

Devoir N°2

Exercice 1 :

```
public class Point {
    private double x,y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public Point() {
        this(0,0);
    }

    public double getX() {
        return x;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getY() {
        return y;
    }

    public void setY(double y) {
        this.y = y;
    }
    @Override
    public String toString(){
        return "("+x+","+y+" ";
    }
    public double calculerDistance(Point p){
return Math.sqrt(Math.pow((this.x - p.x) ,2)) + (Math.pow((this.y - p.y) ,2));
    }

    public Point calculerMilieu(Point p){
        Point pt = new Point ((this.x+p.x)/2 , (this.y+p.y)/2);
        return pt;
    }
}
```

```

public class TroisPoints {
    private Point premier,deuxieme,troisieme;

    public TroisPoints(Point premier, Point deuxieme, Point troisieme) {

        this.premier = new Point(premier.getX(),premier.getY());
        this.deuxieme = new Point(deuxieme.getX(),deuxieme.getY());
        this.troisieme = new Point(troisieme.getX(),troisieme.getY());
    }

    public Point getPremier() {
        return premier;
    }

    public void setPremier(Point premier) {
        this.premier = premier;
    }

    public Point getDeuxieme() {
        return deuxieme;
    }

    public void setDeuxieme(Point deuxieme) {
        this.deuxieme = deuxieme;
    }

    public Point getTroisieme() {
        return troisieme;
    }

    public void setTroisieme(Point troisieme) {
        this.troisieme = troisieme;
    }

    public boolean sontAlignes(){
        if (premier.calculerDistance(deuxieme) ==
premier.calculerDistance(troisieme) + deuxieme.calculerDistance(troisieme)
|| premier.calculerDistance(troisieme) == premier.calculerDistance(deuxieme)
+ deuxieme.calculerDistance(troisieme) ||
deuxieme.calculerDistance(troisieme) == premier.calculerDistance(troisieme) +
premier.calculerDistance(deuxieme))
            return true;
        return false;
    }

    public boolean estIsocele(){
        if (premier.calculerDistance(deuxieme) ==
premier.calculerDistance(troisieme) || premier.calculerDistance(deuxieme)
== deuxieme.calculerDistance(troisieme) ||
deuxieme.calculerDistance(troisieme) == premier.calculerDistance(troisieme))
            return true;
        return false;
    }

}

```

```

public class ClasseDeTest {
    static Scanner s = new Scanner(System.in);
    public static void main(String []args){
        Point p1,p2,p3;
        double x,y;
        System.out.print("Entrez les coordonnées de la premier point ");
        x = s.nextDouble();
        y = s.nextDouble();
        p1 = new Point(x,y);

        System.out.print("Entrez les coordonnées de la deuxième point ");
        x = s.nextDouble();
        y = s.nextDouble();
        p2 = new Point(x,y);

        System.out.print("Entrez les coordonnées de la troisième point ");
        x = s.nextDouble();
        y = s.nextDouble();
        p3 = new Point(x,y);

        System.out.println("la premier point : "+p1.toString()+"\nla deuxième point : 
        "+p2.toString()+"\nla troisième point : "+p3.toString()+"\n");

        System.out.println("La distance entre p1 et p2 : 
        "+p1.calculerDistance(p2)+"\nLa distance entre p1 et p3 : 
        "+p1.calculerDistance(p3)+"\nLa distance entre p2 et p3 : 
        "+p2.calculerDistance(p3)+"\n");

        System.out.println("Le milieu de p1 et p2 : "+p1.calculerMilieu(p2)+"\nLe 
        milieu de p1 et p3 : "+p1.calculerMilieu(p3)+"\nLe milieu de p2 et p3 : 
        "+p2.calculerMilieu(p3)+"\n");

        TroisPoints trinome = new TroisPoints (p1,p2,p3);
        if (trinome.sontAlignes())
            System.out.println("Les points p1,p2 et p3 sont alignés");
        else
            System.out.println("Les points p1,p2 et p3 ne sont pas alignés");

        if (trinome.estIsocele())
            System.out.print("Les points p1,p2 et p3 forment un triangle isocèle");
        else
            System.out.print("Les points p1,p2 et p3 ne forment pas un triangle isocèle") ;
    }
}

