

Premiers pas en Java, variables, tests, boucles, fonctions

Exercice 1 - Hello Groland

On rappelle qu'il est préférable en Java que chaque classe soit définie dans un fichier qui lui est propre. Le nom du fichier doit être le nom de la classe qu'il contient, auquel on ajoute le suffixe **.java**. Les noms des classes doivent être constitués de mots accolés dont la première lettre est une majuscule.

Dans un premier temps nous allons écrire des petits programmes permettant de se familiariser avec le compilateur, la machine virtuelle et les méthodes.

- 1 Créer un répertoire `td-java` dans votre espace de travail (avec la commande `mkdir`)
- 2 puis aller dans ce répertoire, avec la commande `cd`.
- 3 Écrire le programme suivant :

```
public class HelloGroland {
    public static void main(String[] args) {
        System.out.println("Hello Groland");
    }
}
```

dans votre éditeur de texte préféré et sauvegarder celui-ci dans le répertoire `td-java` sous le nom `HelloGroland.java`

- 4 Compiler le programme en utilisant la commande `javac` puis vérifier que le fichier `.class` correspondant existe bien (avec la commande `ls` (unix) ou `dir` (windows).

```
javac HelloGroland.java
```

- 5 Exécuter le programme avec la commande `java`

```
java HelloGroland
```

On ne met pas `".class"` parce que la machine virtuelle le rajoute toute seule.

Exercice 2 - Calculette simple

Écrire un programme prenant un nombre sur l'entrée standard et affichant celui-ci. Pour cela, on utilisera un objet `Scanner` et particulièrement sa méthode `nextInt()`.

```
import java.util.Scanner;

public class Calculette {
    public static void main(String[] args) {
        Scanner scanner;
        scanner=new Scanner(System.in);
        int value;
        value=scanner.nextInt();
        // compléter ici
    }
}
```

- 1 Indiquer dans le programme où sont les variables et quel est leur type associé.

- 2 Recopier le programme précédent et le compléter pour qu'il affiche le nombre saisi par l'utilisateur.
- 3 Modifier le programme pour déclarer et initialiser les variables en une seule ligne
- 4 Expliquer la ligne :

```
import java.util.Scanner;
```
- 5 Modifier le programme pour qu'il demande deux entier et affiche la somme de ceux-ci.
- 6 Afficher en plus de la somme, la différence, le produit, le quotient et le reste.

Exercice 3 - Bien le bonjour

Écrire un programme demandant à un utilisateur quel est son nom et affichant "bonjour" suivi du nom rentré précédemment.

Faite en sorte d'afficher le texte "bonjour" et le nom sur une seul et même ligne

Soit un utilisant `System.out.print()`

Soit en utilisant le concaténation des chaîne de caractère (le +).

Exercice 4 - À propos artificiel

Convertisons avec notre programme.

- 1 Écrire un programme prenant un nombre sur l'entrée standard et affichant :
 - si le nombre est inférieur à 20 la phrase : "vous êtes jeunes"
 - si le nombre est supérieur à 60 la phrase : "vous êtes vieux"
- 2 Ajouter une condition testant si le nombre est superieur ou égal à 20 et inférieur à 30 : la phrase sera alors "vous n'êtes plus tout jeune et pas complètement vieux"

Exercice 5 - Apprendre à compter

- 1 Écrire un programme prenant un nombre sur l'entrée standard et affichant les nombres de 1 à 10 inclus.
Vous utiliserez pour cela une boucle `while`.
- 2 Modifier votre programme pour qu'il demande à l'utilisateur de rentrer une valeur, puis affiche les nombre de 1 à la valeur rentrée.
- 3 Transformer la boucle `while` en boucle `for` et utiliser `i++` pour effectuer l'incréméntation.

Exercice 6 - Compter à l'envers

LCréer un nouveau programme faisant la même chose que le programme précédent mais en affichant les nombres de 10 à 1, donc à l'envers.

Exercice 7 - Afficher les multiples

Afficher les multiples de 3 compris entre 1 et une valeur rentrée par l'utilisateur.

Exercice 8 - Le perroquet

Écrire un programme qui affiche tous les mots que tape l'utilisateur jusqu'à ce que celui-ci

tape controle-D (unix) ou controle-Z suivi de Entrée (windows).

Pour savoir s'il reste des mots à lire sur l'entrée standard, on utilise la méthode `Scanner.hasNext()`.

Exercice 9 - Calcul flottant et précision

On souhaite multiplier 1.0 par 2.0 puis utiliser le résultat et multiplier celui-ci encore par 2.0 jusqu'à ce que la valeur soit la même d'une itération à l'autre.

Expliquer pourquoi la valeur va finir par se stabiliser.

Quelle est cette valeur ?

Même chose si l'on essaye de diviser par 2.0 à chaque itération au lieu de multiplier.

Indiquer combien d'itération il a fallu dans le cas où l'on multiplie par 2 en utilisant des `float` puis en utilisant des `double`.