

INFO0201

TD1- introduction à la POO

ANNEE 2020-2021

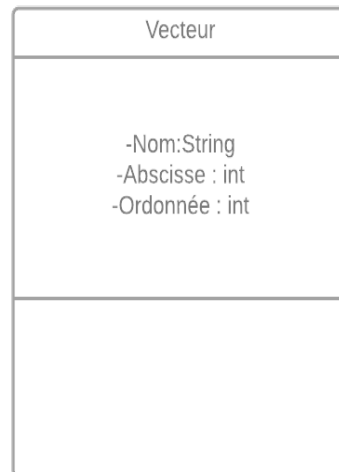
Leeder dominic

Exercice 1 :

1.

Nom attribut	Nom	Abscisse	Ordonnée
Type	String	Int	Int
Niveau d'accès	Privé	Privé	privé

Diagramme de classe correspondant :



2.

```

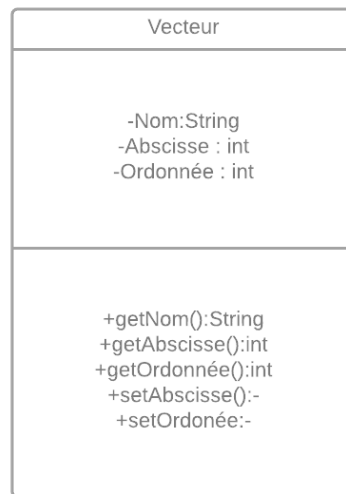
1  class Vecteur{
2      private String Nom;
3      private int Abscisse;
4      private int Ordonnée;
5  }
6
7  //constructeur
8
9  public Vecteur(String n,int a,int o){
10     Nom = n;
11     Abscisse = a;
12     Ordonnée = o;
13 }
    
```

3. Pour les getters :

- getNom()
- getAbscisse()
- getordonnée()

Les attributs pertinents à modifier sont l'abscisse et l'Ordonnée du coup on obtient :

- setAbscisse()
- setOrdonnée()



4. Code complet de la classe Vecteur :

```
class Vecteur{
    private String Nom;
    private int Abscisse;
    private int Ordonnée;
}

//constructeur

public Vecteur(String n,int a,int o){
    Nom = n;
    Abscisse = a;
    Ordonnée = o;
}

//getters
public String getNom(){
    return Nom;
}
public int getAbscisse(){
    return Abscisse;
}
public int Ordonnée(){
    return Ordonnée;
}

//setters

public void setAbscisse(int a){
    Abscisse = a;
}
public void setOrdonnée(int o){
    Ordonnée =o;
}

public String toString(){
    String s ="le vecteur "+Nom+" a pour abscisse : "+Abscisse+" et pour Ordonnée : "+Ordonnée;
    return s;
}
```

Code de la classe TestVecteur :

```

Class TestVecteur{
    private static Scanner clavier = new Scanner(System.in);
    Run | Debug
    public static void main(String[] args){
        String n;int a,o;

        // saisie utilisateur

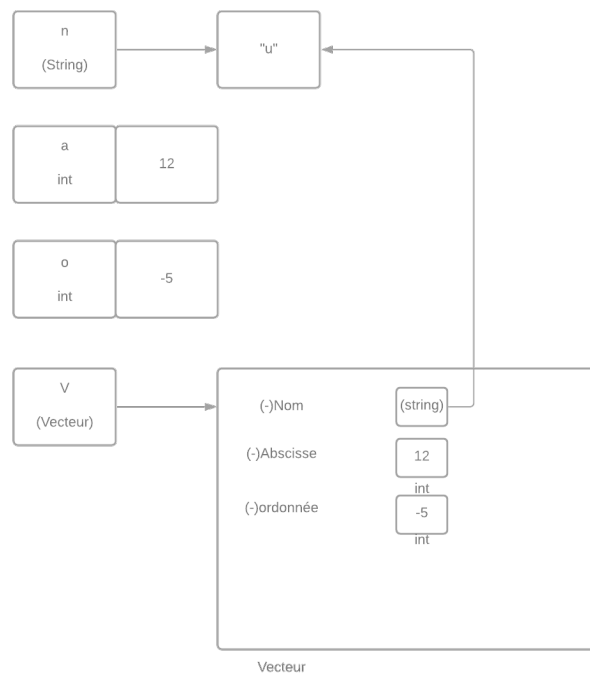
        System.out.println("entrer nom du vecteur");
        n = clavier.next();
        System.out.println("entrer l'abscisse");
        a = clavier.nextInt();
        System.out.println("entrer l'ordonnée");
        o = clavier.nextInt();

        //construction du vecteur

        Vecteur v1 = new vecteur(n,a,o);

        //affichage vecteur
        v1.toString();
    }
}
    
```

Etat mémoire :



Exercice 2 :

1.

```
class Livre{  
    private String Titre;  
    private int ISBN;  
    private double Prix;  
  
    //constructeur par défaut  
  
    public Livre(){  
        this("X",0,0.0);  
    }  
  
    //constructeur par initialisation  
  
    public Livre(String t,int i,double p){  
        Titre = t;  
        ISBN = i;  
        Prix = p;  
    }  
}
```

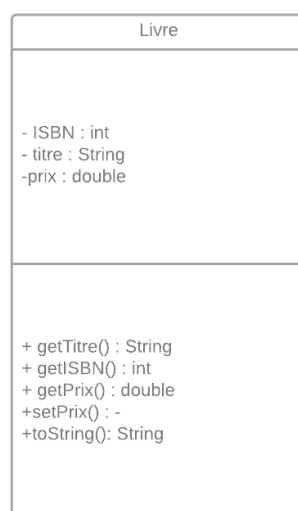
2. Getters:

- getTitre();
- getISBN();
- getPrix();

Setters:

- setPrix();

3.4.



