

**Examen Programmation Orientée Objet, Java**  
(1ère session)

**Aucun document autorisé**

**Exercice 1 – Questions de cours**

Répondez par Vrai ou par Faux aux questions suivantes :

1. Dès qu'un constructeur est défini de manière explicite, le constructeur par défaut (constructeur sans paramètres) ne peut plus être implicitement défini.
2. Le constructeur sans paramètres de la classe *Object* s'exécute avant tous les autres constructeurs sans afficher de message
3. Quand il n'y a pas d'appel explicite à un constructeur de la super-classe, le constructeur sans paramètres de la super-classe est implicitement appelé
4. Une méthode de classe peut être redéfinie dans les héritiers de la classe
5. Une interface doit contenir au moins une méthode abstraite
6. Une interface peut hériter d'une ou de plusieurs interfaces
7. Une classe abstraite est une classe qui contient au plus une méthode abstraite
8. Les instructions contenues dans un bloc *finally* ne sont exécutées qu'à la suite d'une exception
9. Une classe ne peut implémenter plusieurs interfaces
10. Toute classe qui implémente une interface est sous type de cette interface

**Exercice 2 – Analyse et correction de code**

Corrigez si nécessaire, les morceaux de code ci-dessous, en donnant une courte (maximum deux lignes) explication (même si vous estimez que le code est correct).

```
public class A {  
    public A(int valeur) {  
        System.out.println( "Dans le constructeur de la classe A : " + valeur);  
    }  
}  
public class B extends A {  
    public B(){  
        System.out.println( "Dans le constructeur de la classe B");  
    }  
}
```

```
public class TabToVector  
{  
    public static List<Double> transferer (double[] tab)  
    {  
        List<Double> tableau = new Vector<Double>()  
        for (int i = 0; i < tab.length; i++){  
            tableau.add(tab[i]);  
        }  
        return tableau;  
    }  
}
```

```
public class ChaîneToTab
{
//chaîne contient une série de 10 entiers séparés par '*'
//Exemple: 123*12*98*57*66*3*645*7*213*52
    public static int[] transferer (String chaîne)
    {
        StringTokenizer st = new StringTokenizer(chaîne, "*");
        int[] tableau = new int[10];
        int i =0;
        while(st.hasMoreTokens()){
            tableau[i]=st.nextToken();
            i++;
        }
        return tableau;
    }
}
```

```
public class Comparer {
    public static void main(String[] args) {
        if(args[0] != args[1])
            System.out.println(args[1] + " est différent de " + args[0]);
    }
}
```

### Exercice 3 – programmation

Il s'agit d'écrire un petit programme Java pour permettre de travailler sur des polynômes. Un polynôme d'une variable est considéré comme une liste (par exemple ArrayList) de monôme (on a donc une forme particulière de liste). Chaque monôme est caractérisé par un coefficient et un exposant. Prenons comme exemple le polynôme suivant :  $P(x) = 3x^2 + 2.5x + 4$

#### Question1 : Ecrire la classe *Monome*.

La classe *Monome* possède deux attributs (*coefficient* de type double et *exposant* de type int).

Elle doit redéfinir la méthode *toString()* pour fournir une chaîne de caractères représentant les caractéristiques du monôme. (Exemple : "3x\*\*2").

Elle doit également fournir une méthode publique nommée *valeurMonome* pour calculer la valeur du monôme pour une valeur v donnée de x. (double valeurMonome(double v))

#### Question 2 : La classe *Polynome*

Chaque polynôme est une liste (particulière) de monômes à laquelle on souhaite ajouter un nom. Ecrivez la classe Polynôme, elle doit proposer les méthodes suivantes :

*public void ecrirePolynome()* : affiche à l'écran le polynôme comme par exemple  
"Le Polynôme nommé P1 : 3x\*\*2 + 2.5x + 4.0"

*public double valeurPolynome (double v)* : calcule la valeur du polynôme pour x égal à v  
Pour cette deuxième méthode vous devez obligatoirement utiliser l'interface *Iterator*

**N.B. Méthode utile : double Math.pow(double x, double exposant)**