TD1

Exercice 1 **-** Conventions
Soit le programme java suivant :

```
public class Calculatrice {
  public static void main(String[] args) {
    if (args.length !=3) {
       System.out.println("Nombre d'arguments incorrect ! ");
    }
    else {
      int op1 = Integer.parseInt(args[0]);
      int op2 = Integer.parseInt(args[2]);
      switch (args[1].charAt(0)) {
        case '+' :   System.out.println(op1+op2);
                     break;
        case '-' :   System.out.println(op1-op2);
                     break;
        case'*'  :   System.out.println(op1*op2);
                     break;
        case '/' :   System.out.println(op1/op2);
                     break;
        default :    System.out.println("operateur inconnu ...");
      } // switch
   } // else (nombre d'arguments corrects)
 } // method main
} // class
```

Lecture du code source

1. Donner le nombre et le nom de(s) classe(s) dans ce code source.
2. Donner le nom des attributs pour chaque classe du code source.
3. Donner le nom des méthodes pour chaque classe du code source.
4. Indiquer s'il y a un constructeur dans ce code source.
5. Vérifier que la syntaxe est respectée (définition de classes, de méthodes, instruction switch).
6. Vérifier que les conventions de nommage des identifiants sont respectées (nom de classes, d'attributs, de méthodes).

7. Expliquer pourquoi les instructions System.out.println(...) sont correctes.
8. Préciser pour chaque méthode si elle est méthode de classe ou méthode d'instance.

Édition - compilation – exécution

Donner les différentes étapes à effectuer pour exécuter un programme donné sous forme de code source

1. Dire quel doit être le nom du fichier source.
2. Dire comment compiler le programme.
3. Dire comment exécuter le programme. Donner quelques exemples.

   Remarque : * sur la ligne de commande sera mal interprété (interprétation du SE) ;

   pour cet opérateur taper "*" ou *.

Modification du programme

1. Modifier l'affichage du résultat pour avoir un texte tel que :
   - "Le résultat de la somme est : ..." . Idem pour les autres opérateurs.
   - si les arguments sont 5 + 6, l'affichage est "5 + 6 = 11". Idem pour les autres opérateurs.
2. Ajouter les instructions permettant de calculer un modulo (%) et x^y (x à la puissance y).

   Remarque :
   - utiliser dans la classe Math, la méthode pow définie comme suit :

     public static double pow (double a, double b)
   - faire des cast (double → int) avant d'afficher le résultat final.
     Dans les anciennes versions de Java, il était impératif de faire des cast int → double pour utiliser la méthode pow (sinon, pas d'erreur mais résultat faux).
3. Modifier le programme pour que les lectures soient interactives (lecture terminal).

Amélioration du programme : gestion des erreurs

1. Donner les erreurs susceptibles de se produire et qui sont traitées (maladroitement) dans ce code source.
2. Donner les autres erreurs susceptibles de se produire et non traitées dans le code source.

Exercice 2 – Création d'objets

Considérer le fichier CreationPoint.java suivant :

```
class Point {
    public int x ;
    public int y ;

    public Point(int x0, int y0) {
        x=x0 ;
        y=y0;
    }
}

public class CreationPoint {
    public static void main (String [] args) {
        System.out.println("Bonjour !") ;
        Point point1 = new Point(0,0) ;
        Point point2 = new Point(3,4)
        Point point3 = new Point(7,1);
    }
}
```

1. Donner les noms des fichiers et les instructions pour sauvegarder, compiler et exécuter ce programme.
2. Donner l'ordre des méthodes invoquées par le programme.
3. Ajouter du code dans la méthode main pour afficher les coordonnées des points point1, point2 et point3.
4. Une autre manière, plus propre, de faire ceci est d'ajouter une méthode afficher à la classe Point. Ecrire cette méthode ainsi que les appels correspondants dans la méthode main de la classe CreationPoint.

Note : nous verrons ultérieurement une solution plus élégante qui consiste à redéfinir la méthode toString().

Exercice 3 – Types

1. Quels sont les types primitifs existants ?
2. Quelle est la différence entre int et Integer ?
3. Écrire un programme qui calcule la moyenne des nombres entiers passés en arguments de la ligne de commande.

   Écrire les 2 versions : les données sont de type int, les données sont de type Integer.

   Remarque :

   - utiliser args.length
   - pour convertir un String en Integer : utiliser dans la classe Integer, le constructeur Integer(String)