

Algorithmique et Structure de données

Partie III *Structures de contrôle*



Objectifs

Objectif du chapitre

- **Savoir utiliser les opérateurs conditionnels**
- **Savoir utiliser les structures conditionnelles**
- **Savoir utiliser les structures répétitives**

III - Structures de contrôle

- Les structures de contrôle permettent de **contrôler le déroulement du programme en fonction des résultats d'un test de condition**
- **Agir** sur l'**ordre** ou la **fréquence** d'exécution des instructions

Deux grands types de structures de contrôle:

- Structures conditionnelles ou alternatives
- Structures répétitives, encore appelées boucles



III - Structures de contrôle

Structures conditionnelles

III - Structures de contrôle

•Instructions conditionnelles

- L'alternative

utilisée pour l'exécution d'un ensemble d'instructions en fonction de la valeur d'une condition (expression booléenne)

Représentation algorithmique

Si <Condition> Alors

 <TraitVrai>

[Sinon

 <TraitFaux>]

Finsi

L'expression conditionnelle est évaluée : elle est **vraie ou fausse**

III - Structures de contrôle

• Instructions conditionnelles

- L'alternative - Exemple

Variable

n : ENTIER

Debut

Ecrire("entrez un nombre")

Lire(n)

Si ($n > 0$) **Alors** // cas où la condition $n > 0$ est **vraie**

Ecrire("valeur positive")

Sinon //cas où la condition $n > 0$ est **fausse**

Ecrire ("valeur négative ou nulle")

FinSi

FIN

III - Structures de contrôle

- L'alternative

Les **traitements** après les mots Alors et Sinon peuvent être **simple** ou un **ensemble d'instructions** (bloc d'instructions)

- La condition est d'abord évaluée.
- Si vraie, **TraitVrai** exécuté puis FinSi
- Si fausse, **TraitFaux** exécuté puis FinSi

III - Structures de contrôle

.Instructions conditionnelles

- L'alternative

→ Plusieurs alternatives peuvent se succéder

Exemples :

Si *<Condition1>* Alors

< Trait1 >

Sinon Si *<Condition2>* Alors

< Trait2 >

Sinon Si *<Condition3>* Alors

< Trait3 >

Sinon

< TraitFaux >

Finsi

III - Structures de contrôle

•Instructions conditionnelles

- L'alternative - Exemple

Variable

n : ENTIER

Debut

Ecrire("entrez un nombre")

Lire(n)

Si ($n > 0$) **Alors** // cas où la condition $n > 0$ est **vraie**

Ecrire("valeur positive")

Sinon Si ($n < 0$) **Alors** //cas où la condition $n < 0$ est **vrai**

Ecrire ("valeur négative")

Sinon //cas où les deux premières conditions sont **fausses**

Ecrire ("Valeur nulle")

FinSi

FIN

III - Structures de contrôle

Programme Operation

Variable

nb1,nb2, res: entier

op : caractère

Début

Ecrire("Entrez deux nombres")

Lire(nb1, nb2)

Ecrire("entrez la première lettre de l'opération : somme ou produit")

Lire(op)

Si (op == 's') **Alors**

res \leftarrow nb1 + nb2

Sinon si (op == 'p') **Alors**

res \leftarrow nb1 * nb2

Sinon

Ecrire("Opération impossible ")

FinSi

Ecrire("Le resultat est ", res)

Fin

III - Structures de contrôle

.Instructions conditionnelles

- L'alternative

→ Plusieurs tests peuvent être imbriqués

Exemples :

Si *<Condition1>* Alors

< Trait1 >

Sinon

Si *<Condition2>* Alors

< Trait2 >

Sinon

< TraitFaux >

Finsi

Finsi

III - Structures de contrôle

.Instructions conditionnelles

- L'alternative **Selon**

- **Selon** choisit le traitement en fonction de la valeur d'une variable ou d'une expression.
- Remplace avantageusement une structure **Si**.

Syntaxe

```
Selon (expression) Faire  
    valeur1 : traitement1  
    valeur2 : traitement2  
    ...  
    valeurN : traitementN  
    Sinon traitement  
FinSelon
```

III - Structures de contrôle

.Instructions conditionnelles

- L'alternative Selon

- **expression** est un type scalaire : entier, caractère, booléen ou énuméré
- **expression** est évaluée, puis sa valeur est successivement comparée à chacune des valeurs:
 - Si correspondance, arrêt comparaisons et traitement associé exécuté.
 - Si aucune correspondance le traitement associé au Sinon, s'il existe, est exécuté

Selon (*expression*) **Faire**

valeur1 : traitement1

valeur2 : traitement2

...

