

COURS ALGORITHMIQUE ET PROGRAMMATION INFORMATIQUE

GRETA CFA PROVENCE – CICALA Lionel - 2020

SELECTION CHOIX MULTIPLES

« SELON » (1)

selon <identificateur>

(liste de) valeur(s) : instructions

(liste de) valeur(s) : instructions

...

[**autres**: instructions]

S'il y a plus de deux choix possibles, l'instruction **selon** permet une facilité d'écriture

SELECTION CHOIX MULTIPLES

« SELON » (2)

selon abréviation

"M" : afficher (" Monsieur ")

"Mme" : afficher (" Madame ")

"Mlle" : afficher (" Mademoiselle ")

autres : afficher(" Monsieur, Madame ")

Équivalent avec instruction Conditionnelle

si abréviation = "M "

alors afficher("Monsieur")

sinon **si** abréviation = « Mlle »

alors afficher("Mademoiselle")

sinon **si** abréviation = "Mme"

alors afficher("Madame")

sinon afficher("Monsieur, Madame ")

fsi

fsi

fsi

SELECTION CHOIX MULTIPLES

« SELON » (3) AVEC INVERSION DES TESTS

selon abréviation

"M" : afficher (" Monsieur ")

"Mme" : afficher (" Madame ")

"Mlle" : afficher (" Mademoiselle ")

autres : afficher(" Monsieur, Madame ")

Équivalent avec instruction Conditionnelle

si abréviation = " Mme "

alors afficher(" Madame ")

sinon **si** abréviation = « Mlle »

alors afficher("Mademoiselle")

sinon **si** abréviation = "M"

alors afficher("Monsieur")

sinon afficher("Monsieur, Madame ")

fsi

fsi

fsi

SELECTION CHOIX MULTIPLES (4)

AVEC SI ... ALORS ... FSI SÉQUENTIELS

selon abréviation

"M" : afficher (" Monsieur ")

"Mme" : afficher (" Madame ")

"Mlle" : afficher (" Mademoiselle ")

autres : afficher(" Monsieur, Madame ")

Équivalent avec instruction Conditionnelle

si abréviation = " Mme "

alors afficher(" Madame ")

fsi

si abréviation = « Mlle »

alors afficher("Mademoiselle")

fsi

si abréviation = "M"

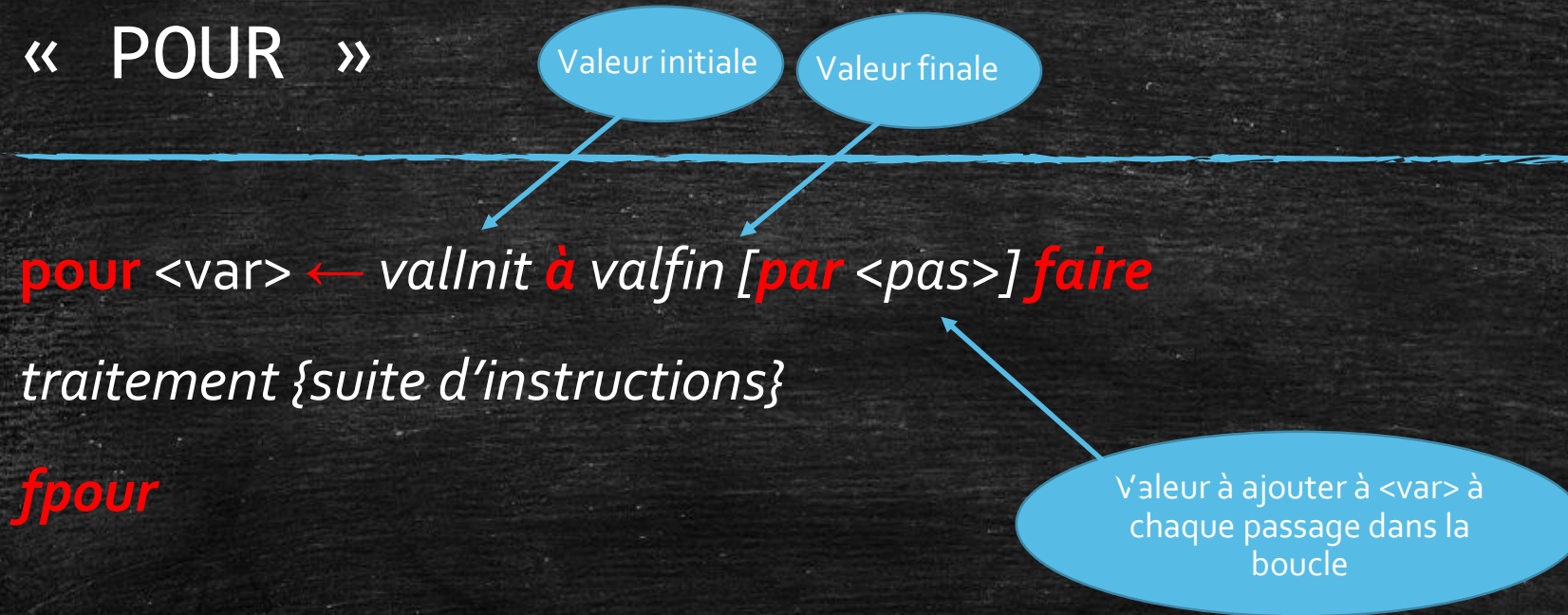
alors afficher("Monsieur")

