

Exercises 3: some small programs

Exercise 3.1: compile errors

Question 1: classifying errors

The following program does not compile.

Report errors, and explain the nature of error: syntax, typing, variable undeclared variable declared but not initialized or invalid method call.

```
1 public class Td3 1 {
2     public static void main (String [] arguments)  {
3         d double;
4         int x=7;
5         int z;
6         char c = 'k';
7         x = x + z;
8         Terminal.ecrireString("c= " + c);
9         boolean tt;
10        tt=x+4;
11        bb = 3 < tt;
12        x+1 = 3;
13        Terminal.lireInt();
14        Terminal.lireInt(7);
15        Terminal.ecrireInt(Terminal.lireInt() + 4);
16        Math.min(Terminal.lireInt(), x);
17        Terminal.ecrireInt(Math.min(Terminal.lireInt(), x));
18        x = Terminal.ecrireInt(5);
19        Terminal.ecrireInt(5);
20        Terminal.ecrireInt();
21    }
22 }
```

Question 2: Results of unused calls

In the previous program, some method calls return a result, but these results are recovered nowhere: they are stored in any variable, or used in a calculation, nor are given in argument to another call method. These are not compilation errors but probably the **program logic errors**. Report these calls.

Exercise 3.2: coding algorithms

Code in Java the following algorithms (given in the first sheet of exercises).

1. **Calculate** the final grade material according to the following rules. The final grade depends on notes obtained during the final exam (mandatory), and the partial examination (which is optional). The calculation rule is:
 - a score below 7 on the final exam is eliminatory,
 - we take the average of the two scores (final part) if the average benefits the student.
2. **Test** if three given (keyed) integers are sorted in ascending order.
3. **Calculate** and display the largest of three integers.

Exercise 3.3: blocks and local variables

Question 1: Display, memory

Give the messages displayed by the following program and show the variable values in memory.

```
3      int a = 2;
4      int x = 4;
5      Terminal.ecrireStringln(" * Bloc main * ");
6      Terminal.ecrireStringln("a =" + a );
7      Terminal.ecrireStringln("x =" + x );
8      if (a<12) {
9          /* Bloc 1 */
10         int k = 12;
11         x = a+x;
12         k = k+1;
13         Terminal.ecrireStringln(" * Bloc 1 * ");
14         Terminal.ecrireStringln("a =" + a );
15         Terminal.ecrireStringln("k (locale) =" + k );
16         Terminal.ecrireStringln("x =" + x );
17         Terminal.ecrireStringln(" * Fin Bloc 1 * ");
18     }
19     // Terminal.ecrireStringln("k =" + k );
20     int k = 3;
21     Terminal.ecrireStringln("k =" + k );
22     Terminal.ecrireStringln("a =" + a );
23     Terminal.ecrireStringln("x =" + x );
24 }
25 }
```

Question 2

What happens in this program if you remove the comments in the line? Is this program correct?

Exercice 3.4 : petits programmes

Question 1 : positif ou négatif

Ecrire un programme qui détermine si un nombre est positif ou négatif. Il considérera 0 comme positif. Ce programme demandera un nombre à l'utilisateur puis affichera si le nombre est positif ou négatif au moyen d'un message approprié (chaîne de caractère).

Question 2 : rectangle

Le programme suivant (exemple du chapitre 2) dessine à l'écran un rectangle composé de **l** lignes d'étoiles.

```
public class Rectangle2 {
    public static void main (String[] args) {
        int l;
        Terminal.ecrireString("combien de lignes d'étoiles ?:");
        l=Terminal.lireInt();
        for (int i=0;i<l;i=i+1) {
            Terminal.ecrireStringln("*****");
        }
    }
}
```

Modifiez ce programme pour qu'il affiche un rectangle vide au lieu d'un rectangle plein. Par exemple, si **l** vaut 5, le rectangle sera :

```
*****
*   *
*   *
*   *
*   *
*****
```

Pour cela, il faut afficher séparément la première et la dernière ligne, alors que les autres peuvent être affichées par une boucle.

Question 3 : sommes des premiers entiers

1. Ecrivez un programme qui calcule la somme des 10 premiers entiers. Cette somme sera mise dans une variable qui contiendra d'abord la somme des 0 premiers entiers, puis celle des 1 premiers entiers, puis des 2 premiers entiers, etc. A la fin du calcul, le résultat sera affiché à l'écran. Vous utiliserez une boucle for. Vous pouvez vous inspirer de l'exemple `premEntiers` du chapitre 2 (page 13).
2. Modifiez le programme pour qu'il calcule la somme des *n* premiers entiers où *n* est un nombre saisi au clavier et entré par l'utilisateur.
3. Modifier le programme pour calculer non plus la somme mais la moyenne des *n* entiers.