

Exercice 1 : (5pts)

1. Citer les différents indicateurs de visibilité (accessibilité)

public , protected, private, par défaut (aucun indicateur)

2. Que signifie le mot « static » devant un attribut d'une classe ?

C'est une variable de classe qui est partagée entre toutes les instances d'une même classe

3. Que signifie le mot « static » devant une méthode d'une classe ?

La méthode appartient non pas à l'instance mais à la classe contenant la méthode

4. Que signifie le mot « final » devant une méthode d'une classe ?

Une méthode final ne peut être redéfinie (overridden) dans une classe fille

5. Vérifier si les deux codes suivants contiennent des erreurs ou s'exécutent correctement et pour chaque erreur indiquer sa ligne et proposer une correction.

<pre>// code 1 1. public int sAccount(int[25] number){ 2. number = new int[15]; 3. for (int i=0;i <=number.length;i++) 4. number[i]=number[i-1]+number[i+1]; 5. return number; }</pre>	<pre>1. public int []sAccount(int[] number){ 3. for (int i=1;i <number.Length-1;i++)</pre>
<pre>// code 2 2. public class A{ 1. int a; 2. A(int x){a=x;} 3. A(char x){a=x;} 4. A(double b){a=b;} 5. A(Double b){this();} }</pre>	<pre>4. A(double b){a= (int)b;} 5.A(Double b){this((double)b);</pre>

Exercice 2 (7pts):

1. Créez une classe Liquide contenant seulement une méthode afficher() qui affiche : "I am a liquide !". Un liquide est caractérisé par une couleur.

```
public class Liquide {  
    private String couleur;  
    Liquide() {couleur="white";}   
    Liquide(String c) {couleur=c;}   
    String getCouleur() {return couleur;}   
    void setCouleur(String c) {couleur = c;}   
    void afficher (){System.out.print("I am a liquid ! ");}  
}
```

2. Créez 3 classes dérivées de la classe Liquide, les classes Cafe, Milk, et Tea, dont les méthodes afficher() affichent respectivement "I am a liquide ! I'm a Brown Cafe", "I am a liquide ! I'm a White milk", " I am a liquide ! I'm a Green tea"

```
class Cafe extends Liquide{  
    Cafe(){super("Brown");}  
    void afficher (){super.afficher();  
        System.out.println("I am a Brown Cafe ");}  
}  
class Milk extends Liquide{  
    Milk(){super("White");}  
    void afficher (){super.afficher();  
        System.out.println("I am a White Milk ");}  
}  
class Tea extends Liquide{  
    Tea(){super("Green");}  
    void afficher (){super.afficher();  
        System.out.println("I am a Green Tea ");}  
}
```

3. Créez une classe Tasse qui contient un Liquide. La classe Tasse a une méthode AjouterLiquide pour ajouter un liquide et une méthode imprimer().

```
class Tasse {  
    Liquide L;  
    void AjouterLiquide (Liquide l){L=l;}  
    void print(){L.afficher();}  
}
```

4. Testez l'utilisation : créer et afficher un tableau de Tasses qui contient 3 différents liquides, de sorte que le programme affiche ce qui suit:

```
I am a liquid ! I am a Brown Cafe  
I am a liquid ! I am a White Milk  
I am a liquid ! I am a Green Tea
```

```
Tasse T[]=new Tasse[3];
T[0]=new Tasse(); T[0].AjouterLiquide(new Cafe());
T[1]=new Tasse(); T[1].AjouterLiquide(new Milk());
T[2]=new Tasse(); T[2].AjouterLiquide(new Tea());
T[0].print(); //ou utiliser une boucle for pour afficher les éléments du tableau
T[1].print();
T[2].print();
```

Exercice 3(8pts):

1. *Ecrire une classe Complexe permettant de représenter des nombres complexes. Un nombre complexe est un nombre qui comporte deux parties: une partie réelle et une partie imaginaire. La classe Complexe comporte trois méthodes: plus (qui renvoie la somme du nombre complexe courant et un autre nombre complexe), moins (qui renvoie la soustraction du nombre complexe courant et un autre nombre complexe) et print (pour afficher le nombre complexe courant)*

```
package exam2023;
public class Complexe {
    private double real, image;
    //Constructors' code here
    Complexe(){real=1;image=1;}
    Complexe(double a, double b){real=a;image=b;}
    //Getters and setter's code here
    public double getReal() {return real;}
    public double getImage() {return image;}
    public void setReal(double r) {real = r;}
    public void setImage(double i) {image = i;}

    //plus methode code here
    Complexe plus(Complexe a){return new Complexe(a.real+real,a.image+image);}

    //moins methode code here
    Complexe moins(Complexe a){return new Complexe(a.real-real,a.image-image);}

    //print methode code here
    void print(){System.out.println(real+"i"+image);}
}
```

2. Ecrire la classe Polynome2D pour représenter un polynôme de deuxième degré (كثير حدود درجة 2) en découpant le code et les méthodes en sous méthodes auxiliaires dans tous les cas possibles.

La classe Polynome2D comporte principalement les méthodes suivantes :

- print : pour afficher le polynôme
- ResoudreR: Pour trouver et afficher les zéros du polynôme dans l'ensemble des nombres réels R.
- ResoudreC: pour trouver et afficher les zéros du polynôme dans l'ensemble des nombres complexes C.

Rappelle : $\Delta = b^2 - 4 * a * c$, $x1 = \frac{-b-\sqrt{\Delta}}{2*a}$, $x1 = \frac{-b+\sqrt{\Delta}}{2*a}$, $z1 = \frac{-b-i\sqrt{|\Delta|}}{2*a}$, $z2 = \frac{-b+i\sqrt{|\Delta|}}{2*a}$

Remarque : pour simplifier on suppose que les coefficients du polynôme sont différents à 0.

```
package exam2023;

public class Polynome2D{

    private double a,b,c;

    //Constructors code here

    Polynome2D(){a=1;b=1;c=1;}

    Polynome2D(double x,double y,double z){a=x;b=y;c=z;}

    // Getters and setters'code here

    double geta(){return a;}    void seta(double x){a=x;}

    double getb(){return b;}    void setb(double x){b=x;}

    double getc(){return c;}    void setc(double x){c=x;}

    // The other methods codes here

    double delta(){return (b*b-4*a*c);}

    double x1(){return ((-b-Math.sqrt(delta()))/(2*a));}

    double x2(){return ((-b+Math.sqrt(delta()))/(2*a));}

    void resoudreR(){
        if(delta())>=0) printSolutionR();
        else System.out.println("Pas de solution");
    }

    void resoudreC(){
        if(delta())>=0) printSolutionR();
        else {ComplexSolution();}
    }
}
```

```
void ComplexSolution(){
    Complexe x1=new Complexe(-b/2*a,-Math.sqrt(Math.abs(delta()))/(2*a));
    Complexe x2=new Complexe(-b/2*a,Math.sqrt(Math.abs(delta()))/(2*a));
    x1.print();x2.print();
}

void printSolutionR(){System.out.println("Two solutions : "+x1()+" "+x2());}

void print(){System.out.println(a+"x^2 + "+b+"x + "+c);}
}
}
```

3. On veut identifier chaque instance de la classe Polynome2D par un identifiant unique. Proposer une solution :

```
int id, ; int static i =0;
```

Dans les constructeurs on incrémente i par 1 et on affecte la valeur à id

```
(i=i+1 ; id=i ;)
```