

Les structures répétitives

Institut Supérieur Informatique

Prof. : Ibrahima Sy

Prof : Mr SY

Concepts

1. **Tant que ... Faire**
2. La boucle **Pour**
3. La boucle **Faire ... Tant que**

Structures répétitives

Les structures répétitives aussi 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.

Il existe trois grands types principaux de structures répétitives:

- la structure **Tant que ... Faire**, qui permet d'effectuer une instruction tant qu'une condition est satisfaite
- la structure **Pour** qui permet de répéter une instruction un certain nombre de fois
- la structure **Tant ... Faire Jusqu'à**, qui comme son nom l'indique, permet de répéter une instruction jusqu'à ce qu'une condition soit satisfaite.

Seule la boucle **Tant que** est fondamentale. Avec cette boucle, on peut réaliser toutes les autres boucles alors que l'inverse n'est pas vrai. La boucle **Pour** est très utilisée aussi car elle permet de simplifier la boucle **Tantque** lorsque le nombre de tour de boucle est connu d'avance.

1 La boucle Tant que

La boucle **Tant que** permet de répéter un traitement tant qu'une condition est vraie.

La syntaxe générale de cette structure est la suivante:

Algorithm 1 SYNTAXE

- 1: **tant que** *condition* **faire**
 - 2: *instructions*
 - 3: **fin tant que**
-

Si, dès le début, la condition est fausse, alors le traitement ne sera pas exécuté.

Exemple:

Donner algorithme qui calcule le cube des nombres non nuls saisis par l'utilisateur. Tant que le nombre saisi par l'utilisateur n'est pas nul, on affiche son cube et on recommence.

2 La boucle Pour

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

Algorithm 2 SYNTAXE

- 1: **pour** compteur = valeur_initiale à valeur_finale **par** pas de increment **faire**
 instruction1
 - 2: **fin pour**
-

La variable compteur est de type entier. Elle est initialisée à la valeur valeur_initiale. Le compteur augmente (implicitement) de la valeur increment à chaque exécution du traitement. Lorsque la variable compteur vaut la valeur valeur_finale, le traitement est exécuté une dernière fois puis le programme sort de la boucle. Par défaut, l'incrément est de 1

3 La boucle Faire ... Tant que

Cette boucle sert à répéter une instruction tant qu'une condition est vraie. Son formalisme est le suivant:

Algorithm 3 SYNTAXE

- 1: **faire**
 - 2: instructions
 - 3: **tant que** (condition)
-

Le traitement est exécuté une première fois, puis la condition est vérifiée. Si elle est vraie, on retourne au début de la boucle et le traitement est répété. Si la condition est fausse, on sort de la boucle. A chaque fois que le traitement est exécuté, la condition est de nouveau vérifiée à la fin.

4 Instructions de rupture de séquence

4.1 L’instruction Arrêter

L’instruction **Arrêter** peut être employée à l’intérieur de n’importe quelle boucle. Elle permet d’interrompre le déroulement de la boucle, et passe à la première instruction qui suit la boucle. En cas de boucles imbriquées, Arrêter fait sortir de la boucle la plus interne.

4.2 L’instruction Continuer

L’instruction **Continuer** permet de passer directement à l’étape suivante d’une boucle sans exécuter les autres instructions de la boucle qui la suivent.