

Atelier n°1 : Programmation Orientée Objets

Exercice 1

1/

Classe Point :

using System;

namespace Atelier1_EX1

{

 public class Point

 {

 private String nom;

 private int abscisse;

 public Point(String nom, int abscisse)

 {

 this.nom = nom;

 this.abscisse = abscisse;

 }

 public String getNom()

 {

 return nom;//return this.nom;

 }

 public int getAbscisse()

 {

 return this.abscisse;//return abscisse;

 }

 public void setNom(String nom)

 {

 this.nom = nom;

 }

 public void setAbscisse(int abscisse)

 {

 this.abscisse = abscisse;

 }

 public void afficher1()

 {

 Console.WriteLine("Nom:" + nom + " Abscisse:" + abscisse);

 }

```
        public String afficher2()
        {
            return "Nom:" + nom + " Abscisse:" + abscisse;
        }
        public void translate(int t)
        {
            abscisse = abscisse + t;
        }
    }
}
```

2/

Classe Main :

```
namespace Atelier01_EX1;
class Program
{
    static void Main(string[] args)
    {
        //création d'un point p1 : A(3) -> Classe obj = new Constructeur(...)
        Point p1 = new Point("A",3);
        //Afficher le point p1
        p1.affiche();
        //Traduire p1 de 3 pas
        p1.translate(3);
        //Afficher le point p1
        p1.affiche();
        //Traduire le point p1 d'un nombre de pas donné par l'utilisateur
        Console.WriteLine("Donnez le pas:");
        int pas = Convert.ToInt32(Console.ReadLine());
        //int pas = int.Parse(Console.ReadLine());
        p1.translate(pas);
        //Afficher le point p1
        p1.affiche();
        Console.Read();
    }
}
```

Exercice 2

1/

Classe Point :

using System;

namespace Atelier1_EX2

```
{  
    public class Point  
    {  
        private float x, y;  
        private String nom;  
        public Point()//Constructeur par défaut  
        {  
            x = y = 0;//x=0; y=0;  
        }  
        public Point(float x, float y, string nom)  
        {  
            this.x = x;  
            this.y = y;  
            this.nom = nom;  
        }  
        public void afficher()  
        {  
            Console.WriteLine("Nom:" + nom + "(" + x + "," + y + ")");  
        }  
        public void deplacer(float a, int b)  
        {  
            x = x + a;  
            y = y + b;  
        }  
    }  
}
```

Classe Disque :

using System;

namespace Atelier01_EX2

```
{  
    public class Disque
```

```
{
    private Point centre;
    private float rayon;
    public Disque()
    {
        rayon = 10;
        centre = new Point();
    }
    public Disque(Point centre, float rayon)
    {
        this.centre = centre;
        this.rayon = rayon;
    }
    public float getSurface()
    {
        return (float)(Math.PI * Math.Pow(rayon,2));
    }
    public void afficher()
    {
        centre.affiche();Console.WriteLine("Rayon:" + rayon);
    }
    public void deplacer(float a, float b)
    {
        /*Méthode1
        float xx = centre.getX();
        float yy = centre.getY();
        xx = xx + a;
        yy = yy + b;
        centre.setX(xx);
        centre.setY(yy);
        */
        /*Méthode 2
        * centre.setX(centre.getX()+a)
        * centre.setY(centre.getY()+b);
        */
    }
}
```

```
        /*Méthode 3
        * centre.deplacer(a,b);
        */
    }
}
}
```

Classe Main

```
namespace Atelier01_EX2;
class Program
{
    static void Main(string[] args)
    {
        Point p1 = new Point();
        Point p2 = new Point(1, 23, "A");
        p1.affiche();
        p2.affiche();
        p2.deplacer(5, 5);
        p2.affiche();
        Disque d1 = new Disque();
        Disque d2 = new Disque(p1, 10);
        d1.afficher();
        d2.afficher();
        Console.WriteLine("Surface de d1:" + d1.getSurface());
        Console.WriteLine("Surface de d2:" + d2.getSurface());
        d2.deplacer(3, 5);
        d2.afficher();
        Console.WriteLine("Surface de d2:" + d2.getSurface());
        Console.Read();
    }
}
```