\text{variable} \leftarrow \text{expression}$

▶ 37

I. Actions élémentaires

Affectation

- ▶ Complément
- ▶ L'affectation a donc un double rôle :
 - ▶ elle détermine la valeur de l'expression à droite de \leftarrow
 - ▶ elle range le résultat dans la variable située à gauche.
- ▶ Exemple
 - ▶ $z \leftarrow 1$ (z prend la valeur 1)
 - ▶ $\text{résultat} \leftarrow 2*3+5$ (prend la valeur du résultat de l'opération $2*3+5 = 11$)
 - ▶ $\text{solution} \leftarrow -b / a$ ($-b/a$ est évalué à l'aide des valeurs des variables a et b . Le résultat de cette évaluation est affecté à solution)
 - ▶ $nb \leftarrow nb+1$ (nb augmente de 1)

▶ 38

I. Actions élémentaires

Les opérations entrées / sorties

- ▶ **Définition**
- ▶ Permettent de récupérer une valeur venant de l'extérieur (**lecture**) ou de transmettre une valeur à l'extérieur (**écriture**).
- ▶ **Exemple :**
 - ▶ **Lire** : saisie (entrée) par clavier
 - ▶ **Ecrire** : Affichage sur écran (Moniteur)

▶ 39

I. Actions élémentaires

Les opérations entrées / sorties

- ▶ Exemple

```

VAR A, B, C : ENTIER
DEBUT
LIRE (A, B)
C ← A+B
ECRIRE (C)
ECRIRE (A+B)
FIN

```

▶ 40

I. Actions élémentaires

Structure de sélection simple (conditionnelle)

Syntaxe

```
si <condition> alors
    < séquence d'instructions >
fin de si
```

Ou

```
si <condition> alors
    < séquence d'instructions >
sinon
    < séquence d'instructions >
fin de si
```

► 41

I. Actions élémentaires

Structure de sélection simple (conditionnelle)

Exemple

```
ALGORITHME résultat
VAR note : REEL
DEBUT
LIRE (note)
SI note ≥ 10 ALORS
    ECRIRE('Admis' )
SINON
    ECRIRE('Ajourné')
FINSI
FIN
```

► 42

I. Actions élémentaires

Structure de sélection simple (conditionnelle)

Exemple

```

si a ≥ b alors
    max ← a
sinon
    max ← b
fin de si
  
```

► 43

I. Actions élémentaires

Structure de sélection simple (conditionnelle)

Exemple

```

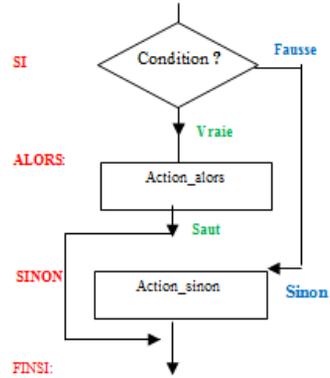
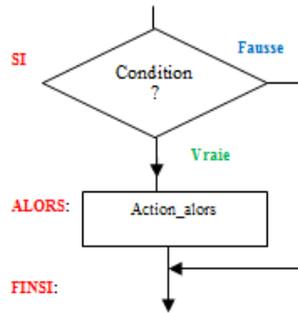
SI (A ≥ 10) ALORS
    SI ( A ≥ 12) ALORS
        ECRIRE ('Admis mention')
    SINON
        ECRIRE ('Admis passable')
    FINSI
SINON
    ECRIRE ('Ajourné')
FINSI
  
```

► 44

I. Actions élémentaires

Structure de sélection simple (conditionnelle)

Organigramme



► 45

I. Actions élémentaires

Structures répétitives

Exemple introductif : Ecrire l'algorithme permettant d'afficher la table de multiplication par 9.

