

Université d'Aix-Marseille – L2 Informatique
Introduction à la programmation orientée objet
TP n°1

Une première classe

À l'aide d'un éditeur de texte, saisir le code Java ci-dessous, le compiler (commande `javac`) et puis l'exécuter (commande `java`).

```
public class MonPremierProgrammeJava{
    public static void main(String args[]){
        System.out.println("Coucou, je suis étudiant(e) en L2MI");
    }
}
```

1. Combien de classes y a-t'il dans ce programme?
2. Quel nom associer au fichier d'extension `.java`?
3. Quel est le résultat de l'exécution de ce programme?
4. Quel est donc le rôle de l'instruction `System.out.println`?

Tableaux, saisie et affichage

On dispose de la méthode `LireCaractere` qui permet la lecture d'une chaîne de caractère depuis le clavier. Dans un premier temps, il n'est pas demandé de comprendre toutes les instructions. Par ailleurs, afin de pouvoir l'utiliser, il faudra ajouter à l'entête de votre programme la directive d'inclusion du package de gestion des entrées/sorties suivante: `import java.io.*;`

```
static String LireCaractere(){
    String s=null;
    //Définition d'une source de données (ici le clavier)
    BufferedReader source = new BufferedReader(new InputStreamReader (System.in));
    try{
        s=source.readLine();
    }
    catch (java.io.IOException e){
        System.out.println("Erreur de lecture depuis le clavier");
        System.exit(-1);
    }
    return s;
}
```

Écrire un programme Java qui effectue les opérations suivantes :

1. Lecture d'un entier `taille` au clavier. On utilisera la méthode `Integer.parseInt(String)` qui permet la conversion d'une chaîne de caractère passée en paramètre en un entier.
2. Déclaration d'un tableau `entiers` de `taille` d'entiers.
3. La recherche et l'affichage à l'écran du plus grand et du plus petit élément de `entiers`.
4. Le tri dans un ordre croissant des éléments de `entiers`.
5. Le calcul et l'affichage de la moyenne arithmétique des éléments de `entiers`.

Des arguments

1. Écrire un programme Java (`Afficheur.java`) qui affiche les arguments qui lui sont passés en paramètres. Par exemple :

```
$ java Afficheur programmation facile
$ programmation facile
```

2. Écrire un programme Java qui permet de calculer le factoriel d'un nombre entier passé en paramètre.

Piles

Écrire une classe Java `Pile` qui permet de modéliser une pile d'entiers initialement vide. En outre, cette classe possède les méthodes suivantes :

1. `void empiler(int element)` qui rajoute la valeur `element` au sommet de la pile.
2. `int depiler()` qui retourne puis supprime la valeur de l'élément qui est au sommet de la pile.
3. `boolean estVide()` qui retourne `true` si la pile est vide et `false` sinon.
4. `int taille()` qui retourne le nombre d'éléments de la pile.
5. La méthode `main` pour tester la classe `Pile`.

Compte à rebours

Écrire la classe `CompteRebours` pour modéliser le principe d'un compte à rebours et qui contient :

1. L'attribut `max` de type `int` qui correspond à la valeur maximale du compteur.
2. La méthode `dec()` qui décrémente la valeur de `max` si celle-ci n'est pas nulle.
3. La méthode `valeur()` qui retourne la valeur de `max`.
4. La méthode `affiche()` qui affiche la valeur de `max`. Si `max=0` alors le message "BIP BIP" est affiché.
5. Tester cette classe dans une méthode `main`.