```

Exercise 2 :

```
public class Complexe {
    private float re,im;

    public Complexe(float re, float im) {
        this.re = re;
        this.im = im;
    }

    public Complexe(){
        this(0,0);
    }
    public Complexe(Complexe c){
        this.re = c.re;
        this.im = c.im;
    }

    @Override
    public String toString() {
        return re+" + "+im+"i";
    }

    public float getRe() {
        return re;
    }

    public void setRe(float re) {
        this.re = re;
    }

    public float getIm() {
        return im;
    }

    public void setIm(float im) {
        this.im = im;
    }
    public Complexe addition(Complexe c){
        return new Complexe(this.re + c.re, this.im + c.im);
    }

    public Complexe soustraction(Complexe c){
        return new Complexe(this.re - c.re, this.im - c.im);
    }
    public Complexe multi(Complexe c){
        return new Complexe((this.re*c.re) - (this.im*c.im) ,
(this.re*c.im)+(c.re*this.im));
    }
    public Complexe division(Complexe c){
        float res = (float) (Math.pow(c.re, 2) + Math.pow(c.im, 2));
        return new Complexe( ((this.re*c.re) + (this.im*c.im))/res , ((this.im*c.re)-
(c.im*this.re))/res );
    }
    public static Complexe additionner(Complexe a , Complexe b){
        return new Complexe (a.re + b.re , a.im + b.im);
    }
    public Complexe multipNbr(int a){
        return new Complexe(a*this.re, a*this.im);
    }
    @Override
    public boolean equals(Object ob){
        Complexe c = (Complexe)ob;
        return (this.re == c.re) && (this.im == c.im);
    }
}
```

```

public double argument() {
    double z = Math.atan(this.im/this.re);
    if (this.re>0 && this.im>0)
        return z;
    else if (this.re<0 && this.im<0)
        return 3.14 + z;
    else if ( this.re>0 && this.im<0 )
        return -z;
    else if (this.re<0 && this.im>0)
        return 3.14 - z;
    else
        return 0 ;
}

public Complexe conjugue(){
    return new Complexe (this.re, -this.im);
}

public Complexe inverse(){
    float res = (float) (Math.pow(this.re, 2) + Math.pow(this.im, 2));
    return new Complexe(this.re/res , -this.im/res);
}
}

public class ClassDeTest {
    static Scanner s = new Scanner(System.in);
    public static void main(String args[]){
        Complexe c1,c2;
        float x,y;
        System.out.print("Entrez la partie réel de la premier nombre complexe : ");
        x = s.nextFloat();
        System.out.print("Entrez la partie sa imaginaire : ");
        y = s.nextFloat();
        c1 = new Complexe(x,y);

        System.out.print("Entrez la partie réel de la deuxième nombre complexe : ");
        x = s.nextFloat();
        System.out.print("Entrez la partie sa imaginaire : ");
        y = s.nextFloat();
        c2 = new Complexe(x,y);

        System.out.println("Le nombre complexe 1 : "+c1.toString()+"\nLe nombre complexe 2 : "+c2.toString()+"\n");

        System.out.println("L'addition de "+c1.toString()+" et "+c2.toString()+" est : "+c1.addition(c2));
        System.out.println("La soustraction de "+c1.toString()+" et "+c2.toString()+" est : "+c1.soustraction(c2));
        System.out.println("La multiplication de "+c1.toString()+" par "+c2.toString()+" est : "+c1.multi(c2));
        System.out.print("Entrez un entier : ");
        int a = s.nextInt();
        System.out.println("La multiplication de "+c1.toString()+" par "+a+" est : "+c1.multipNbr(a));
        System.out.println("La multiplication de "+c2.toString()+" par "+a+" est : "+c2.multipNbr(a));
        //L'addition de deux nombres complexes on utilisant la méthode static additionner
        System.out.println(Complexe.additionner(c1, c2).toString());

        System.out.println("La division de "+c1.toString()+" par "+c2.toString()+" est : "+c1.division(c2));
        System.out.println("Le module de "+c1.toString()+" est : "+c1.module()+" et de "+c2.toString()+" est : "+c2.module());
        System.out.println("L'argument de "+c1.toString()+" est : "+c1.argument()+" et de : "+c2.toString()+" est : "+c2.argument());
        System.out.println("Le conjugué de "+c1.toString()+" est : "+c1.conjuge()+" et de : "+c2.toString()+" est : "+c2.conjuge());
        System.out.println("L'inverse de "+c1.toString()+" est : "+c1.inverse()+" et de : "+c2.toString()+" est : "+c2.inverse());
    }
}

