

## Programmation Objet en Java - Feuille de TD 1

**Exercice 1. Syntaxe**

```
public class Mystery {  
    public static void main(String[] args) {  
        var n = 42; // ou n'importe quelle autre valeur  
        if (n <= 1) {  
            System.out.println(n);  
        } else {  
            var i = 0;  
            var j = 1;  
            var k = 1;  
            while (k < n) {  
                var tmp = i;  
                i = j;  
                j = i + tmp;  
                k++;  
            }  
            System.out.println(j);  
        }  
    }  
}
```

- Quelles sont les différences que vous avez remarquées entre Java et Python dans cet exemple de code ?
- Comment s'appelle le fichier qui contient ce programme ?
- Que fait ce programme ?
- Il existe une boucle `for` en Java dont la forme générale est :

```
for (initialisation; terminaison; increment) {  
    instructions;  
}
```

L'initialisation est une instruction exécutée une seule fois, avant le début de la boucle. La terminaison est une expression booléenne; si elle est fausse la boucle s'arrête. L'incrément est une instruction appelée à la fin de chaque tour de boucle.

Ré-écrivez la boucle `while` de l'exemple en utilisant une boucle `for`.

**Exercice 2. Premiers programmes**

La méthode `Math.random` permet de tirer un nombre aléatoire dans l'intervalle  $[0.0, 1.0[$ .

- Écrire un programme qui tire un nombre aléatoire et l'affiche 2 fois de suite.
- Écrire un programme qui tire un nombre aléatoire et affiche "plus grand" s'il est plus grand que 0.5 et "plus petit" sinon.
- Écrire un programme qui affiche 100 nombres aléatoires...
  - ... en utilisant une boucle `while`;
  - ... en utilisant une boucle `for`;
- Écrire un programme qui tire 100 nombres aléatoires et affiche le maximum de ces 100 nombres.

**Exercice 3. Premiers records**

- a) Écrire un record `Bird` pour représenter des oiseaux. Un oiseau a un nom (`name`) et une envergure (`wingspan`) en cm. Quels types peut-on utiliser pour définir ces champs ?
- b) Écrire une classe `BirdTest` qui contient une méthode `main` qui crée 2 oiseaux et les affiche.
- c) Ajouter de quoi afficher `true` si ces deux oiseaux sont identiques et `false` sinon.
- d) Ajouter de quoi afficher `true` si ces deux oiseaux ont la même envergure et `false` sinon.
- e) Ajouter une méthode `sameWingSpan` qui permet de tester si deux oiseaux ont la même envergure. Où doit-on l'écrire ? Que doit-elle prendre en paramètre ?
- f) Ajouter une méthode `sameName` qui permet de tester si deux oiseaux ont le même nom.

#### Exercice 4. Code de sécurité

- a) Écrire un record `Code` pour représenter des codes de sécurité composés d'une valeur entière `value` et d'une clé `key` qui est une lettre de l'alphabet de type `char`.
- b) Rajouter une méthode `compareTo` qui permet de comparer 2 codes. Le code `c1` est plus grand que le code `c2` si la clé de `c1` est strictement plus grande que la clé de `c2`. En cas d'égalité de clés, `c1` est plus grand que `c2` si sa valeur est strictement plus grande que celle de `c2`. Sinon, les codes sont égaux. La méthode `compareTo` doit renvoyer :
  - une valeur négative si le paramètre est plus grand que le record auquel on l'applique,
  - une valeur positive si le paramètre est plus petit que le record auquel on l'applique,
  - et 0 sinon.
- c) On suppose que l'on dispose d'une méthode `Code.newRandomCode` sans paramètre qui permet de renvoyer un `Code` aléatoire.

Écrire une classe `CodeTest` qui contient une méthode `main` qui tire 10 000 codes aléatoires et affiche le plus grand.
- d) Rajouter de quoi tester et afficher si d'autres codes sont égaux au plus grand parmi ceux qui sont tirés.