sinon afficher("Monsieur,Madame ")

fsi

LES BOUCLES

« POUR »



Fonction: répéter une suite d'instructions un certain nombre de fois

Pour utilisée **quand le nombre d'itération** est connu

SÉMANTIQUE BOUCLE

« POUR »

- l'instruction **pour**:
 - initialise une variable de boucle (le compteur)
 - incrémente cette variable de la valeur de « pas »
 - vérifie que cette variable ne dépasse pas la borne supérieure
- Attention : •
 - Le traitement ne doit pas modifier la variable de boucle

~~Pour~~ cpt \leftarrow 1 à MAX ~~faire~~

~~si (...) alors~~

~~cpt \leftarrow MAX~~

~~fpour~~

INTERDIT !

RÉPÉTITION D'UN TRAITEMENT

BOUCLE « POUR »

Exemple

Algorithme FaitLeTotal

{Cet algorithme fait la somme des nbVal données qu'il saisit}

variables nbVal, cpt : **entiers**
 valeur, totalValeurs: **réels**

début

 {initialisation du traitement}

afficher("Combien de valeurs voulez-vous saisir ?")

saisir(nbVal)

 {initialisation du total à 0 avant cumul}

 totalValeurs←0

 {traitement qui se répète nbVal fois}

pour cpt ←1 **à** nbVal **faire**

afficher("Donnez une valeur :")

saisir(valeur)

 totalValeurs←totalValeurs+ valeur {cumul}

fpour

 {édition des résultats}

afficher("Le total des ", nbVal, "valeurs est ", totalValeurs)

fin

LES BOUCLES

« TANT QUE ... FAIRE »

amorçage

Initialisation de la (des)
variable(s) de condition

tant que <expression logique (vraie)> **faire**

traitement

Suite d'instructions à
exécuter si condition vraie

relance

Ré-affectation de la (des)
variable(s) de condition

ftq

- **Fonction:** répéter une suite d'instructions un certain nombre de fois

LES BOUCLES

« TANT QUE ... FAIRE »

- Structure itérative "universelle"
 - n'importe quel contrôle d'itération peut se traduire par le « **tant que** »
- Structure itérative irremplaçable dès que la **condition d'itération** devient **complexe**

RÉPÉTITION D'UN TRAITEMENT À NOMBRE ITÉRATIONS INCONNU « TANT QUE ... FAIRE »

- Exemple :

Algorithme FaitLeTotal

{Cet algorithme fait la somme des nbVal données qu'il saisit, arrêt à la lecture de -1 }

constante (STOP : entier) ← -1

variables val, totalValeurs: entiers

début

totalValeurs ← 0

afficher("Donnez une valeur,", STOP, " pour finir.") {amorçage}

saisir(val)

tant que val ≠ STOP **faire**

totalValeurs ← totalValeurs + val {traitement}

Afficher ("Donnez une autre valeur,", STOP, " pour finir.")

Saisir (val) {relance}

ftq

afficher("La somme des valeurs saisies est", totalValeurs)

fin

LES BOUCLES

« TANT QUE ... FAIRE »

- **Exemple :**

- Saisir des valeurs, les traiter, et s'arrêter à la saisie de la valeur d'arrêt -1 ou après avoir saisi 5 données.

Constantes (STOP : entier) $\leftarrow -1$

 (MAX : entier) $\leftarrow 5$

Variables nbVal, val : entiers

Début

nbVal $\leftarrow 0$ {**compte les saisies traitées**}

saisir(val) {saisie de la 1ère donnée}

tant que val \neq STOP **et** nbVal < MAX **faire**

 nbVal \leftarrow nbVal + 1 {traitement de la valeur saisie}

saisir(val) {relance}

Ftq

afficher(val, nbVal) {valeurs en sortie de boucle}

- **Remarque :** La valeur d'arrêt n'est jamais traitée (et donc, jamais comptabilisée)

LES BOUCLES

« TANT QUE ... FAIRE »

