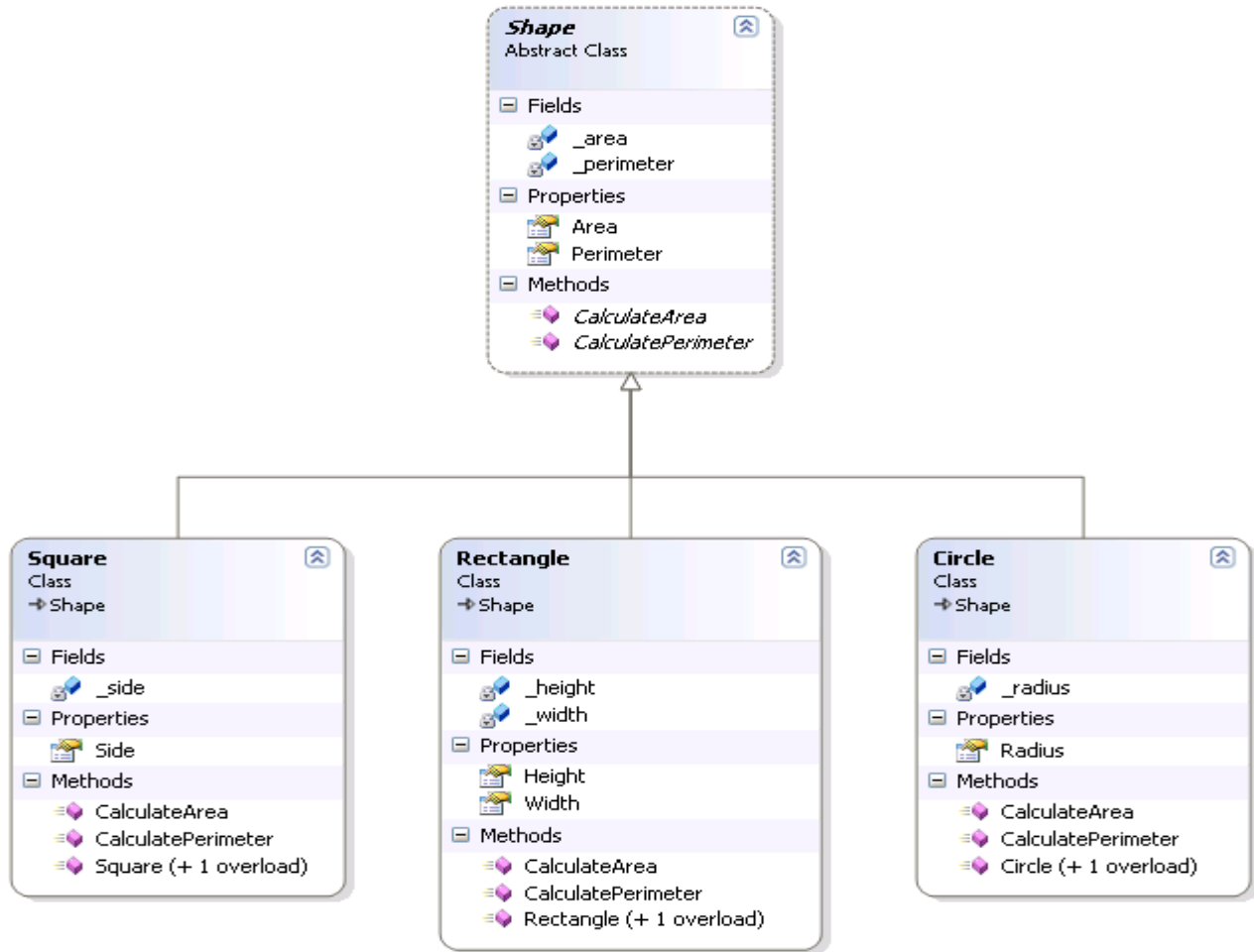


Abstract Class Example

Class Diagram



Form Design Screen

The application window displays three panels for calculating area and perimeter:

