

Feuille d'exercices (3): petits programmes

Exercice 3.1 : erreurs à la compilation

Question 1 : classifier les erreurs

Le programme suivant ne compile pas. Signalez les erreurs, et expliquez leur nature : de syntaxe, de typage, variable non déclarée, variable déclarée mais non initialisée, ou appel de méthode invalide.

```
1 public class Td31 {
2     public static void main (String [] arguments) {
3         double d;
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 : résultats d'appels non utilisé

Dans le programme précédent, certains appels de méthodes retournent un résultat, mais ces résultats ne sont récupérés nul part : ils ne sont stockés dans aucune variable, ni utilisés au cours d'un calcul, ni sont donnés en argument d'un autre appel de méthode. Ce ne sont pas des erreurs de compilation mais probablement de la logique du programme. Signalez ces appels.

Exercice 3.2 : codage d'algorithmes

Codez en Java les algorithmes suivants (donnés en exercice dans la feuille 1 d'exercices).

1. Calculer la note finale d'une matière selon les règles suivantes. La note finale dépend des notes obtenues lors de l'examen final (obligatoire), et de l'examen partiel (qui est optionnel). La règle de calcul est :
 - une note inférieure à 7 à l'examen final est éliminatoire,
 - on prend la moyenne des deux notes (final et partiel) si cette moyenne avantage l'étudiant.
2. Tester si trois nombres entiers saisis au clavier sont triés dans l'ordre croissant.
3. Calculer puis afficher le plus grand parmi trois nombres entiers saisis au clavier.

Exercice 3.3 : blocs et variables locales

Question 1 : Affichages, mémoire

Donnez les messages affichés par le programme suivant et dessinez les valeurs des variables en mémoire.

```
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

Que se passe-t-il dans ce programme si on enlève les commentaires dans la ligne ? Ce programme est-il 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.