

TP 5 : fonctions, boucles for

Informatique Fondamentale (IF1)

Semaine du 25 octobre 2010

1 Fonctions

Exercice 1. Sans utiliser les opérateurs logiques `■ || ■`, `■ && ■`, `■ ! ■` etc., écrivez les quatre fonctions suivantes dans une même classe `OperateursBooleens` :

- `■ not ■` de signature `public static boolean not(boolean b)`, qui prend un booléen en paramètre, et qui calcule sa négation logique ;
- `■ and ■` de signature `public static boolean and(boolean b, boolean c)`, qui prend deux booléens en paramètre, et qui calcule leur conjonction logique ;
- `■ or ■` de signature `public static boolean or(boolean b, boolean c)`, qui prend deux booléens en paramètre, et qui calcule leur disjonction logique ; et
- `■ xor ■` de signature `public static boolean xor(boolean b, boolean c)`, qui prend deux booléens en paramètre, et qui calcule leur disjonction exclusive.

Testez vos quatre fonctions (sur toutes les valeurs possibles de leurs paramètres) dans le programme principal de la classe, en affichant leurs résultats.

Exercice 2. Dans une classe `Trie`, écrivez trois fonctions appelées `premier`, `deuxieme` et `troisieme`, qui prennent chacune en paramètre trois entiers, et qui calculent respectivement le plus petit, le deuxième et le plus grand des trois entiers.

Par exemple, `premier` aura la signature

```
public static int premier(int a, int b, int c).
```

Dans le programme principal de la classe, écrivez un programme qui lit trois entiers sur l'entrée standard et les affiche dans l'ordre croissant. Pour cela, utilisez les fonctions `premier`, `deuxieme` et `troisieme`.

2 Boucles avec accumulateurs

Exercice 3. Écrivez un programme `Puissance` qui demande $a \in \mathbf{R}$ (en précision `double` pour nous), ainsi qu'un entier $n \in \mathbf{N}$ (un `int` positif pour nous), 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 4. Écrivez un programme qui demande $n \in \mathbf{N}$, 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 5. Écrivez un programme qui calcule la factorielle d'un entier positif.

3 Boucles et entrées multiples

Dans cette partie, vous ne vous servirez pas de tableaux.

Exercice 6. Écrivez un programme qui lit un entier n , puis qui lit n nombres à virgule flottante, et affiche la moyenne de ces nombres. 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 7. Écrivez un programme qui lit un entier n , puis qui lit n nombres, et affiche leur maximum.

Exercice 8. 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 9. 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 10. 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 11. Même question avec la pointe vers la gauche :

```
$ java Triangle2
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 referme 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 12. 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 13. 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 14. Écrivez un programme qui affiche un rectangle creux ayant des angles en `(100, 100)` et `(300, 300)`.

Exercice 15. Même question avec un rectangle plein.

Exercice 16. Écrivez un programme qui affiche un segment allant de `(100, 100)` à `(300, 300)`.

Exercice 17. Écrivez un programme qui affiche un triangle ayant des sommets en `(100, 100)`, `(100, 300)` et `(300, 300)`.