```

```

public class Equation {
private float a,b,c,delta;

public Equation(float a, float b, float c) {
    this.a = a;
    this.b = b;
    this.c = c;
    delta = (float) (Math.pow(b, 2) - 4*a*c);
}

public void afficherDiscriminant(){
    System.out.print("Le discriminant delta est : "+delta);
}

public void résout(){
    Complexe z1,z2;
    if (delta <0){
        z1 = new Complexe (-b/2*a, (float)-Math.sqrt(-delta)/2*a);
        z2 = new Complexe (-b/2*a, (float)Math.sqrt(-delta)/2*a);
        System.out.print("Les racines de l'équation sont : "+z1+" et "+z2);
    }
    else
        System.out.print("Pas de solution dans C uniquement dans R");
}

public void afficherSolutions(){
    System.out.print( " (Acos("+(float)-Math.sqrt(-delta)/2*a+"x) +
    Bsin("+(float)-Math.sqrt(-delta)/2*a+"x))e^("+-b/2*a+"x) " );
}
}

public class ClasseDeTest2 {
    static Scanner s = new Scanner(System.in);
    public static void main(String args[]){
        Equation e;
        float a,b,c;
        System.out.print("Entrez la valeur de la premier entier : ");
        a = s.nextFloat();
        System.out.print("Entrez la valeur de la deuxième entier : ");
        b = s.nextFloat();
        System.out.print("Entrez la valeur de la troisième entier : ");
        c = s.nextFloat();
        e = new Equation(a,b,c);

        e.afficherDiscriminant();
        e.résout();
        System.out.print("La solution de l'équation : " );
        e.afficherSolutions();
    }
}

```

Exercise 3:

```
public class Stagiaire {
    private String nom;
    private float []notes;
    public Stagiaire(String nom, float[] notes) {
        this.nom = nom;
        this.notes = notes;
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public float[] getNotes() {
        return notes;
    }
    public void setNotes(float[] notes) {
        this.notes = notes;
    }
    public float calculerMoyenne(){
        float som=0;
        for(int i=0;i<notes.length;i++)
            som = som + notes[i];

        return som/notes.length;
    }
    public float trouverMax(){
        float max = notes[0];
        for (int i=0;i<notes.length;i++)
            if (notes[i]>max)
                max = notes[i];

        return max;
    }
    public float trouverMin(){
        float min = notes[0];
        for (int i=0;i<notes.length;i++)
            if (notes[i]<min)
                min = notes[i];

        return min;
    }
}
```

```

public class Formation {
private String intitule;
private int nbrJours;
private Stagiaire stagiaire[];
public Formation(String intitule, int nbrJours, Stagiaire[] stagiaire) {
    this.intitule = intitule;
    this.nbrJours = nbrJours;
    this.stagiaire = stagiaire;
}
public String getIntitule() {
    return intitule;
}
public void setIntitule(String intitule) {
    this.intitule = intitule;
}
public int getNbrJours() {
    return nbrJours;
}
public void setNbrJours(int nbrJours) {
    this.nbrJours = nbrJours;
}
public Stagiaire[] getStagiaire() {
    return stagiaire;
}
public void setStagiaire(Stagiaire[] stagiaire) {
    this.stagiaire = stagiaire;
}
public float calculerMoyenneFormation(){
float somMoy=0;
for(int j =0;j<stagiaire.length;j++)
    somMoy = somMoy + stagiaire[j].calculerMoyenne();

return somMoy/stagiaire.length;
}
public int getIndexMax(){
    int max = 0;
    for(int k=0;k<stagiaire.length;k++)
        if (stagiaire[k].calculerMoyenne() > stagiaire[max].calculerMoyenne() )
            max = k;

    return ++max;
}

public void afficherNomMax(){
    int i = getIndexMax();
    System.out.println( stagiaire[--i].getNom());
}
public void afficherMinMax(){
    int i = getIndexMax();
    System.out.println(stagiaire[--i].trouverMin());
}
public void trouverMoyenneParNom(String nom){
    for(int j=0;j<stagiaire.length;j++)
        if (stagiaire[j].getNom().equals(nom)) {
            System.out.print("La moyenne du stagiaire "+nom+" est :
"+stagiaire[j].calculerMoyenne());
            break;
        }
}
}
}