valeurN : traitementN

Sinon *traitement*

FinSelon

III - Structures de contrôle

.Instructions conditionnelles

- L'alternative Selon

Exemple

```
Programme Mois_Annee
Variable mois : chaine de caracteres
Debut
  Ecrire("Donner le numéro du mois")
  Lire(mois)
  Selon (mois) Faire
    1 : Ecrire("Janvier")
    2 : Ecrire("Février")
    3 : Ecrire("Mars")
    4 : Ecrire("Avril")
    ...
    11: Ecrire("Novembre")
    12: Ecrire("Décembre")
    Sinon Ecrire("Un numéro de mois doit être compris entre 1 et 12")
  FinSelon
FIN
```

III - Structures de contrôle

.L'alternative

Condition simple: $\langle \text{expression1} \rangle \langle \text{opérateur} \rangle \langle \text{expression2} \rangle$

Condition composée: $\text{non} \langle \text{cond simple} \rangle$

$\langle \text{cond simple} \rangle \langle \text{opérateur logique} \rangle \langle \text{cond simple} \rangle$

Opérateurs : $=, \neq, <, >, \leq, \geq$ Opérateurs logiques : *non, et, ou*

Exemples :

Condition simple :

Si $\text{type} = 1$ **Alors**

$\text{TVA} \leftarrow 5,5$

Sinon

$\text{TVA} \leftarrow 19,6$

Finsi

Condition composée :

Si $\text{lettre} = 'a'$ ou $\text{lettre} = 'e'$ ou $\text{lettre} = 'i'$ ou
 $\text{lettre} = 'o'$ ou $\text{lettre} = 'u'$ ou $\text{lettre} = 'y'$ **Alors**

Ecrire "voyelle"

Sinon

Ecrire "consonne"

Finsi



III - Structures de contrôle

Structures répétitives

III - Structures de contrôle

- **Instructions répétitives**

- Appelées boucles permettent de répéter un traitement (c'est à dire une instruction simple ou composée) autant de fois qu'il est nécessaire

- soit un nombre déterminé de fois
 - soit tant qu'une condition est vraie

III - Structures de contrôle

•Instructions répétitives

- L'itérative Tant que.... Faire. / Faire Tantque

→utilisée pour réaliser des boucles c'est à dire réaliser un même traitement plusieurs fois sans le réécrire

Représentation algorithmique

Tant que <Cond> *Faire*

<Traitement>

Fin TQ

→L'expression conditionnelle va être évaluée et tant qu'elle prend la valeur vrai, le traitement est effectué, lorsqu'elle devient fausse la boucle est terminée.

Faire

<Traitement>

Tant que <Cond> *Fin TQ*

→Dans le cas du FaireTantQue le traitement est exécuté avant l'évaluation de la condition, il sera toujours exécuté au moins une fois.

III - Structures de contrôle

•L'itérative

Exemple :

Programme Moyenne

Var

note, total, moyenne : REEL

nbnotes : ENTIER

Début

total $\leftarrow 0$

nbnotes $\leftarrow 0$

Lire (note)

TantQue *note* $\neq -1$ *Faire*

total $\leftarrow total + note$

nbnotes $\leftarrow nbnotes + 1$

Lire (note)

FinTQ

moyenne $\leftarrow total / nbnotes$

Ecrire("La moyenne vaut", moyenne)

Fin



III - Structures de contrôle

•La boucle Pour

La boucle Pour permet de répéter une instruction un nombre connu de fois. Elle a le formalisme suivant:

Pour <initialisation><condition><incrémentations> **Faire**

<instructions>

FinPour

III - Structures de contrôle

•L'itérative

Représentation algorithmique

***Pour** <initialisation> <condition> <incrémentations> **Faire**
<instructions>*

FinPour

***Pour** <initialisation> <condition> <décrémentations> **Faire**
<instructions>*

FinPour

→ **Initialiser** une variable c'est lui affecter une valeur pour la première fois.

→ **Incrémenter** (décrémenter) une variable c'est modifier sa valeur par ajout (par retrait).

→ **Remarque**: attention à la condition et à l'incrémentations (décrémentations), la condition doit passer à faux à un moment ou un autre sinon on aura une boucle infinie.



III - Structures de contrôle

•L'itérative Pour

Exemple

Pour $x \leftarrow 0$ à 20 par pas de 2 Faire

Ecrire(x)

FinPour

Pour $x \leftarrow 20$ à 0 par pas de -2 Faire

Ecrire(x)

FinPour



III - Structures de contrôle

• L'itérative Pour comparaison avec Tant que

Exemple

Pour $x \leftarrow 0$ à 20 **par pas de 2** **Faire**

Ecrire(x)

FinPour

$x \leftarrow 1$

Tant que ($x \leq 20$) **Faire**

Ecrire(x)

$x \leftarrow x+1$ // incrémentation explicite

FinTQ