Haute École de Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2018 – 2019
DEV1

DEV1 – Laboratoires Java I

TD 3 – Boucles `while`

Dans ce TD vous trouverez une introduction aux boucles et à l'instruction `while`. Ensuite nous verrons comment lire une série de données.

Table des matières

Version du document
25 juin 2018

1 Boucle - `while`

```java
1  package esi.dev1.td3;
2
3  public class Boucle {
4
5     public static void main(String[] args) {
6        int i = 1;
7        while(i<=10) {
8           System.out.println(i);
9           i = i+1;
10       }
11    }
12 }
```

code/Boucle.java

Le code ci-dessus affiche les nombres de 1 à 10. Ce programme s'exécute comme suit :

- le programme commence à la ligne 6 où la variable i de type entier est déclarée et initialisée à 1 ;
- ensuite, la condition du `while` de la ligne 7 est évaluée, sa valeur est vrai car i vaut 1 et est donc inférieur à 10 ;
- puisque la condition est vraie le *corps* de la boucle s'exécute :
  - la ligne 8 affiche la valeur de l'entier i : 1
  - la ligne 9 assigne la valeur 2 à i (c'est la valeur de l'expression $i + 1$)
- l'exécution du corps de la boucle étant terminé le programme retourne à l'instruction 7, et évalue à nouveau la condition avec la nouvelle valeur de i qui vaut maintenant 2.
- la condition de la ligne 7 étant vraie (car i vaut 2 et est donc inférieur à 10) on exécute à nouveau le corps de la boucle ;
- le programme continue ainsi jusqu'à ce que i atteigne la valeur 11. À ce moment la condition du `while` est évaluée à faux et l'instruction `while` prend fin. Comme aucune instruction ne suit ce `while`, le programme se termine.

Exercice 1 Suites d'entiers

Dans votre package créez un classe `Exercice2`. Dans cette classe écrivez un programme qui demande à l'utilisateur un nombre entier n et affiche :

- les nombres de 1 à n ;
- les nombres pairs qui sont compris entre 1 et n ;
- les nombres de -n à n ;
- les multiples de 5 qui sont compris entre 1 et n ;
- les multiples de n compris entre 1 et 100.

Exercice 2 Ligne

Écrire un programme qui demande à l'utilisateur la longueur d'une ligne et affiche une ligne (une suite de tirets `'-'`) de cette longueur.

Par exemple, si l'utilisateur entre 10, le programme affiche

```
----------
```

Astuce : utiliser la méthode `System.out.print('-')` et non pas `System.out.println('-')`.

2 Lecture de données multiples

Dans cette section nous allons voir comment demander une série de données à l'utilisateur. Par exemple demander d'entrer la température de chaque jour du mois écoulé afin d'en calculer la moyenne. Il y a plusieurs manières de procéder, nous allons en voir 3.

2.1 Variante 1 : le nombre de valeurs est fixe

Dans cette variante le nombre de valeurs à lire est connu et fixé au moment de l'écriture du programme. Ce nombre sera le même chaque fois que l'on lancera le programme. Par exemple on va traiter 10 entiers.

```java
package esi.dev1.td3;

import java.util.Scanner;

public class LectureMultiple1 {

   public static void main(String[] args) {
      Scanner clavier = new Scanner(System.in);

      int somme = 0;
      int i = 1;

      System.out.println("Entrez 10 valeurs");
      while(i <= 10) {
         int valeur = clavier.nextInt();
         somme = somme + valeur;
         i = i+1;
      }

      System.out.println("La somme de toutes ces valeurs: "+ somme);
   }
}
```

code/LectureMultiple1.java

Exercice 3 **Moyenne**

Écrivez un programme qui demande à l'utilisateur 5 valeurs et affiche la somme de ces valeurs et la moyenne de ces valeurs.

Par exemple si l'utilisateur entre 15, 20, 10, 12 et 18, le programme affichera 75 (somme) et 15 (moyenne).

Exercice 4 **Pair ou impair**

Écrivez un programme qui demande 10 nombres entiers à l'utilisateur et pour chacun de ces nombres affiche, au fur et à mesure, s'il est pair ou impair.

