

## TP 4 : boucles for

Informatique Fondamentale (IF1)

Semaine du 20 Octobre 2008

### 1 Boucles simples

**Exercice 1.** Écrivez un programme `Moutons` qui demande le nombre de moutons, et qui compte les moutons à haute voix :

```
$ java Moutons
Combien de moutons ? 3
1 moutons
2 moutons
3 moutons
```

Votre programme fonctionne-t-il si l'utilisateur spécifie un seul mouton ou zéro moutons ? Que se passe-t-il si l'utilisateur spécifie un nombre négatif de moutons ?

**Exercice 2.** Modifiez le programme précédent pour qu'il affiche un seul mouton au singulier (« 1 mouton » plutôt que « 1 moutons »).

**Exercice 3.** Écrivez un programme qui demande à l'utilisateur un entier  $n$ , puis affiche avec des « - » une règle de longueur  $n$  :

```
$ java Regle
Longueur ? 53
-----
```

**Exercice 4.** Modifiez le programme précédent pour qu'il affiche une règle graduée :

```
$ java RegleGraduee
Longueur ? 53
Intervalle ? 10
|-----|-----|-----|-----|-----|-----|
```

### 2 Boucles et accumuleurs

**Exercice 5.** Écrivez un programme `Puissance` qui demande  $a \in \mathbf{R}$  (en précision `double` pour nous), ainsi qu'un entier  $n \in \mathbf{N}$ , et qui calcule  $a^n$  sans se servir de la fonction `Math.pow`. Vérifiez que  $2.5^3$  vaut 15.625.

Modifiez ce programme pour traiter le cas où  $n \in \mathbf{Z}$ . Vérifiez que  $2.5^{-3}$  vaut 0.064.

**Exercice 6.** Écrivez un programme qui demande  $n \in \mathbf{N}$  (un `int` positif pour nous), et qui calcule la somme des carrés des  $n$  premiers entiers. Vérifiez que  $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 = 91$ .

**Exercice 7.** Écrivez un programme qui calcule la factorielle d'un entier positif.

**Exercice 8.** Écrivez un programme qui lit une ligne de texte à l'aide de la fonction `Deug.readLine()` et affiche le nombre de voyelles (majuscules ou minuscules, mais non-accentuées) qu'elle contient.

### 3 Boucles et entrées multiples

**Exercice 9.** Écrivez un programme qui lit un entier  $n$ , puis qui lit  $n$  nombres, et affiche la moyenne de ces nombres (sans se servir d'un tableau). Par exemple :

```
$ java Moyenne
Combien de nombres ? 4
Nombre 1 ? 3
Nombre 2 ? 5
Nombre 3 ? 1
Nombre 4 ? 54
La moyenne est 15.75.
```

**Exercice 10.** Écrivez un programme qui lit un entier  $n$ , puis qui lit  $n$  nombres, et affiche leur maximum.

**Exercice 11.** Écrivez un programme qui lit un entier  $n$ , puis qui lit  $n$  lignes, et qui affiche la ligne la plus longue. Par exemple :

```
$ java LigneLaPlusLongue
Combien de Lignes ? 5
Ligne 1 ? Ceci
Ligne 2 ? est
Ligne 3 ? un
Ligne 4 ? essai
Ligne 5 ? ...
Ligne la plus longue : essai
```

### 4 Boucles imbriquées

**Exercice 12.** Écrivez un programme qui demande à l'utilisateur un entier  $n$ , puis affiche avec des étoiles un carré de côté  $n$  :

```
$ java Carre
Côté ? 4
****
```

```
****
****
****
```

**Exercice 13.** Écrivez un programme qui demande à l'utilisateur un entier  $n$ , puis affiche un triangle rectangle de côté  $n$ , pointe vers la droite :

```
$ java Triangle
Côté ? 4
*
**
***
****
```

**Exercice 14.** Même question avec la pointe vers la gauche :

```
$ java Triangle
Côté ? 4
*
**
***
****
```

## 5 De vrais graphiques

La classe « Deug » contient des fonctions qui permettent de faire des graphiques. Dans ce TP, on pourra utiliser les fonctions suivantes :

- `static void startDrawings(int width, int height)`, qui crée une zone d'affichage de taille `width × height` pixels;
- `static void drawPoint(int x, int y)`, qui affiche le point de coordonnées `(x, y)`;
- `static void stopDrawings()`, qui ferme la zone d'affichage.

En particulier, vous n'utiliserez pas les fonctions telles que `drawLine` qui permettent de résoudre la plupart de ces exercices en une seule ligne.

**Exercice 15.** Écrivez un programme qui crée une zone d'affichage de largeur 400 et de hauteur 200, et qui dessine un segment de droite commençant en `(100, 100)` et se terminant en `(300, 100)`.

**Exercice 16.** Écrivez un programme qui affiche un segment de droite allant de `(100, 100)` à `(100, 300)`. (À vous de choisir la taille de la zone d'affichage).

**Exercice 17.** Écrivez un programme qui affiche un rectangle creux ayant des angles en `(100, 100)` et `(300, 300)`.

**Exercice 18.** Même question avec un rectangle plein.

**Exercice 19.** Écrivez un programme qui affiche un segment allant de `(100, 100)` à `(300, 300)`.

**Exercice 20.** Écrivez un programme qui affiche un triangle ayant des sommets en `(100, 100)`, `(100, 300)` et `(300, 300)`.