

exercices divers

Exercice 2.2.1 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 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 }
```

Réponse :

- ligne 3 : erreur de syntaxe. La syntaxe correcte est : `double d;`
- ligne 7 : la variable `z` est déclarée mais n'a pas de valeur initiale alors qu'elle est nécessaire pour réaliser le calcul de l'expression `x + z` à droite de l'affectation.
- ligne 10 : erreur de typage. Le type de l'expression `x + 4` à droite de l'affectation est `int`. Il est incompatible avec le type de la variable `tt` à gauche, qui est `boolean`;
- ligne 11 : deux erreurs
 - la variable `bb` n'est pas déclarée ;
 - (erreur de typage) on ne peut pas utiliser l'opérateur de comparaison `>` sur un `int` et un `boolean`;
- ligne 12 : l'opérateur d'affectation `=` doit avoir à gauche une variable et non pas une expression ;

-
- ligne 14 : erreur de typage. L'appel de méthode `Terminal.lireInt(7)` ne doit pas prendre d'argument ;
 - ligne 18 : erreur de typage. L'appel de méthode `Terminal.ecrireInt(5)` ne retourne pas un résultat de type `int`, qui est le type attendu pour affecter la variable `x` ;
 - ligne 20 : erreur de typage. L'appel de méthode `Terminal.ecrireInt()` doit prendre un argument.

Question 2 résultats d'appels non utilisés

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.

Réponse : Il s'agit des appels dans les lignes 13, 14 et 16. Dans la ligne 16, c'est le résultat de `Math.min` qui est non utilisé.

Exercice 2.2.2 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.

Réponse : Dans ce programme nous supposons que les notes saisies sont correctes : on se dispense donc de tests de validation des notes saisies. L'énoncé du problème ne nous dit pas quelle doit être la note finale de l'élève au cas où il est "éliminé", autrement dit, s'il a une note inférieure à 7 à l'examen final. Dans ce cas, il est clair qu'on ne peut pas lui attribuer la moyenne des deux notes, car il pourrait ne pas être éliminé. Nous avons choisi de lui attribuer la note de l'examen. Notez que ce programme ne calcule la moyenne des deux notes que si elle avantage l'élève.

```
1 public class NoteMatiere {
2     public static void main (String[] args) {
3         /* Calcule la note finale a partir des notes d'examens
4            partiel et final.
5            On suppose:
6            -les notes saisies sont correctes.
7            -si une epreuve n'est pas passee, sa note saisie est 0
8         */
9         double nPart, nExam, nFinale;
10        Terminal.ecrireString("Entrez la note du partiel (0 si absent): ");
11        nPart = Terminal.lireDouble();
12        Terminal.ecrireString("Entrez la note a l'examen (0 si absent): ");
13        nExam = Terminal.lireDouble();
14        if (nExam < 7 || nExam >= nPart) {
15            nFinale = nExam;
```

```

16         } else {
17             nFinale = (nExam + nPart)/2;
18         }
19         Terminal.ecrireStringln("La note finale est: "+ nFinale);
20     }
21 }

```

2. Tester si trois nombres entiers saisis au clavier sont triés dans l'ordre croissant.

Réponse : Cette réponse utilise une variable booléenne qui est affectée (ligne 14) avec la valeur de la condition qui teste si les trois entiers sont triés. Notez également que le test sur la valeur de `cr` ligne 15 est `if (cr) ...` et non pas `if (cr==true) ...` qui aurait été redondant.

```

1 public class TestTroisInts {
2     public static void main (String[] args) {
3         /* Teste si trois entiers saisis au clavier sont triés
4            dans l'ordre croissant (non strict).
5         */
6         int a,b,c;
7         boolean cr;
8         Terminal.ecrireString("Entrez un nombre entier: ");
9         a = Terminal.lireInt();
10        Terminal.ecrireString("Entrez un nombre entier: ");
11        b = Terminal.lireInt();
12        Terminal.ecrireString("Entrez un nombre entier: ");
13        c = Terminal.lireInt();
14        cr = (a <= b && b <= c);
15        if (cr) {
16            Terminal.ecrireStringln("Les entiers sont triés. ");
17        } else {
18            Terminal.ecrireStringln("Les entiers ne sont pas triés. ");
19        }
20    }
21 }

```

Une autre solution con site à calculer la valeur de `cr` à l'aide d'une conditionnelle de la forme

```

1     if (a <= b && b <= c) {
2         cr = true;
3     } else {
4         cr = false;
5     }

```

3. Calculer puis afficher le plus grand parmi trois nombres entiers saisis au clavier.

Réponse : Une solution qui demande peu de tests et qui est facile à lire, est de garder dans une variable `max` le plus grand parmi `a` et `b`, puis de comparer cette valeur avec celle de `c`.