5.

```

import java.util.*;

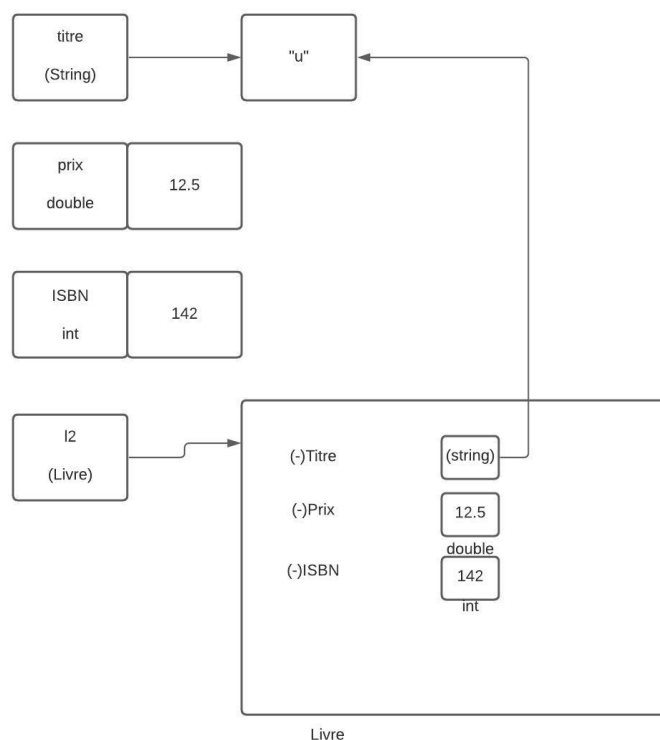
class testlivre{
    private static Scanner clavier = new Scanner(System.in);
    Run | Debug
    public static void main(String[] args){
        String titre;int ISBN;double prix;

        System.out.println("entrer le titre du livre");
        titre = clavier.next();
        System.out.println("entrer l'ISBN");
        ISBN = clavier.nextInt();
        System.out.println("entrer prix");

        Livre l1 = new Livre();
        Livre l2 = new Livre(titre, ISBN, prix);

        System.out.println(l1.toString());
        System.out.println(l2.toString());
    }
}

```



Exercice 3 :

```
class ClasseA{
    private int a;
    private double b;

    public ClasseA(int a,double b){
        this.a = a;
        this.b = b;
    }

    public void setA(int a){
        this.a = a;
    }

    public void setB(double b){
        this.b = b;
    }

    public int getA(){
        return a;
    }

    public double getB(){
        return b;
    }

    public void afficher(){
        System.out.println(a+" "+b);
    }
}
```

```
class ClasseB{
    private String chaine;
    private boolean a;

    public ClasseB(String c,boolean a){
        chaine = c;
        this.a = a;
    }

    public void setA(boolean a){
        this.a = a;
    }

    public String getChaine(){
        return chaine;
    }

    public boolean getA(){
        return a;
    }

    public String toString(){
        return chaine+" "+a;
    }

    public void afficher(){
        System.out.println(chaine+" "+a);
    }
}
```

Exercice 4 :

1. Code class Cercle :

```
class Cercle {
    private double rayon;
    private double a;
    private double b;
    private double[] centre = {0,0};

    //constructeur par défaut
    public Cercle(){
        this(1,0,0);
    }

    //constructeur par initialisation
    public Cercle(double r,double a,double b){
        if(r>=0){
            rayon = r;
        }

        this.a = a;
        this.b = b;

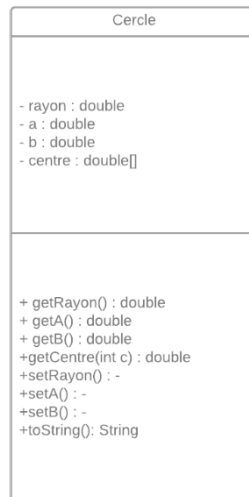
        centre[0] = a;centre[1] = b;
    }

    //getters
    public double getRayon(){
        return rayon;
    }
    public double getA(){
        return a;
    }
    public double getB(){
        return b;
    }
    public double getCentre(int c){
        return centre[c];
    }
}
```

```
//setters
public void setRayon(double r){
    if(r>=0){
        rayon = r;
    }
}
public void setA(double a){
    if(a>=0){
        this.a = a;
        centre[0] = a;
    }
}
public void setB(double b){
    if(b>=0){
        this.b = b;
        centre[1] = b;
    }
}

//toString
public String toString(){
    return "le cercle est de rayon : "+rayon+" et de coordonnees ["+a+" "+b+"] et de centre : "+centre[0]+" "+centre[1]+".";
}
```

2.



3. Code testCercle :

```

import java.util.*;

class testCercle {
    private static Scanner clavier = new Scanner(System.in);

    Run | Debug
    public static void main(String[] args){
        double distance,abscisse1,abscisse2;

        Cercle c1 = new Cercle();
        Cercle c2 = new Cercle(1,3.5,-1.6);

        distance = Math.sqrt((Math.pow(c2.getCentre(0),2)+Math.pow(c2.getCentre(1),2)));
        System.out.println("la distance du cercle de l'origine est : "+distance);

        abscisse1 = c1.getA();
        abscisse2 = c2.getA();

        if(abscisse1>0){
            System.out.println(c1.toString());
        }
        else{
            if(abscisse2>0){
                System.out.println(c2.toString());
            }
        }
    }
}
    
```