ALGORITHME TABLE_MULTI

DEBUT

ECRIRE (1*9)

ECRIRE (2*9)

ECRIRE (3*9)

ECRIRE (4*9)

ECRIRE (5*9)

ECRIRE (6*9)

ECRIRE (7*9)

ECRIRE (8*9)

ECRIRE (9*9)

ECRIRE (10*9)

FIN

► 46

I. Actions élémentaires

Structures répétitives

Syntaxe

```
Tant que <condition> faire  
    <séquence d'instructions >  
Fin tant que
```

ou

```
Répéter  
<séquence d'instructions>  
jusqu'à condition
```

► 47

I. Actions élémentaires

Structures répétitives

Exemple

```
répéter  
écrire(''entrez un nombre inferieur à 10 '')  
lire (n)  
jusqu'a n < 10
```

► 48

I. Actions élémentaires

Structures répétitives

Exemple

```

i ← 1
TANTQUE ( i ≤ 5 ) FAIRE
  ECRIRE ( i * i )
  i ← i + 1
FIN TANT QUE

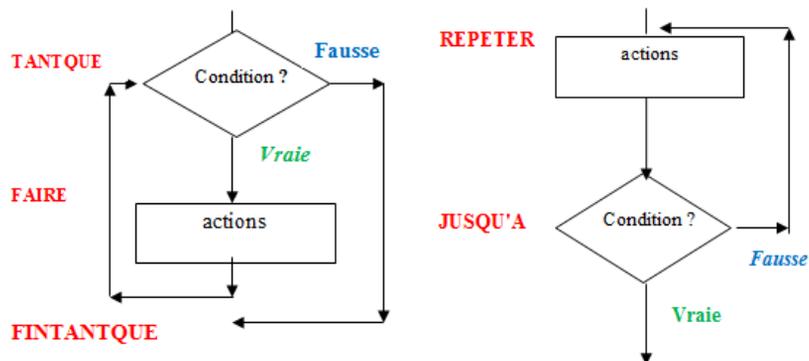
```

► 49

I. Actions élémentaires

Structures répétitives

Organigramme



► 50

I. Actions élémentaires

Structures répétitives : Structure Pour

Syntaxe

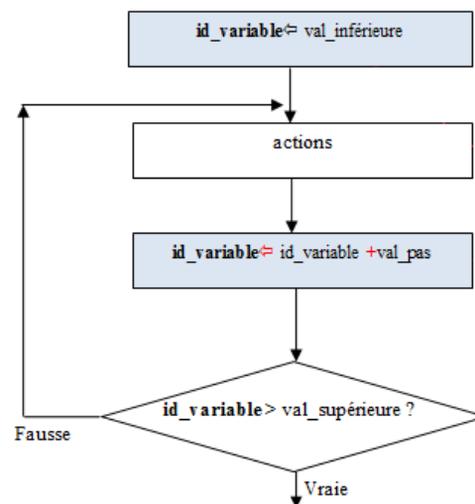
```
POUR <id_variable> DE <val_inférieure> A
<val_supérieure> [PAR PAS de <val_pas>] FAIRE
    <séquence d'instructions >
FINPOUR
```

Remarque : [] = facultatif

► 51

I. Actions élémentaires

► Structures répétitives : Structure Pour



► 52

I. Actions élémentaires

► Structures répétitives : Structure Pour

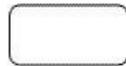
Commentez les deux boucles suivantes !

BOUCLE TANTQUE	BOUCLE REPETER
val ← -1 TANTQUE (val <=0) FAIRE Ecrire (" ENTREZ un nombre positif !!! :") LIRE(val) FINTANTQUE	REPETER Ecrire (" ENTREZ un nombre positif :!!!") LIRE(val) JUSQU'À (val >0)

► 53

I. Représentations d'un algorithme

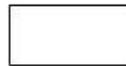
► Organigramme



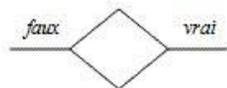
début / fin d'un algorithme



entrées / sorties (lire, afficher)



traitement (séquence d'instructions)



test



sens de lecture

► 54