**TP5 : Classes Abstraites et interfaces**

**Exercice 1 :**

**Partie 1 :**

public abstract class Forme {

public abstract double perimetre();

public abstract double surface();

}

class Point {

private double x;

private double y;

public Point(double x,double y){

this.x=x;

this.y=y;}

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

public double getX() { return (this.x); }

public void setY (double y) { y=y; }

public double getY() {return ( this.y); }

public String toString() { return"(x:"+x+",y:"+y+")"; }

}

public class Cercle extends Forme {

private Point centre;

private double rayon;

public Cercle(Point c,double r){

this.centre=c;

this.rayon=r;}

public void setCentre(Point c) { centre=c; }

public Point getCentre() { return(this.centre); }

public void setRayon(double r) { rayon=r; }

public double getRayon() { return (this.rayon); }

public double perimetre() { return 2\*rayon\*Math.PI; }

public double surface() { return rayon\*rayon\*Math.PI; }

public String toString(){ return"cercle(centre="+centre.toString()+",rayon="+rayon+",perimetre="+perimetre()+",surface="+surface()+")";}

}

public class Rectangle extends Forme {

private double longueur;

private double largeur;

public Rectangle(double lon,double lar){

this.longueur=lon;

this.largeur=lar;}

public void setLongueur(double lon) { longueur=lon; }

public double getLongueur() { return(this.longueur); }

public void setLargeur(double lar) { largeur=lar; }

public double getLargeur() { return(this.largeur); }

public double perimetre() { return 2\*(longueur+largeur); }

public double surface() { return longueur\*largeur; }

public String toString(){ return"rectangle(longueur="+longueur+",largeur="+largeur+",perimetre="+perimetre()+",surface="+surface()+")";}

}

public class TestForme {

public static void main(String[] args) {

Point p=new Point(2,3);

Cercle c=new Cercle(p,4);

Rectangle r=new Rectangle(2,2);

System.out.println (p);

System.out.println (c);

System.out.println (r);}

}

**Partie 2 :**

public interface Forme {

public abstract double perimetre();

public abstract double surface();

}

public interface Affichable {

public abstract void afficher();

}

class Point implements Affichable {

private double x;

private double y;

public Point(double x,double y){

this.x=x;

this.y=y;}

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

public double getX() { return (this.x); }

public void setY(double y) { y=y; }

public double getY() { return (this.y); }

public String toString() { return"(x:"+x+",y:"+y+")"; }

public void afficher() { System.out.println(toString()); }

}

public class Cercle implements Forme,Affichable {

private Point centre;

private double rayon;

public Cercle(Point c,double r){

this.centre=c;

this.rayon=r;}

public void setCentre(Point c) { centre=c; }

public Point getCentre() { return (this.centre); }

public void setRayon(double r) { rayon=r; }

public double getRayon() { return (this.rayon); }

public double perimetre() { return 2\*rayon\*Math.PI; }

public double surface() { return rayon\*rayon\*Math.PI; }

public String toString(){ return"cercle(centre="+centre.toString()+",rayon="+rayon+",perimetre="+perimetre()+",surface="+surface()+")";}

public void afficher() {System.out.println(toString());}

}

public class Rectangle implements Forme,Affichable {

private double longueur;

private double largeur;

public Rectangle(double lon,double lar){

this.longueur=lon;

this.largeur=lar;}

public void setLongueur(double lon) { longueur=lon; }

public double getLongueur() { return (this.longueur); }

public void setLargeur(double lar) { largeur=lar; }

public double getLargeur() { return (this.largeur); }

public double perimetre() { return 2\*(longueur+largeur); }

public double surface() { return longueur\*largeur; }

public String toString(){ return"rectangle(longueur="+longueur+",largeur="+largeur+",perimetre="+perimetre()+",surface="+surface()+")";}

public void afficher() { System.out.println(toString()); }

}

public class TestFormeInterface {

public static void main(String[] args) {

Point p=new Point(2,3);

Affichable c=new Cercle(p,4);

Affichable r=new Rectangle(2,2);

Affichable TabObjet[]=new Affichable[3];

TabObjet[0]=p;

TabObjet[1]=c;

TabObjet[2]=r;

for(int i=0;i<TabObjet.length;i++) { TabObjet[i].afficher();} }

}