

Chapitre 3 : Les itérations

Juan Carlos QUEZADA

INSA Strasbourg

Exécution itérative

Répéter une série d'instruction tant qu'un test est vrai

Syntaxe Tant que

```
tant que <expression booléenne> faire  
  <instructions>  
fin tant que
```

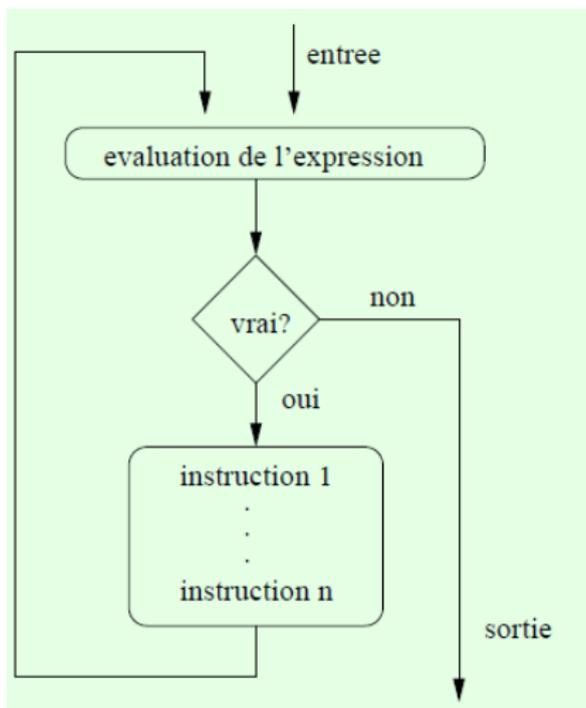
Syntaxe Faire - Tant que

```
faire  
  <instructions>  
tant que <expression booléenne>
```

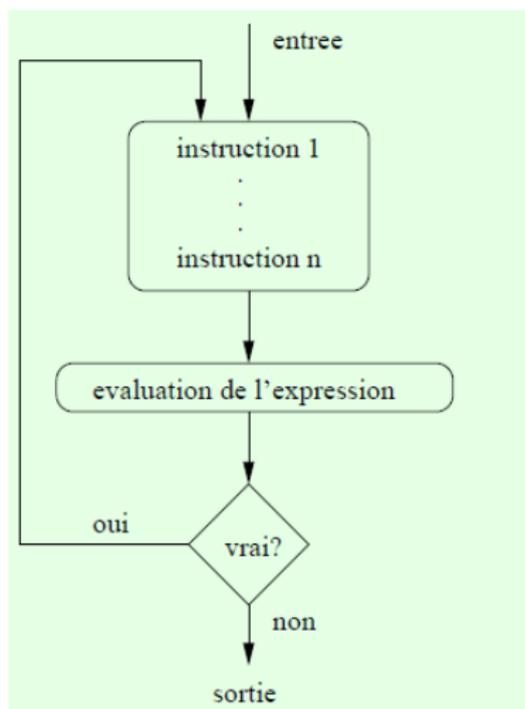
Syntaxe Pour

```
Pour (var de init à fin de pas n)  
  <instructions>  
fin Pour
```

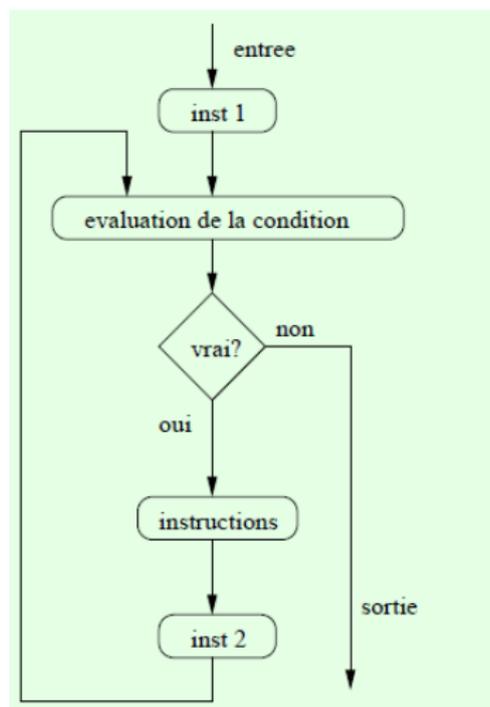
Déroulement d'une boucle Tant que



Déroulement d'une boucle Faire Tant que



Déroulement d'une boucle Pour



Les instructions itératives en Java

L'instruction itérative *tant que*

```
while (expression booléenne)
{
    traitement;
}
```

- L'*expression booléenne*, qui représente la condition de continuation, est évaluée.
- Si elle est vraie, le *traitement* est exécuté et l'*expression booléenne* est évaluée à nouveau ; ce cycle se répète jusqu'à ce que cette expression devienne fausse.

