

Lycée Secondaire Raccada



Chapitre2

Structures de contrôle

Lecon2

Les Structures de contrôle itératives

4^{ème} Sc

Enseignant: Sami MEJRI

2015-2016

Introduction

Le traitement conditionnel, combinant test et séquence, permet d'exécuter chaque action 0 ou 1 fois. Nous aurons parfois besoin d'exécuter une séquence d'actions, plusieurs fois. Le nombre de fois peut être quelconque. C'est l'objet des structures répétitives ou itératives ou boucles.

1- la structure itérative complète (La boucle Pour)

Syntaxe

Analyse et Algorithmme	Pascal
[Init] Pour c De Vi À Vf faire Inst1 Inst2 Inst n FinPour	[Init] ; For c := Vi To Vf Do Begin Inst1 ; Inst2 ; ; Inst n ; End ;

Remarques:

- La boucle POUR est considérée comme une structure répétitive complète car le nombre d'itérations est connu au préalable
- C: compteur (doit être de type scalaire)
- Vi et Vf: la valeur initiale et valeur final
- La partie INIT contient les éventuelles initialisations des variables qui seront mises à jour au niveau du traitement répétitif.
- L'avancement du compteur est automatique.
Il se fait par un pas = 1 (Pour Compteur De Vi À Vf faire) ; avec $V_i \leq V_f$
Il se fait par un pas = -1 (Pour Compteur De Vf A Vi faire) ; avec $V_i \leq V_f$
- Lorsque le pas = -1; la notation en Turbo Pascal change :
For Compteur := Vf Downto Vi Do
- Lorsque le traitement répétitif est composé de plus qu'une instruction, les expressions Begin et End sont nécessaires.

La notion de trace d'exécution

La notion de trace d'exécution permet de bien comprendre la notion de boucle (itérations).

Exemple

Prenons la structure
Pour suivante
S ← 0
Pour i De 1 A 5 Répéter
S ← S + 2 * i
FinPour

La trace d'exécution sera la suivante :

Etat initial: S = 0

N° de l'itération	1	2	3	4	5
Valeur de i	1	2	3	4	5
Valeur de S	2	6	12	20	30

Application

Soit à afficher la moyenne arithmétique **ma** d'une classe de **n** élèves dont les moyennes **me** sont saisies au clavier.

Analyse

Résultat = Ecrire (Moyenne)

Moyenne \leftarrow Total / n

Total = [Total \leftarrow 0]

Pour i de 1 à n **répéter**

Ecrire (" Introduire une moyenne ")

ME = Donnée (" ME",i, "=")

Total \leftarrow Total + ME

FinPour

n = Donnée("Donner le nombre d'élève")

TDO

Objet	Type/Nature	Rôle
ME	Réel	Saisie de la moyenne d'un élève
n	Entier	Saisie du nombre des élèves
i	Entier	Compteur
Total	Réel	Calcul du total des moyennes
Moyenne	Réel	Calcul de la moyenne arithmétique

Algorithme

Début Calcul_moy

Ecrire (" Introduire le nombre des élèves : "), Lire (n)

$T \leftarrow 0$

Pour i de 1 à n répéter

Ecrire (" Introduire une moyenne : ") Lire (ME)

$T \leftarrow T + ME$

FinPour

$M \leftarrow T / n$

Ecrire (M)

Fin Calcul_moy

•Traduction en Pascal

```
Program Calcul_moyenne ;
Uses wincrt ;
    Var    me,Total, Moyenne :real ;
           n,i:integer;

Begin
Writeln('introduire le nombre d"eleves : ');
Readln(n) ;
Total :=0 ;
For i := 1to n do
begin
Writeln('introduire une moyenne');
Readln (me);
Total :=Total+me ;
End ;
Moyenne :=Total/n ;
Writeln('la moyenne de la classe est : ',Moyenne) ;
End.
```

II- La structure itérative a condition d'arrêt (La boucle répéter)

Syntaxe

Analyse et Algorithmme	Pascal
<pre>[Init] Répéter Inst1 Inst2 Instn Jusqu'à condition</pre>	<pre>[Init] ; Repeat Inst1 ; Inst2 ; ; Instn ; Until condition ;</pre>

Remarques:

- ☀ S'il y a un éventuel compteur, il faut l'initialiser avant la boucle ; de même on doit assurer son avancement au sein de la boucle.
- ☀ Le traitement répétitif de la boucle Répéter peut s'exécuter 1 ou n fois ($n \geq 2$).
- ☀ La condition à vérifier à chaque fois est considérée comme une condition de sortie car elle nous permet de quitter la boucle.
- ☀ Même si le traitement répétitif est composé de plusieurs instructions, on n'a jamais besoin des expressions Begin et End.
- ☀ La boucle répéter est utilisée entre autres dans le contrôle des données saisies

III- La structure itérative a condition d'arrêt (La boucle Tant que)

Syntaxe

Analyse et Algorithmme	Pascal
Init Tant que condition Faire	Init ; While condition Do
Inst1 Inst2 Instn	Begin Inst1 ; Inst2 ; ; Instn ;
Fin Tant que	End ;

Remarques:

- ✿ S'il y a un éventuel compteur, il faut l'initialiser avant la boucle; de même on doit assurer son avancement au sein de la boucle.
- ✿ Le traitement répétitif de la boucle Tant que peut s'exécuter 0 ou n fois ($n > 1$). 0 fois, lorsque dès le début, la condition n'est pas vérifiée.
- ✿ La condition à vérifier à chaque fois est considérée comme une condition d'entrée car elle nous permet d'accéder au corps de la boucle.
- ✿ Si le traitement répétitif est composé de plusieurs instructions, on doit utiliser les expressions Begin et End.

Application

Soit à saisir une suite de réels et d'afficher la partie entière de chaque nombre.

Analyse

Résultat = Affichage

Affichage = [] Tant Que Majus (réponse)= "O" Faire

 Ecrire("Introduire un réel :")

 Elément = Donnée

 Ecrire("La partie entière de",élément, "est: ",INT(élément))

 Répéter

 Ecrire("Y a t il un autre réel à saisir ? (O/N) ")

 Réponse = Donnée

 Jusqu'à (Majus (réponse) Dans ["O", "N"]

 Fin Tant Que

Réponse = [] Répéter

 Ecrire("Y a t il un autre réel à saisir ? (O/N) ")

 Réponse = Donnée

 Jusqu'à (Majus(réponse) Dans ["O","N"])

TDO

Objet	Code	Type/Nature	Rôle
Elément	E	Réel	Saisie d'un réel
Réponse	R	Caractère	Saisie de la réponse (O pour oui et N pour non)

Algorithmme

0) Début Composante_entière

1) Répéter

Ecrire("Y a t il un autre réel à saisir ? (O/N) ")

Lire (R)

Jusqu'à (Majus(R) Dans ["O","N"])

2) Tant Que Majus (R) = "O" Faire

Ecrire("Introduire un réel :") , LIRE (E)

Ecrire("La partie entière de", E , "est: ",INT(E))

Répéter

Ecrire("Y a t il un autre réel à saisir ? (O/N) ")

Lire (R)

Jusqu'à (Majus (R) Dans ["O", "N"])

Fin Tant Que

3) Fin Composante_entière

**Merci pour votre
attention**

Chapitre 3-

Les sous programmes