

TP Algorithmique programmation

TP 2 : Premiers programmes Java : études des boucles

Exercice 1. Écrire une classe `Date` comportant les quatre méthodes suivantes (on testera au fur et à mesure chaque méthode) :

- `estBissextile` qui retourne un booléen indiquant si l'année passée en paramètre est bissextile ou non. On rappelle que les années bissextiles sont les années multiples de 4, à l'exception des années multiples de 100 mais pas de 400. Indication : l'instruction `p%q` donne le reste de la division entière de `p` par `q`. Par exemple, `14%4` vaut 2.
- `estCoherente` qui retourne un booléen indiquant si la date en paramètre décrite par trois entiers est possible. Par exemple, `estCoherente(22,3,1993)` retournera `true` alors que `estCoherente(35,1,2002)` retournera `false`. Indication : on utilisera, de préférence, une conditionnelle par cas.
- `dateLendemain` qui, étant donnée une date décrite par trois entiers en paramètre, affiche la date du lendemain. Indication : il suffit de rajouter 1 à la date en paramètre et de vérifier si le résultat est cohérent...
- `nombreDAnneeBissextile` qui calcule le nombre d'années bissextiles entre deux années données.

Exercice 2. (exercice "papier")

Considérez les boucles suivantes :

```
// Boucle1
int i = 5 ;
while (i > 0) {
    System.out.println ("Test");
    i--;
}
```

```
// Boucle 2
int i = 0 ;
while (i > 0) {
    System.out.println("Test") ;
    i--;
```

```

}

// Boucle 3
for (int i = 1; i != 5 ; i = i + 2)
    System.out.println("Test") ;

// Boucle 4
for (int i = 1; i > 0; i++)
    System.out.println("Test") ;

// Boucle 5
int i = 1 ;
while (i < 100)
    System.out.println(i*i) ;

```

- Combien de messages “Test” affichent les boucles 1, 2, 3 et 4?
- Écrivez une boucle for équivalente à la boucle 1 (c’est à dire, s’exécutant de la même façon).
- Combien de messages affiche la boucle 3 si on change `int i = 1` ; en `int i = 0` ;?
- Combien de tours réalise la boucle 5?

Exercice 3. Écrire une classe **Max** possédant une méthode qui calcule le maximum de n nombres entiers fournis par l'utilisateur. On écrira une première version avec une boucle `for` puis une seconde avec une boucle `while`. Quel est l'invariant de boucle (cf. chapitre 7 du cours)? Pourquoi l'exécution de la boucle est-elle de durée finie?

Exercice 4. Écrire une classe **Puissance** possédant une méthode qui, étant donné un entier x et un entier positif y calcule x^y . Adapter ce programme pour qu'il traite le cas où y est négatif. Quel est l'invariant de boucle?

Exercice 5. Écrire une classe **TableDeMultiplication** qui possède une méthode affichant la table de multiplication d'un chiffre donné.

Exemple d'exécution :

```

$ java TableDeMultiplication 5
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40

```

```
9 * 5 = 45
$
```

Exercice 6. Écrire une classe **Rectangle** qui contient une méthode qui affiche (sous format texte) un rectangle de largeur k et de longueur n en utilisant les caractères +, - et |. Pour tester cette méthode, on écrira une fonction **main** dans laquelle l'utilisateur sera invité à donner les deux entiers positifs k et n .

Exemple d'exécution :

```
$ java Rectangle
```

=> Dans un JOptionPane, on récupère la longueur. Par exemple, ici, 10.

=> Dans un JOptionPane, on récupère la largeur. Par exemple, ici, 5.

=> Résultat :

```
+-----+
|       |
|       |
|       |
|       |
+-----+
```

```
$
```