

# Cours: Programmation Temps Réel et Concurrente

Filière: MP-GL2

## Chapitre 2: Les instructions de contrôle

Réalisé par:

Dr. Sakka Rouis Taoufik

1

### Chapitre 2: Les instructions de contrôle

#### I. Introduction

Les instructions de contrôle servent à contrôler le déroulement de l'enchaînement des instructions à l'intérieur d'un programme.

Ces instructions peuvent être des instructions:

- ❖ **conditionnelles** : permettent de réaliser des tests, et suivant le résultat de ces tests, d'exécuter des parties de code différentes.
  - ➔ Ada distingue deux types de structures conditionnelles:
    - les structures conditionnelles simples **if** ; **if else** et **if elsif**
    - la structure à choix multiples **case**.
- ❖ **itératives**: sont commandées par les 4 types de boucles : **loop**, **while**, **for** et **goro**

2

## II. Les structures conditionnelles en Ada

### Exemple 1:

```
if Reponse = 'o'
  then Put("Vous avez bien de la chance.");
else Put ("Ha ... Dommage. ");
end if ;
```

### Exemple 2: utilisation des if imbriquées

```
if Reponse = 'o'
  then Put ("Vous avez bien de la chance");
else
  if Reponse = 'n'
    then Put ("Ah... Dommage.");
    else Put ("C'est pas une réponse ça !");
  end if ;
end if ;
```

3

## II. Les structures conditionnelles en Ada

### Exemple 3: utilisation de elsif

```
if Reponse = 'o'
  then Put("Vous avez bien de la chance");
elsif Reponse = 'n'
  then Put("Ah... Dommage.");
elsif Reponse = 'p'
  then Put("Reponses normandes non valides");
  else Put("C'est pas une réponse ça !");
end if ;
```

4

## II. Les structures conditionnelles en Ada

### Exemple 3: utilisation de elsif

```
if Reponse = 'o'
  then Put("Vous avez bien de la chance");
elsif Reponse = 'n'
  then Put("Ah... Dommage.");
elsif Reponse = 'p'
  then Put("Reponses normandes non valides");
  else Put("C'est pas une réponse ça !");
end if ;
```

## II. Les structures conditionnelles en Ada

### Exemple 4: utilisation du case

```
case Reponse is
  when 'o' => Put("Vous avez bien de la chance.") ;
  when 'n' => Put("Ah... dommage. ") ;
  when 'p' => Put("Reponses normandes non valides") ;
  when 'f' => Put("J'aurais pas du apprendre l'allemand...") ;
  when others => Put("C'est pas une reponse.") ;
end case ;
```

### III. Les structures itératives en Ada

#### Syntaxe

```
loop      --début des instructions qu'il faut répéter
          -- traitement
end loop;  --indique la fin des instruction à répéter
```

#### Exemple 1:

```
loop
  exit when Compteur = Nb ;
  Put('#') ;
  Compteur := Compteur +1 ;
end loop ;
```

### III. Les structures itératives en Ada

#### Exemple 2: Nommer une boucle

```
Ma_Boucle : loop
  exit Ma_Boucle when Nb = 0 ;
  Nb := Nb - 1 ;
  Put('#') ;
end loop Ma_Boucle ;
```

### III. Les structures itératives en Ada

#### Exemple 3: utilisation de la boucle while

```
while Compteur /= Nb loop
  Compteur := Compteur +1 ;
  Put('#') ;
end loop ;
```

#### Exemple 4: utilisation de la boucle for

for i in 1.. Nb loop	for i in reverse 1..nb loop
...	...
end loop ;	end loop ;

**Attention**, si Nb est plus petit que 1, la boucle for ne s'exécutera toujours pas !

### III. Les structures itératives en Ada

#### Exemple 5: utilisation du goto

```
<<debut>>    --on place une balise appelée "debut"
...
...
goto debut ;
```

#### Exemple 6: utilisation du goto

```
<<debut>>
if Compteur <= Nb
  then Put('#') ;
       Compteur := Compteur + 1 ;
       goto debut ;
end if ;
```

### III. Les structures itératives en Ada

#### Exemple 5: utilisation du goto

```
<<debut>>    --on place une balise appelée "debut"  
...  
...  
goto debut ;
```

#### Exemple 6: utilisation du goto

```
<<debut>>  
if Compteur <= Nb  
  then Put('#') ;  
    Compteur := Compteur + 1 ;  
    goto debut ;  
end if ;
```

11

### IV. Exercices d'application

#### Exercice 1 :

Un nombre est dit palindrome s'il est écrit de la même manière de gauche à droite ou de droite à gauche.

**Exemples :** 101 ; 22 ; 3663 ; 10801, etc.

Écrire un programme Ada permettant de déterminer et d'afficher tous les nombres palindromes compris dans l'intervalle [100..9999].

#### Exercice 2 :

Réaliser en Ada un algorithme qui affiche la suite de tous les nombres parfaits inférieurs ou égaux à un nombre naturel non nul donné noté n. Un nombre est dit parfait s'il est égal à la somme de ses diviseurs autre que lui-même.

**Exemple:**  $28 = 1 + 2 + 4 + 7 + 14$

Voici la liste des nombres parfaits inférieurs à 10000 : 6, 28, 496, 8128.

12

### IV. Exercices d'application

#### **Exercice 3 :**

Les nombres de Fibonacci sont donnés par la récurrence :

$$F_n = F_{n-2} + F_{n-1} \text{ avec } F_0 = 1 \text{ et } F_1 = 1.$$

Écrire un programme A qui affiche les 20 premiers nombres de Fibonacci.

#### **Exercice 4 :**

Deux entiers N1 et N2 sont dits frères si chaque chiffre de N1 apparaît au moins une fois dans N2 et inversement.

#### **Exemples :**

Si N1 = 1164 et N2 = 614 alors le programme affichera :

N1 et N2 sont frères

Si N1 = 405 et N2 = 554 alors le programme affichera :

N1 et N2 ne sont pas frères

Écrire un programme Ada qui saisit deux entiers N1 et N2, vérifie et affiche s'ils sont frères ou non.