

IF1: Introduction à l'informatique et à la programmation

Premier devoir sur table du groupe M1
21 octobre 2011 - Durée: 1h00

*Documents non autorisés. Le barème est donné à titre indicatif.
La lisibilité du code Java sera prise en compte dans l'évaluation.*

Exercice 1 (6 points)

Le score d'un match de rugby est calculé selon les règles suivantes:

- essai transformé: 7 points.
- essai non transformé: 5 points.
- pénalité transformée: 3 points.

Écrire une méthode `scoreDeRugby` qui prend comme paramètres 6 entiers désignant dans l'ordre: le nombre d'essais transformés par l'équipe A, le nombre d'essais non transformés par l'équipe A, le nombre de pénalités transformées par l'équipe A, le nombre d'essais transformés par l'équipe B, le nombre d'essais non transformés par l'équipe B, le nombre de pénalités transformées par l'équipe B, et qui affiche l'équipe gagnante.

Par exemple, l'appel `scoreDeRugby(2,0,3,1,1,1)` affichera:

L'équipe A gagne

le score étant, dans ce cas de figure, de 23 à 15 en faveur de A.

L'appel `scoreDeRugby(0,0,3,1,1,0)` affichera:

L'équipe B gagne

le score étant de 12 à 9 en faveur de B.

Finalement l'appel `scoreDeRugby(0,3,0,0,0,5)` affichera:

match nul

le score étant de 15 à 15.

Exercice 2 (10 points)

Un triplet d'entiers positifs (a, b, c) est un *triplet pythagoricien* si $a^2 + b^2 = c^2$. Par exemple, $(3, 4, 5)$ est un triplet pythagoricien, et $(5, 12, 13)$ en est un autre. Par contre, $(5, 3, 4)$ n'est pas un triplet pythagoricien.

- Écrire une méthode `tripletPythagoricien` qui:
 - prend comme arguments trois entiers positifs,
 - renvoie la valeur booléenne `true` si les trois arguments forment un triplet pythagoricien, pris dans un certain ordre.
 - renvoie la valeur `false` sinon.

Par exemple, les appels `tripletPythagoricien(3,4,5)` et `tripletPythagoricien(5,3,4)` doivent renvoyer `true`, et l'appel `tripletPythagoricien(5,2,4)` doit renvoyer `false`.

- Écrire une méthode `trouveTriplet` qui prend comme arguments deux entiers positifs `a` et `b`, qui renvoie l'entier `c` tel que (a,b,c) est un triplet pythagoricien, si un tel entier existe, et qui renvoie l'entier 0 sinon. La méthode `trouveTriplet` ne doit utiliser ni de méthodes de la classe `Math`, ni de variables de type `double` (donc, une boucle est nécessaire pour trouver l'entier `c` dont il est question).
- Écrire une classe `Triplet` qui contient la méthode `trouveTriplet` ci-dessus (qu'il n'est pas nécessaire de réécrire), et une méthode `main` qui demande à l'utilisateur de rentrer au clavier deux entiers positifs, qui affiche ensuite le troisième élément du triplet pythagoricien dont les deux premiers éléments sont ceux donnés par l'utilisateur, si un tel triplet existe, et qui signale sinon qu'un tel triplet n'existe pas.

Voici deux résultats de l'appel `java Triplet`:

```
>java Triplet
Entrer le premier entier positif:
3
Entrer le deuxième entier positif:
4
3,4,5 est un triplet pythagoricien

>java Triplet
Entrer le premier entier positif:
3
Entrer le deuxième entier positif:
5
Aucun triplet pythagoricien commence par 3,5
```

Exercice 3 (4 points)

On considère la classe suivante:

```
class Test{
    public static void boucle (int n){
        int i;
        for (i=0;i<n;i++)
            {System.out.print(" ");}
    }
    public static void main(String[] args){
        int j;
        for (j=1;j<5;j++)
            {System.out.print("a"); boucle(j);System.out.println("b");}
    }
}
```

Qu'affiche l'exécution du `main` de la classe `Test`?