Utilisation de while

Exemple

```
public class ExampleWhile {
    public static void main (String[] args) {
        int n, som;
        som = 0;
        while (som<100)
        {
            System.out.print("donnez un nombre : ");
            n=Lire.i();
            som += n ;
        }
        System.out.print("somme obtenue : "+som);
    }
}
```

Les instructions itératives en Java

L'instruction itérative *faire tant que*

```
do
{
    traitement
}
while (condition) ;
```

Cette instruction répète le *traitement* tant que la *condition* mentionnée est **vraie**.

Utilisation de do while

Exemple

```
public class ExampleDoWhile {
    public static void main (String[] args) {
        int n ;
        do
        {
            System.out.println("donnez un nb > 0 : ");
            n =Lire.i();
            System.out.println("vous avez fourni : "+n);
        }
        while (n <= 0) ;
        System.out.println("reponse correcte");
    }
}
```

Les instructions itératives en Java

L'instruction itérative *pour*

```
for (expr_init; expr_cont; expr_itér)
{
    traitement
}
```

- `expr_init` : expression d'initialisation
- `expr_cont` : expression de continuation
- `expr_itér` : expression d'itération

L'expression d'initialisation est évaluée une seule fois, au début de la boucle. Puis l'expression de continuation est évaluée. Si elle est vraie, le traitement est exécuté, et puis l'expression d'itération est exécutée. Ensuite, l'expression de continuation est évaluée à nouveau ; ce cycle se répète jusqu'à ce que l'expression de continuation devienne fausse.

```
for (i=0; i<n ; i++)
{
    traitement
}
```

Utilisation de for

Exemple

```
public class ExampleFor {  
    public static void main (String[] args) {  
        int i, n, som ;  
        som = 0 ;  
        for (i=0 ; i<4 ; i++)  
        {  
            System.out.println("donnez un entier : ");  
            n = Lire.i();  
            som += n ;  
        }  
        System.out.println("Somme : "+som);  
    }  
}
```

Exercices

Exercice 1 : Boucle pour

Dans une suite de 20 entiers saisis au clavier, calculer :

- 1 La quantité d'entiers négatifs saisis.
- 2 La moyenne des entiers impairs saisis.

Exercice 2 : Boucle Tant Que

Refaire l'exercice 1, mais maintenant, la saisie des données s'arrête après l'arrivée d'une valeur particulière constante FIN (par exemple, 0).

Exercice 3 : Variante de l'exercice 2

Dans une suite d'entiers saisis au clavier comme dans l'exercice 2, calculer le minimum et le rang auquel il a été saisi.

Exercices

Exercice 4 : Contrôle de saisie

Reprendre l'algorithme de l'exercice 4 du Chapitre 2 et le modifier pour que la capacité de chauffage soit saisie par l'utilisateur. Les radiateurs dont on dispose ont une capacité de chauffage comprise entre 0,5 et 8 m³.

Remarque

Pensez à utiliser la boucle "do while" pour contrôler la saisie de l'utilisateur

Exercices

Exercice 5

Parmi les entiers compris entre 2 et 1000, seuls 4 sont égaux à la somme des cubes de leurs chiffres. Ainsi, $153 = 1^3 + 5^3 + 3^3$. Écrire un algorithme permettant de déterminer ces nombres.

Remarque

$153/100 = 1$
 $153\%100 = 53$
etc...

Exercices

Exercice 6 : Le Calendrier

L'objectif de cet exercice est de déterminer quelles années entre 2012 et 2112 sont bissextiles ou non et d'afficher le résultat correspondant.

Depuis l'ajustement du calendrier grégorien, sont bissextiles les années :

- soit divisibles par 4 mais non divisibles par 100
- soit divisibles par 400.