2.2 Variante 2 : le nombre de valeurs est connu

Ici le nombre de valeurs à lire est demandé à l'utilisateur. Ce nombre peut être différent à chaque exécution du programme mais il sera connu avant de commencer la lecture des valeurs.

```java
package esi.dev1.td3;

import java.util.Scanner;

public class LectureMultiple2 {
```

```java

   public static void main(String[] args) {
      Scanner clavier = new Scanner(System.in);
      System.out.println("Combien de valeurs voulez-vous entrer?");
      int n = clavier.nextInt();
      int somme = 0;
      int i = 1;

      while(i <= n) {
         System.out.println("entrez une valeur: ");
         int valeur = clavier.nextInt();
         somme = somme + valeur;
         i = i+1;
      }

      System.out.println("vous avez rentré : "+ n +" valeurs");
      System.out.println("La somme de toutes ces valeurs: "+ somme);
   }
}
```

code/LectureMultiple2.java

Exercice 5 **Moyenne**

Modifiez l'Exercice 3 afin que le programme demande à l'utilisateur le nombre n de valeurs qu'il veut introduire, lise ces n valeurs au clavier et affiche la somme et la moyenne.

Exercice 6 **Positifs, négatifs et nuls**

Écrire un programme qui demande à l'utilisateur le nombre n de valeurs qu'il veut introduire, lit ces n valeurs au clavier et affiche le nombres de valeurs positives, le nombre de valeurs négatives et le nombre de valeurs nulles.

Exemple, si l'utilisateur entre les 8 valeurs :

```
5 9 -1 0 12 -7 -4 3
```

le programme affiche

```
positifs : 4
négatifs : 3
nuls : 1
```

2.3 Variante 3 : valeur sentinelle

Dans cette variante, l'utilisateur entre une série de nombres et une valeur spéciale permet d'indiquer la fin des données. Par exemple si l'utilisateur doit entrer des nombres positifs on pourra utiliser -1 comme valeur de fin. Cette valeur spéciale s'appelle *valeur sentinelle*.

```java
package esi.dev1.td3;

import java.util.Scanner;

public class LectureMultiple3 {

   public static void main(String[] args) {
      Scanner clavier = new Scanner(System.in);
      System.out.println("Entrez les valeurs (terminez par -1)");
      int valeur = clavier.nextInt();
      int somme = 0;
      int i = 0;

      while(valeur > -1) {
         somme = somme + valeur;
```

```
16            i = i+1;
17            System.out.println("entrez une valeur: ");
18            valeur = clavier.nextInt();
19        }
20
21        System.out.println("vous avez rentré : "+ i +" valeurs");
22        System.out.println("La somme de toutes ces valeurs: "+ somme);
23    }
24 }
```

code/LectureMultiple3.java

Exercice 7 **Premier et dernier**

Écrivez un programme qui demande à l'utilisateur une série de nombres positifs qui se termine par -1 (valeur sentinelle). Le programme affiche le premier et le dernier nombre de la série.

Exemple, si l'utilisateur entre

```
5 9 1 0 12 7 4 3 -1
```

le programme affiche

```
premier : 5
dernier : 3
```

Exercice 8 **Maximum**

Écrivez un programme qui demande à l'utilisateur une série de nombres positifs qui se termine par -1 (valeur sentinelle). Le programme affiche le maximum de la série.

Par exemple, si l'utilisateur entre

```
5 9 1 0 12 7 4 3 -1
```

le programme affiche

```
maximum : 12
```

3 Exercices Récapitulatifs

Exercice 9 **Maximum et minimum**

Écrivez un programme qui demande à l'utilisateur le nombre n de valeurs qu'il veut introduire, lit ces n valeurs au clavier et affiche le maximum et le minimum.

Attention les valeurs peuvent être négatives.

Exercice 10 **Comparer le premier et le dernier**

Écrivez un programme qui demande à l'utilisateur une série de nombres positifs qui se termine par -1 (valeur sentinelle). Le programme affiche l'un des messages suivants :

- ▷ Le premier est inférieur au dernier
- ▷ Le premier est égal au dernier
- ▷ Le premier est supérieur au dernier
- ▷ Le série est vide (si le premier nombre entré est -1)

Exercice 11 **Série croissante**

Écrivez un programme qui demande à l'utilisateur une série de nombres positifs qui se termine par -1 (valeur sentinelle) et affiche si cette série est strictement croissante ou non. On suppose que la série contient au moins 2 nombres.