- **Interpréter l'arrêt des itérations**

.....

nbVal ← 0 {compte les saisies traitées}

saisir(val) {saisie de la 1ère donnée}

tant que val ≠ STOP **et** nbVal < MAX **faire**

nbVal ← nbVal + 1...{traitement de la valeur saisie}

saisir(val) {relance}

Ftq

si val = STOP

alors {la dernière valeur testée était la valeur d'arrêt}

afficher(«Sortie de boucle car saisie de la valeur d'arrêt »)

{toutes les données significatives ont été traitées.}

sinon {il y avait plus de 5 valeurs à tester}

afficher(«Sortie de boucle car nombre maximum de valeurs à traiter atteint »)

{des données significatives n'ont pas pu être traitées.}

fsi

COMPARAISON BOUCLES

« POUR » ET « TANT QUE » (1)

pour cpt \leftarrow 1 **à** nbVal **faire**

afficher("Donnez une valeur :")

saisir (valeur)

 totalValeurs \leftarrow totalValeurs + valeur {cumul}

fpour

- Est équivalent à :

cpt \leftarrow 0

tant que cpt < nbVal **faire**

afficher ("Donnez une valeur :")

saisir (valeur)

 totalValeurs \leftarrow totalValeurs + valeur {cumul}

 cpt \leftarrow cpt + 1 {compte le nombre de valeurs traitées}

ftq

COMPARAISON BOUCLES

« POUR » ET « TANT QUE » (2)

- Implicitement, l'instruction **pour**:
 - initialise un compteur
 - incrémente le compteur à chaque pas
 - vérifie que le compteur ne dépasse pas la borne supérieure
- Explicitement, l'instruction **tant que** doit
 - initialiser un compteur {amorçage}
 - incrémenter le compteur à chaque pas {relance}
 - vérifier que le compteur ne dépasse pas la borne supérieure {test de boucle}

QUAND CHOISIR

« POUR » OU « TANT QUE » ?

- Nombre d'itération connu à l'avance : **POUR**
 - Parcours de tableaux
 - Test sur un nombre donné de valeurs
- Boucle s'arrête sur événement particulier : **TANT QUE**
 - Itération avec arrêt décidé par saisie utilisateur

BOUCLE « RÉPÉTER ...TANT QUE »

Répéter

(ré)affectation de la (des) variable(s) de condition traitement

Tant que <expression logique(vraie)>

- **Fonction** : exécuter une suite d'instructions **au moins une fois** et la répéter tant qu'une condition est remplie
- **Remarque** : le traitement dans l'exemple précédent se limite à la ré-affectation de la variable de condition (**saisir** (valeur))

ON N'A PAS FINI D'ITÉRER !

- **Boucle « répéter ... tant que »** : exemple

Algorithme Essai

{Cet algorithme a besoin d'une valeur positive paire}

Variables valeur : entier

Début

Répéter

afficher("Donnez une valeur positive non nulle : ")

saisir(valeur)

tant que valeur ≤ 0

afficher("La valeur positive non nulle que vous avez saisie est ")

afficher(valeur)...{traitement de la valeur saisie}

fin

COMPARAISON «RÉPÉTER» ET «TANT QUE»

Répéter

afficher ("Donnez une valeur positive paire :")

saisir (valeur)

tant que (valeur < 0 ou (valeur % 2) ≠ 0)

- Équivaut à :

Afficher ("Donnez une valeur positive paire :")

Saisir (valeur)

tant que (valeur < 0 ou (valeur % 2) ≠ 0) **faire**

Afficher ("Donnez une valeur positive paire:")

Saisir (valeur)

ftq

COMPARAISON «RÉPÉTER» ET «TANT QUE»

- boucle **tant que**
 - condition vérifiée **avant** chaque exécution du traitement
 - le traitement peut donc ne pas être exécuté
 - de plus : la condition porte surtout sur la saisie de nouvelles variables (relance)
- • boucle **répéter ... tant que**
 - condition vérifiée **après** chaque exécution du traitement => le traitement est **exécuté au moins une fois**
 - de plus: la condition porte surtout sur le résultat du traitement
- • **Remarque**: la boucle répéter est typique pour les saisies avec vérification

DE L'ÉNONCÉ À LA BOUCLE

- *Saisir des données et s'arrêter dès que leur somme dépasse 500*

DE L'ÉNONCÉ À LA BOUCLE

saisir (val)

somme \leftarrow val

tant que somme ≤ 500 **faire**

 Saisir (val)

 somme \leftarrow somme + val

ftq

DE L'ÉNONCÉ À LA BOUCLE

somme \leftarrow 0

répéter

Saisir (val)

somme \leftarrow somme + val

tant que somme \leq 500

