

TD5

Rappels sur les methodes statiques

1. Static ou pas static ?

Précisez si les méthodes proposées sont static ou non :

- La méthode `cos` de la classe `Math` (qui prend en entrée un double et renvoie un double) ?
- La méthode `get` de la classe `ArrayList` ?
- La méthode `random` de la classe `Math` qui renvoie un double compris entre 0 et 1 ?
- La méthode `equals` de la classe `Object` ?
- La méthode `exit` de la classe `System` qui permet de quitter le programme courant ?
- La méthode `println` utilisée dans `System.out.println` ?
- La méthode `toString` de la classe `Object` ?

2. Si on essaye de compiler le code suivant, quel sera le message d'erreur généré ?

```
public class Machin{
    public int renvoie5(){
        return 5;
    }
    public static int chose(){
        return renvoie5();
    }
}
```

3. Selon vous, le code suivant produit-il une erreur à la compilation ?

```
public class Machin{
    public static int renvoie5()
    {
        return 5;
    }
    public int chose()
    {
        return renvoie5();
    }
}
```

Object : equals et toString

1. Précisez les affichages (1) à (8) générés par le code suivant. Expliquez.

```

public class Executable
{
    public static void main(String[] args)
    {
        Integer x = 3;
        Integer y = new Integer(3);
        System.out.println( x == y);           //(1)
        System.out.println( x.equals(y));       //(2)
        System.out.println( x.toString() );     //(3)
        System.out.println( x );               //(4)
        System.out.println( x.hashCode() == y.hashCode() ); //(5)

        List<Object> tableau = new ArrayList<Object>();
        List<Object> tab2 = new ArrayList<Object>();
        tableau.add(new Integer(5));
        tableau.add(new Double(5.));
        tableau.add(new Boolean(true));
        tableau.add(tab2);
        System.out.println(tab2);              //(6)

        tableau = new ArrayList<Object>();
        tab2 = new ArrayList<Object>();
        tableau.add(new Integer(4));
        tab2.add(new Integer(4));
        System.out.println(tab2 == tableau);    //(7)
        System.out.println(tab2.equals(tableau)); //(8)
    }
}

```

Pour un jeu de SpaceInvaders on a défini une classe projectile de la manière suivante :

```

public class Projectile
{
    private int posX, posY;

    public Projectile(int x, int y)
    {
        this.posx = x;
        this.posy = y;
    }

    public String toString()
    {
        return "Projectile à la position : (" + this.posx + "," + this.posy + ") ";
    }
}

```

2. Précisez les affichages (1) à (4) générés par le code suivant. Expliquez.

```

public class Executable
{
    public static void main(String[] args)
    {
        Projectile p1;
        Projectile p2 = new Projectile(10,10);
        Projectile p3 = new Projectile(10,10);
        System.out.println(p1);              //(1)
        System.out.println(p2);              //(2)
        System.out.println(p2 == p3);         //(3)
        System.out.println(p2.equals(p3));    //(4)
    }
}

```

3. Comment faire pour que la méthode `equals` de `Projectile` fonctionne comme nous le voudrions ?
4. Dans l'exécutable suivant, ajouter un morceau de code pour calculer la somme des éléments de `tableau` sous la forme d'un entier.

```
public class ExecNumberArray
{
    public static void main(String [] args)
    {
        ArrayList<Number> tableau = new ArrayList<Number>();
        tableau.add(5);
        tableau.add(6.);
        tableau.add(7.f);
    }
}
```

5. Quelles sont les méthodes `toString()` qui sont appelées lorsqu'on exécute l'instruction suivante :

```
System.out.println(tableau);
```

6. Quel sera l'affichage généré par l'exécution du code suivant ? Pourquoi ?

```
Number x = new Integer(5);
Number y = new Double(5);
System.out.println(x.equals(y));
```

1. Écrivez une méthode qui prend en paramètre deux listes d'entiers et qui renvoie vrai si les deux listes ont au moins un élément en commun et faux sinon. Par exemple, sur l'entrée `[1,5,2,4], [12,3,7,9]` la réponse est `false`, et sur l'entrée `[1,45,3,2], [6,5,1,12]` la réponse est `true`.
2. Écrivez une méthode qui prend en paramètre une liste et un nombre (double), et qui renvoie la valeur de l'entier le plus proche du nombre appartenant à la liste. Par exemple, sur l'entrée `[12,3,4,9], 1.5` le résultat est `3`.
3. Écrivez une classe `Personne` contenant un attribut nom et un attribut age. Écrivez une méthode qui prend en paramètre une liste de personnes et renvoyant l'écart d'âge minimum entre deux personnes de la liste. .. raw:: latex

newpage