

Programmazione orientata agli oggetti

Sviluppare una classe `Cerchio`. Il cerchio dovrà essere descritto da:

- raggio
- centro

Il centro deve essere un oggetto della classe `Point`.

La classe `Cerchio` dovrà avere i metodi per accedere ai dati, modificarli, calcolare circonferenza e area.

Classe Punto

```
class Punto{  
    private double x;  
    private double y;  
    ...  
}
```

Dati membro private della classe: non saranno visibili al di fuori della classe!

Classe Punto - Costruttori

I costruttori servono a inizializzare i membri della classe in fase di creazione dell'oggetto. Dovremo prevedere un costruttore per ogni inizializzazione che vogliamo fare (overloading dei costruttori)

```
Class Punto{  
    ...  
    public Punto(){}  
  
    public Punto(double x, double y){  
        this.x = x;  
        this.y = y;  
    }  
  
    ...  
}
```

Costruttore di default. Le variabili sono inizializzate a 0!

Costruttore che prevede 2 double come parametri per inizializzare x e y.

Classe Punto - Test

E' utile testare il comportamento della classe durante il suo sviluppo. Costruiamo quindi una classe `Test` contenente il `Main()`

```
class Test{  
    public static void Main(){  
        Punto p1 = new Punto(1.2, 2.4);  
        //Console.WriteLine(p1.x);  
        //p1.x=3.5;  
    }  
}
```

Tentativi di accedere ad un dato membro privato al di fuori della classe: si ottiene un errore in compilazione *'Lezione.Punto.x' is inaccessible due to its protection level*

Classe Punto – Accesso ai dati

Dobbiamo fornire dei metodi che permettano di accedere ai dati membro privati della classe. Il programmatore decide quali sono i dati accessibili dall'esterno e come renderli accessibili, pertanto è consigliabile mantenere i membri privati e fornire le funzioni pubbliche che permettono tale accesso. In C# esistono le properties che permettono di mantenere tale incapsulamento con un'interfaccia per l'utente intuitiva.

```
public double X
{
    set
    {
        x = value;
    }
    get
    {
        return x;
    }
}
```

Strutture Software 1

5

Classe Punto – Accesso ai dati

Non sempre l'accesso ai dati tramite le properties è conveniente, pertanto può essere necessario fornire metodi public che permettano l'accesso e la modifica dei dati private.

```
public void SetXY(double xx, double yy)
{
    x = xx;
    y = yy;
}
```

Strutture Software 1

6

Classe Punto – ToString()

Tutte le classi in C# derivano da System.Object, pertanto ereditano i membri virtuali che la classe Object fornisce, tra cui il metodo ToString() che fornisce una string rappresentante l'oggetto.

```
class Punto{
    ...
    public override string ToString(){
        string tmp;
        tmp = "x= " + x + " y= " + y;
        return tmp;
    }
    ...
}
```

La classe Punto esegue l'overriding del metodo ToString() fornito dalla classe Object

Strutture Software 1

7

Classe Punto - Test

```
class Test{
    public static void Main(){
        Test1();
    }

    public static void Test1(){
        Punto p1 = new Punto(1.2, 2.4);

        Console.WriteLine(p1.X);
        Console.WriteLine(p1.Y);
        Console.WriteLine(p1.ToString());

        p1.X = 3.4;
        p1.Y = 5.6;

        Console.WriteLine(p1.ToString());
        p1.SetXY(7.7, 8.8);
        Console.WriteLine(p1.ToString());
    }
}
```

```
1.2
2.4
x=1.2 y=2.4
x=3.4 y=5.6
x= 7.7 y=8.8
```

Strutture Software 1

8

Classe Cerchio

```
class Cerchio{
    private double raggio;
    private Punto centro;
    ...
}
```

Reference: di default
sono inizializzati a null

Classe Cerchio - Costruttori

```
class Cerchio{
    ...
    public Cerchio(double r, double x, double y)
    {
        raggio = r;
        centro = new Punto(x, y);
    }
    public Cerchio(double r, Punto c)
    {
        raggio = r;
        centro = new Punto(c.X, c.Y);
    }
    ...
}
```

Creazione di un oggetto
di tipo Punto, attraverso il
costruttore della classe Punto

Classe Cerchio – Accesso ai dati

```
class Cerchio{
    ...
    public double Raggio
    {
        set
        {
            raggio = value;
        }
        get
        {
            return raggio;
        }
    }
    ...
}
```

properties

Classe Cerchio – Accesso ai dati

```
class Cerchio{
    ...
    public Punto GetCentro()
    {
        Punto tmp = new Punto(centro.X, centro.Y);
        return tmp;
    }
    public void SetCentro(Punto c)
    {
        centro.X=c.X;
        centro.Y=c.Y;
    }
    ...
}
```

E' necessario tornare un nuovo
oggetto (copia dell'oggetto corrente)
e non un reference per evitare
modifiche allo stato dell'oggetto non
volute!

Classe Cerchio – Altri metodi

```
class Cerchio{
    ...
    public double CalcolaArea()
    {
        return 3.14 * raggio * raggio;
    }

    public double CalcolaCirconferenza()
    {
        return 2 * 3.14 * raggio;
    }

    public override string ToString()
    {
        string tmp;
        tmp = "C:("+centro.X+", "+centro.Y+ " ), R: " + raggio;
        return tmp;
    }
    ...
}
```

Classe Cerchio - Test

```
class Test{
    ...
    Test2(){
        Punto p1 = new Punto(1.2, 2.4);
        Cerchio c1 = new Cerchio(3, p1);
        Console.WriteLine(c1.Raggio);
        Console.WriteLine(c1.GetCentro().ToString());
        Console.WriteLine(c1.ToString());
    }
    ...
}
```

Accesso al dato membro tramite properties

Viene chiamato il metodo ToString() dell'oggetto di tipo Point tornato dal metodo GetCentro()

Viene chiamato il metodo ToString() dell'oggetto di tipo Cerchio

3
x= 1,2 y= 2,4
Centro : (1,2;2,4), Raggio: 3

Classe Cerchio - Test

```
class Test{
    ...
    Test3(){
        Punto p1 = new Punto(1.2, 2.4);
        Cerchio c1 = new Cerchio(3, p1);
        Punto p2 = new Punto(10, 11);
        c1.SetCentro(p2);
        Console.WriteLine(c1.ToString());
        p2.X = 13;
        Console.WriteLine(c1.ToString());
        Console.WriteLine("Area: "+c1.CalcolaArea());
        Console.WriteLine("Crf:"+c1.CalcolaCirconferenza());
    }
}
```

Creiamo un nuovo punto p2 e lo assegniamo come centro del Cerchio c2

Modifichiamo la x di p2 (attraverso le properties) e verificiamo di NON aver cambiato lo stato di c1

Centro : (10;11), Raggio: 3
Centro : (10;11), Raggio: 3
Area: 28,26
Crf: 18,84