

---

**Objectifs :** références, instances (objets), communiquer des objets à une méthode, méthodes  
retournant des objets, objets composés, chaînes de caractères, surcharge, membres statiques

---

## **Exercice 1**

i ) Soit la déclaration de la méthode suivante :

```
public void test(int a) { }
```

Quelles sont les surcharges admises entre ces différentes possibilités ?

```
public void test() { }  
public void test(double a) { }  
public void test(int a, int b) { }  
public int test(int a) { }
```

ii) Le fichier Java, A.java, suivant ne compilera pas. Expliquez pourquoi.

```
public class A {  
    void A(int i) {  
        System.out.println("Hello");  
    }  
    public static void main(String[] args) {  
        A unA = new A(5);  
    }  
}
```

iii) Le code suivant est-il correct ? sinon identifier les erreurs puis corriger-les.

```
//classe Test  
class Test {  
    public int a;  
    public int c;  
    public Test (int b) {  
        this.a = b;  
    }  
    public static int donneC() {  
        return this.c;  
    }  
}  
  
//classe PrincipalTest  
public class PrincipalTest {  
    public static void main(String[] args) {  
        Test unTest = new Test();  
    }  
}
```

## Exercice 2

Réalisez en Java , un programme contenant une classe `Point` , avec ses trois coordonnées dans l'espace x,y,z, et que l'on peut initialiser de trois manières différentes (selon les valeurs initiales connues des trois coordonnées, on connaît soit x, soit x et y , soit x et y et z). Ensuite, intégrez dans la classe une méthode `translate()` qui est surchargée trois fois, dépendant également desquelles des trois valeurs des translations sont connues.

## Exercice 3

Parmi les attributs suivants de la classe «Renault Kangoo », la version avec toutes les options possibles, séparez ceux que vous déclareriez comme statiques des autres : vitesse, nombre de passagers, vitesse maximale, nombre de vitesses, capacité du réservoir, âge, puissance, prix, couleur, nombre de portières.

## Exercice 4

Sachant que vous disposez de la classe *Processeur* rendant les services suivants :

Constructor Summary	
<a href="#"><u>Processeur</u></a> ()	initialise un nouvel objet de type <i>Processeur</i> , sans marque (chaîne vide "") et avec une fréquence nulle (0).
<a href="#"><u>Processeur</u></a> (String m,double f)	initialise un nouvel objet de type <i>Processeur</i> de marque <i>m</i> et de fréquence <i>f</i>
Method Summary	
double	<a href="#"><u>GetFrequence</u></a> () retourne la fréquence du processeur courant
String	<a href="#"><u>getMarque</u></a> () retourne la marque du processeur courant
String	<a href="#"><u>toString</u></a> () renvoie la chaîne de caractères représentant le processeur courant
void	<a href="#"><u>init</u></a> () initialise interactivement le processeur courant

et de la classe *Ecran* rendant les services suivants :

Constructor Summary	
<a href="#"><u>Ecran</u></a> ()	initialise un nouvel objet de type <i>Ecran</i> sans marque (") et de taille nulle (0)
<a href="#"><u>Ecran</u></a> (String m, int t)	initialise un nouvel objet de type <i>Ecran</i> de marque <i>m</i> et de taille <i>t</i> (en pouces)
Method Summary	
String	<a href="#"><u>getMarque</u></a> () retourne la marque de l'écran courant
void	<a href="#"><u>init</u></a> () initialise interactivement l'écran courant
double	<a href="#"><u>getTaille</u></a> () retourne la taille de l'écran courant
String	<a href="#"><u>toString</u></a> () renvoie la chaîne de caractères représentant l'écran courant

écrire une classe *Ordinateur* (et une classe *TestOrdinateur*) sachant qu'ici un ordinateur est simplement caractérisé par son écran et son processeur. La classe *Ordinateur* rendra les services suivants:

1. un constructeur qui initialise un nouvel objet de type *Ordinateur* avec un écran sans marque et de taille nulle , et un processeur sans marque et de fréquence nulle
  2. un constructeur qui initialise un nouvel objet de type *Ordinateur* avec un écran *e* et un processeur *p* (ne pas faire de copie de ces objets)
  3. une méthode qui initialise interactivement l'écran et le processeur de l'ordinateur courant
  4. méthodes d'accès en lecture/écriture pour les variables d'instances
  5. une méthode qui renvoie la chaîne de caractères représentant l'ordinateur courant
- 
- tester les 2 constructeurs en créant 2 ordinateurs *o1* et *o2* (*o1* sera construite par défaut et initialisée interactivement)
  - afficher les caractéristiques des deux instances
  - remplacer le processeur de *o1* par un autre processeur
  - afficher de nouveau les caractéristiques de *o1* pour vérifier le bon remplacement de son processeur
  - échanger les deux écrans des deux ordinateurs *o1* et *o2*
  - afficher de nouveau les caractéristiques des deux ordinateurs *o1* et *o2* pour vérifier l'échange.