```

1 public class Max3Int {
2     public static void main (String [] arguments) {
3         /* Calcule et affiche la valeur la plus grande
4            parmi celles de 3 entiers saisis au clavier
5         */
6
7         int a, b, c, max;

```

```

8      Terminal.ecrireString("valeur de a : ");
9      a=Terminal.lireInt();
10     Terminal.ecrireString("valeur de b : ");
11     b=Terminal.lireInt();
12     Terminal.ecrireString("valeur de c : ");
13     c=Terminal.lireInt();
14     if (a>b){
15         max = a;
16     } else {
17         max = b;
18     }
19     if (c>max){
20         max = c;
21     }
22     Terminal.ecrireString("La valeur la plus grande est ");
23     Terminal.ecrireIntln(max);
24 }
25 }

```

Exercice 2.2.3 blocs et variables locales

Question 1 affichages, mémoire

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

```

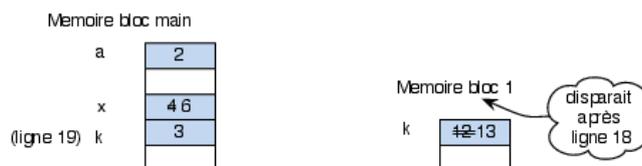
1  public class Exo3_3 {
2      public static void main (String[] args) {
3          int a = 2;
4          int x = 4;
5          Terminal.ecrireStringln("*_Bloc_main_*");
6          Terminal.ecrireStringln("a_=" + a );
7          Terminal.ecrireStringln("x_=" + x );
8          if (a<12) {
9              /* 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 }

```

Ce programme affiche

```
> java Exo3_3
* Bloc main *
a =2
x =4
* Bloc 1 *
a =2
k (locale) =13
x =6
* Fin Bloc 1 *
k =3
a =2
x =6
```

Dans ce programme la mémoire du bloc le plus externe (“bloc main”) possède les variables a et x, qui sont visibles également dans le bloc plus interne (“bloc 1”). Ces variables sont modifiées dans ce dernier bloc. Le bloc 1 déclare une variable locale k, qui disparaît une fois ce bloc terminé. Dans la ligne 19, une nouvelle variable k, est déclarée dans le bloc du main.



Question 2

Que se passe-t-il dans ce programme si on enlève les commentaires dans la ligne 19 ? Ce programme est-il correct ?

Ce programme ne compile pas, car la variable k est inconnue à ce point.

Exercice 2.2.4 petits programmes

Question 1 positif ou négatif

Écrire un programme qui détermine si un nombre est positif ou négatif. Il considérera 0 comme positif.

```
class Exo3_4_1{
    public static void main (String[] arguments){
        int le_nombre;
        Terminal.ecrireStringln("Entrez_un_nombre:");
        le_nombre = Terminal.lireInt();
        if (le_nombre<0){
            Terminal.ecrireStringln("Ce_nombre_est_negatif");
        }else{
            Terminal.ecrireStringln("Ce_nombre_est_positif");
        }
    }
}
```

```
    }  
  }  
}
```

Question 2 *rectangle*

Modifiez le programme `Rectangle2` pour qu'il affiche un *rectangle vide* au lieu d'un rectangle *plein*.

```
class Exo3_4_2{  
  public static void main (String[] args) {  
    int l;  
    Terminal.ecrireString("combien_de_lignes_d'étoiles_?:" );  
    l=Terminal.lireInt();  
    Terminal.ecrireStringln("*****");  
    for (int i=0;i<l-2;i=i+1){  
      Terminal.ecrireStringln("*  *");  
    }  
    Terminal.ecrireStringln("*****");  
  }  
}
```

Question 3 *sommes des premiers entiers*

1. Écrivez un programme qui calcule la somme des 10 premiers entiers.

```
class Exo3_4_3{  
  public static void main (String[] args) {  
    int somme = 0;  
    for (int i=10;i>0;i=i-1){  
      somme = somme + i;  
    }  
    Terminal.ecrireStringln("la_somme_des_10_premiers_entiers_vaut_ " +  
                           somme);  
  }  
}
```

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.

```
class Exo3_4_3_2{  
  public static void main (String[] args) {  
    int somme = 0;  
    int n;  
    Terminal.ecrireString("combien_d'entiers_?:" );  
    n=Terminal.lireInt();  
    for (int i=n;i>0;i=i-1){  
      somme = somme + i;  
    }  
    Terminal.ecrireStringln("la_somme_des_ " + n +  
                           "_premiers_entiers_vaut_ " + somme);  
  }  
}
```

```
}  
}
```

3. Modifier le programme pour calculer non plus la somme mais la moyenne des n entiers. **Attention** : en Java, l'opération de division appliquée sur deux entiers est la division entière, qui donne en résultat un entier. Ainsi, $5 / 2$ donne en résultat 2 et non pas 2.5 . Si vous souhaitez que l'opération appliquée sur les deux nombres soit la division décimale, au moins l'une des deux opérands doit être un nombre à virgule (par exemple un `double`). Par exemple, l'opération $5 / 2.0$ donne en résultat 2.5 .

Solution : La somme de nombres calculée dans le programme précédent doit être divisée par n , et stockée dans une variable moyenne de type `double`. Mais, afin de nous assurer que la division employée lors de l'opération `somme / n` ligne 11, est bien une divisions décimale, nous déclarons la variable `somme` de type `double`.

```
1 class Exo3_4_3_3{  
2     public static void main (String[] args) {  
3         double somme = 0;  
4         int n;  
5         double moyenne;  
6         Terminal.ecrireString("combien_d'entiers_?:" );  
7         n=Terminal.lireInt();  
8         for (int i=n;i>0;i=i-1){  
9             somme = somme + i;  
10        }  
11        moyenne = somme / n;  
12        Terminal.ecrireStringln("la_moyenne_des_ " + n +  
13                               "_premiers_entiers_vaut_" + moyenne);  
14    }  
15 }
```