```



```

public class ClasseDeTest {
    public static void main(String []args){
        Stagiaire s1,s2,s3;
        Formation f;
        float n1[] = new float[5];
        n1[0] = (float) 13.85;
        n1[1] = (float) 15.85;
        n1[2] = (float) 16.85;
        n1[3] = (float) 17;
        n1[4] = (float) 10;

        float n2[] = new float[5];
        n2[0] = (float) 10.85;
        n2[1] = (float) 15;
        n2[2] = (float) 18.75;
        n2[3] = (float) 12.2;
        n2[4] = (float) 14.5;

        float n3[] = new float[5];
        n3[0] = (float) 19.85;
        n3[1] = (float) 13;
        n3[2] = (float) 11.75;
        n3[3] = (float) 15.2;
        n3[4] = (float) 10.5;

        s1 = new Stagiaire("Houssam",n1);
        s2 = new Stagiaire("Hajar",n2);
        s3 = new Stagiaire("sohail",n3);

        Stagiaire []s = new Stagiaire[3];
        s[0] = s1; s[1] = s2; s[2] = s3;
        f = new Formation("java",15,s);

        System.out.println("La moyenne de la formation :
"+f.calculerMoyenneFormation());

        System.out.println("Le numéro de stagiaire qui a le meilleur moyenne de
la formation est : "+f.getIndexMax());

        System.out.print("Le stagiaire qui a le meilleur moyenne est : ");
        f.afficherNomMax();

        System.out.print("La note minimale du stagiaire ayant le meilleur
moyenne de la formation est : ");

        f.afficherMinMax();
        f.trouverMoyenneParNom("Hajar");
    }
}

```

Exercice 4:

```
public class Client {
    private long cin;
    private String nom, prénom;
    private CompteBancaire []compteBancaire;
    public Client(long cin, String nom, String prénom,
        CompteBancaire[] compteBancaire) {
        this.cin = cin;
        this.nom = nom;
        this.prénom = prénom;
        this.compteBancaire = compteBancaire;
    }
    @Override
    public String toString() {
        String s = "Client : cin=" + cin + ", nom=" + nom + ", prénom=" + prénom
            + "\ncomptes bancaire : \n";

        for (CompteBancaire c : compteBancaire)
            s += c.toString()+"\n";

        return s;
    }

    public void afficherCompte(int numC) {
        System.out.println("Les informations sur le compte n°"+numC+" :\n Le
découvert : "+compteBancaire[numC].getDécouvert()+"", Le solde :
"+compteBancaire[numC].getSolde());
    }
    public void afficherEnsComptes(Client c){
        System.out.println("Les informations sur les comptes du propriétaire
"+c+" :");
        for(int j=0;j<compteBancaire.length;j++)
            System.out.println(c.compteBancaire[j]);
    }
}

public class CompteBancaire {
    private int numC;
    private static int n=0;
    private float solde;
    private float découvert;
    public CompteBancaire(float solde){
        numC = 1000 + n;
        n++;
        this.solde = solde;
        if (solde>0 || solde == 0)
            découvert = 0;
        else
            découvert = Math.abs(solde);
    }
    public float getSolde() {
        return solde;
    }
    public float getDécouvert() {
        return découvert;
    }
    public int getNumC() {
        return numC;
    }
}
```

```

public int getN() {
    return n;
}

public boolean créditerCompte(float a){
    if (a>0){
        solde += a;
        return true;
    }
    return false;
}

public boolean débiterCompte(float a){
    if (a>0){
        if (solde-a > découvert){
            solde -= a;
            return true;
        }
    }
    return false;
}

public boolean virement(CompteBancaire c , float a){
    c.solde += a;
    this.solde -=a;
    return true;
}

@Override
public String toString() {
    return "numC=" + numC + ", solde=" + solde
        + ", découvert=" + découvert;
}

}

public class ClassDeTest {
    public static void main(String args[]){
        CompteBancaire cb1,cb2,cb3,cb4;
        cb1 = new CompteBancaire(1700);
        cb2 = new CompteBancaire(500);
        cb3 = new CompteBancaire(600);
        cb4 = new CompteBancaire((float) 800.6);
        Client c1,c2;
        CompteBancaire tab[]=new CompteBancaire[2];
        tab[0] = cb1; tab[1] = cb2;
        c1 = new Client (1234,"ait momin","hajar",tab);

        CompteBancaire tab2[]=new CompteBancaire[2];
        tab2[0] = cb3; tab2[1] = cb4;
        c2 = new Client (1834,"Mrabet","Houssam",tab2);

        System.out.print(c1.toString() +"\n"+c2.toString());
    }
}

```