

UEF4.3. Programmation Orientée Objet**COTROLE INTERMEDIARE**

2 heures - Documents Interdits

Exercice 1 (4pts)

Le programme Java ci-dessous a été écrit par un débutant en POO.

1. Que pensez-vous de ce code ? Expliquez.
2. Proposez une version améliorée.
3. Donnez les commandes nécessaires pour compiler et exécuter ce code en mode console pour les valeurs 2 et 3.

```
public class intsum{

    public static void main(String args[]){
        int suma = Integer.parseInt(args[0]) ;
        int sumb = Integer.parseInt(args[1]) ;
        int sum = suma+sumb;
        int diff = suma-sumb;
        int prod = suma*sumb;
        double avg =(suma+sumb)/2;

        System.out.println("La somme des deux nombres est " + sum);
        System.out.println("La différence des deux nombres est " + diff);
        System.out.println("Le produit des deux nombres est " + prod);
        System.out.println("La moyenne des deux nombres est " + avg);

        if(suma>sumb)    System.out.println("Le nombre le plus grand est " +suma);
        else System.out.println("Le nombre le plus grand est " +sumb);

        if(suma<sumb)    System.out.println("Le nombre le plus petit est " +suma);
        else System.out.println("The smaller of bothnumbersis " +sumb);
    }
}
```

Exercice 3 (8,5pts)

Le programme suivant comporte des erreurs. Indiquez toutes les erreurs, quels que soient leurs types, en fournissant une explication pour chacune d'elles, puis proposez une correction. Répondez dans un tableau suivant le modèle ci-dessous.

N° de ligne	Erreur	Correction

```

1.  import java.util.*;
2.  package P1;
3.  public class CP1 {
4.      HashMap<A, CP2>ens = new HashMap<A, CP2>();
5.      public static void main(String[] args){
6.          A a = newA(3);
7.          B b = newB();
8.          C c = newC();
9.          ens.put(a, new CP2(a.ax));
10.         ens.put(b, new CP2(b.ax));
11.         ens.put(c, newCP2(c.ax));
12.         B.In i = newB.In();
13.         try {
14.             b.setX(2);
15.         }
16.         c.z = 3;
17.         int k,j ;
18.         k=4 ;
19.         j=5 ;
20.         c = b;
21.         c.m(k,j) ;
22.     }
23. }
24. abstract class A {
25.     int ax=3;
26.     private A(int ax){
27.         this.ax = ax;
28.     }
29. }
30. final class B extends A {
31.     private int x;
32.     classIn{ }
33.     setX(int x) throws ArithmeticException {
34.         if (x>0) this.x=x ;
35.         else throw new ArithmeticException() ;
36.     }
37.     public int m(int v) {
38.         return (v+x) ;
39.     }
40. }
41. class C extends B {
42.     final double z = 1;
43.     public void m(int v) {
44.         System.out.println(v) ;
45.     }
46.     public int m(int v, int h) {
47.         return (v*h+x);
48.     }
49. }
50. abstract class D extends A,I {
51.     double d;
52.     int operation(int a) {
53.         return (a * 2);
54.     }
55.     abstract int calcul(int b) {
56.         return (b*a);
57.     }
58.     abstract void afficher();
59. }
60. interface I {
61.     public int operation(int a) {
62.         return (a * 2);
63.     }
64. }

```

Fichier : CP1.java


```

65. package P2 ;
66. class CP2 {
67.     private int a ;
68.     private int xc ;
69.     private static int cpt = 0 ;
70.     public void CP2(int a) {
71.         this.a = a ;
72.         this.xc = xc+cpt ;
73.         cpt++ ;
74.     }
75. }

```

Fichier : CP2.java

Exercice 4 (7,5pts)*Lisez entièrement l'énoncé avant de répondre*

Nous souhaitons définir une classe *Polynome* destinée à contenir un ensemble de monômes (termes) triés de sorte qu'il soit impossible d'insérer deux fois le même terme dans cet ensemble. **Exemple :** on peut insérer $2x^3$ puis $7x^3$ mais on ne peut pas insérer deux fois le monôme $2x^3$. Pour cela, on décide d'utiliser un type de collection Java. Voici le squelette de cette classe :

```

... Polynome {
    private ..contenu = new .. ;
    public abstract String concat ();
    public void inserer(Monome o){ ... }
    public void supprimer(Monome o)... { ... }
    public String toString(){ ... }
    public int taille() { ... }
}

```

1. Quelle collection proposez-vous pour conserver les éléments de l'ensemble (attribut *contenu*) ?
2. Définissez les méthodes *inserer* et *taille* de cette classe.
3. On veut que la méthode *supprimer* retourne une erreur de type *EnsembleVideErreur* si l'ensemble est vide. Implémentez cette méthode en utilisant les exceptions.
4. Redéfinissez la méthode *toString* de la superclasse *Object* afin que les objets qui se trouvent dans l'ensemble soient affichés de la manière suivante: $elt_1 + elt_2 + \dots + elt_n$ par ordre décroissant des puissances des termes.
5. Complétez le code manquant (hormis les méthodes déjà définies) de la classe *Polynome*.
6. Définissez les classes *Monome* et *Polynome3D* de manière à ce que le code suivant:

```

Polynomee = new Polynome3D () ;
e.inserer(new Monome(2,3));
e.inserer(new Monome(5,2));
e.inserer(new Monome(4,1));
e.inserer(new Monome(7));
System.out.println(e.toString());

```

Produise l'affichage suivant: $2x^3 + 5x^2 + 4x + 7$ (qui signifie $2x^3 + 5x^2 + 4x + 7$)

7. Définissez la méthode `public String concat()` qui retourne une chaîne obtenue par concaténation de tous les éléments de l'ensemble.

Exemple : `System.out.println(e.concat());`

Va produire: $2x^3 + 5x^2 + 4x + 7$

8. Tracez le diagramme de classe correspondant à cet exercice.