

# TP1 : Premiers pas avec Java

Florian Boudin

Module X0IC020 - 2013

## Exercice 1

Dans ce premier exercice, vous allez devoir écrire un premier programme Java. Pour vous aider dans cette tâche, la démarche à suivre a été décomposée étape par étape :

1. Ouvrez votre éditeur de texte préféré (e.g. `gedit`) et écrivez les lignes de code suivantes :

```
public class PremierProgramme {  
    public static void main(String[] args) {  
        System.out.println("Que la force soit avec vous !");  
    }  
}
```

Enregistrez le fichier dans `PremierProgramme.java`. Le nom de votre fichier doit être le même que le nom de votre définition de classe suivi de l'extension `.java`. Ce nom est sensible à la casse, ce qui signifie que vous devez capitaliser les lettres d'une manière identique à celle de la définition de classe.

2. Ouvrez ensuite un **terminal** et naviguez vers le répertoire où vous venez de créer votre fichier. Compilez votre programme en utilisant la commande :

```
javac PremierProgramme.java
```

3. Une fois que le compilateur retourne à l'invite, exécutez l'application en utilisant la commande suivante :

```
java PremierProgramme
```

Quel est le résultat de l'exécution de ce programme ? Modifiez la classe `PremierProgramme` pour que le programme affiche le message *n* fois. Une variable étant utilisée pour stocker la valeur de *n*. Le message devra être suivi de l'itération courante entre parenthèses comme suit :

```
Que la force soit avec vous ! (1)  
Que la force soit avec vous ! (2)  
Que la force soit avec vous ! (3)  
...
```

## Exercice 2

La classe `Scanner` permet entre autres de lire les entrées clavier. Un objet `Scanner` découpe l'entrée clavier en jetons (séparés par des blancs) puis les retourne un à la fois. Il permet de convertir les jetons en valeurs de différents types. Par exemple, ce code lit deux nombres entrés au clavier et affiche leur somme :

```
import java.util.Scanner;  
  
public class Somme {  
    public static void main(String[] args) {  
        Scanner saisie = new Scanner(System.in);  
        int i = saisie.nextInt();  
        int j = saisie.nextInt();  
        System.out.println(i+j);  
    }  
}
```

1. Recopiez, compilez et exécutez le code donné ci-dessus. Quel est le résultat de l'exécution de ce programme ?
2. Modifiez la classe `Somme` pour que le programme demande un entier tant que ce dernier est supérieur à 0. Dans le cas contraire, la somme de tous les nombres entrés au clavier sera affichée.

### Exercice 3

Le but de cet exercice est de vous familiariser avec les messages d'erreur de compilation les plus communs. Recopiez et compilez le code donné ci-dessous. Essayez de situer les erreurs dans le code en fonction des messages d'erreur puis de les corriger.

```
public class Erreurs {
    public static void main(String[] args) {
        int 1champ = 10;
        int champ2 = 10.4;
        boolean champ 3 = true;
        double champ4;
        System.out.println(champ4);
    }
}
```

### Exercice 4

Modifiez la classe `ArithmeticDemo` ci-dessous pour qu'elle utilise des opérateurs d'affectation composés :

```
class ArithmeticDemo {
    public static void main (String[] args){
        int result = 1 + 2; // result is now 3
        System.out.println(result);
        result = result - 1; // result is now 2
        System.out.println(result);
        result = result * 2; // result is now 4
        System.out.println(result);
        result = result / 2; // result is now 2
        System.out.println(result);
        result = result + 8; // result is now 10
        result = result % 7; // result is now 3
        System.out.println(result);
    }
}
```

### Exercice 5

Ici, vous devez vous familiariser avec les structures conditionnelles. Écrivez une classe nommée `TestNumber` contenant dans la méthode `main` la partie de code décrite ci-dessous.

1. Quelle sera l'affichage si la variable `aNumber` est initialisée à 3 ? Compilez et exécutez le programme, est-ce que la sortie est celle que vous avez prévue ?
2. En utilisant uniquement des tabulations/espaces et des retours à la ligne, modifiez cette partie de code pour qu'elle soit plus facile à comprendre.
3. Utilisez des accolades pour clarifier encore plus le code.

```
if (aNumber >= 0)
    if (aNumber == 0)
        System.out.println("first string");
else
    System.out.println("second string");
System.out.println("third string");
```

## Sources

Ce document est une adaptation des exercices disponibles dans le tutoriel de Java disponible à l'adresse <http://docs.oracle.com/javase/tutorial/> et sur Wikibooks. Rédigé par Florian Boudin, 2011–12.