

EXERCISE 1)

Un fichier Salut.java contient le texte suivant

```
Class Salut {  
  
    static public void main(String args[]) {  
  
        System.out.println("Bonjour");  
  
    }  
  
};
```

Lesquelles des lignes commandes suivantes sont incorrectes et pourquoi ?

1. javac Salut.java
2. Java Salut.java
3. Javac Salut.class
4. java Salut.class
5. java Salut

-- Reponse --

- **ligne 2. Java est une commande pour exécuter, et .java est un fichier source**
- **ligne 3. Inversement, javac est une commande de compilation et .class est un bytecode**
- **ligne 4. Pas besoin d'indiquer l'extension .class**

EXERCISE 2)

On voudrait définir un objet vecteur de réels de dimension donnée.
Les opérations qu'on voudrait effectuer, entre autres, sont

- Créer un vecteur origine de dimension n donnée
- Affecter une valeur x au ième composant, i et x données
- Accéder au ième composant, donnée i, résultat x, ce ième composant
- Remplir le vecteur avec la même valeur donnée d
- Produit scalaire de deux vecteurs.

1- Définir une Interface pour un tel objet vecteur.

```
interface Vecteur {  
    public void origine(int n);  
        // Crée un vecteur origine de dimension n  
    public void set(int i, double x) ;  
        // met ieme composant à x (0≤i)  
    public double get (int i);  
        // retourne le ieme composant  
    public void setConst (int d);  
        // met tous les composants à d  
    public double produitScalaire (Vecteur r) ;  
        // produit scalaire de deux vecteurs (this et r)  
}
```

2. Implémenter cette interface vecteur

On négligera ici les cas d'erreurs : indice i hors des bornes, n négatif etc. Elles devront faire l'objet de levée d'exception.

```
class Tableau implements Vecteur {
    double [] t;

    public void origine(int n) {
        t = new double [n] ;
        setConst(0);
    }

    public void set(int i, double x){
        t[i] = x;
    }

    public double get (int i){
        return t[i];
    }

    public void setConst (int d){
        for(int i = 0; i< t.length; i++)
            t[i] = d;
    }

    public double produitScalaire (Vecteur r){
        double ps = 0;
        for(int i = 0; i< t.length; i++) {
            ps += t[i]*r.get(i);
        }
        return ps;
        // Tester si même dimension
    }
};
```

3. Ecrire un programme main qui crée les deux vecteurs de trois dimensions v1 (1, 3, 2) et v2 (4, 4, 4) et calcule leur produit scalaire.

```
class TestVecteur {  
    public static void main(String [] args) {  
  
        Vecteur v1 = new Tableau(),  
                v2 = new Tableau();  
  
        v1.origine(3);  
        v1.set (0,1);  
        v1.set (1,3);  
        v1.set (2,2);  
  
        v2.origine(3);  
        v2.setConst(4);  
  
        System.out.println(v1.produitScalaire(v2));  
    }  
}
```