- Circle:** Enter Radius: 5.8. Calculate Area: 105.6832. Calculate Perimeter: 36.44248.
- Rectangle:** Enter Height: 6.3, Enter Width: 7.24. Calculate Area: 45.612. Calculate Perimeter: 27.08.
- Square:** Enter Side: 7.71. Calculate Area: 59.4441. Calculate Perimeter: 30.84.

```
// Shape.cs (This is Abstract Class)
```

```
using System;
```

```
namespace AbstractExample
```

```
{  
    public abstract class Shape  
    {  
        private float _area;  
        private float _perimeter;  
  
        public float Area  
        {  
            get{ return _area; }  
            set{ _area = value; }  
        }  
  
        public float Perimeter  
        {  
            get{ return _perimeter; }  
            set{ _perimeter = value; }  
        }  
  
        public abstract void CalculateArea();  
        public abstract void CalculatePerimeter();  
    }  
}
```

```
// Circle.cs (This is Drived Class from Shape Base Class)
```

```
using System;
```

```
namespace AbstractExample
```

```
{  
    class Circle : Shape  
    {  
        private float _radius;  
  
        public Circle() { }  
  
        public Circle(float radius)  
        {  
            _radius = radius;  
        }  
  
        public float Radius  
        {  
            get{ return _radius; }  
            set{ _radius = value; }  
        }  
  
        public override void CalculateArea()  
        {  
            this.Area = (float)System.Math.PI * _radius * _radius;  
        }  
  
        public override void CalculatePerimeter()  
        {  
            this.Perimeter = (float)System.Math.PI * (_radius * 2);  
        }  
    }  
}
```

```
// Rectangle.cs (This is Drived Class from Shape Base Class)
```

```
using System;
```

```
namespace AbstractExample  
{
```

```
    class Rectangle : Shape  
    {
```

```
        private float _height;  
        private float _width;
```

```
        public Rectangle() { }
```

```
        public Rectangle(float height, float width)  
        {  
            _height = height;  
            _width = width;  
        }
```

```
        public float Height  
        {  
            get{ return _height; }  
            set{ _height = value; }  
        }
```

```
        public float Width  
        {  
            get{ return _width; }  
            set{ _width = value; }  
        }
```

```
        public override void CalculateArea()  
        {  
            this.Area = _height * _width;  
        }  
        public override void CalculatePerimeter()  
        {  
            this.Perimeter = (_height * 2) + (_width * 2);  
        }  
    }  
}
```

```
// Square.cs (This is Drived Class from Shape Base Class)
```

```
using System;
```

```
namespace AbstractExample  
{
```

```
    class Square : Shape  
    {
```

```
        private float _side;
```

```
        public Square() { }
```

```
        public Square(float side)  
        {  
            _side = side;  
        }
```

```
        public float Side  
        {  
            get{ return _side; }  
            set{ _side = value; }  
        }  
    }  
}
```

```

    public override void CalculateArea()
    {
        this.Area = _side * _side;
    }

    public override void CalculatePerimeter()
    {
        this.Perimeter = _side * 4;
    }
}
}
}

```

// Form Code

```

using System;
using System.Windows.Forms;

namespace AbstractExample
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnCalculateCircleArea_Click(object sender, EventArgs e)
        {
            try
            {
                // create a new instance of Circle Class
                Circle objCircle = new Circle();
                // set the radius value to Circle Class through property
                objCircle.Radius = float.Parse(txtRadius.Text);
                //calculate The Area Of Circle
                objCircle.CalculateArea();
                //Display the Value of Calculated Area of Circle using Area Property
                txtCircleArea.Text = Convert.ToString(objCircle.Area);
            }
            catch (Exception ee)
            {
                MessageBox.Show(ee.Message);
            }
        }

        private void btnCalculateCirclePerimeter_Click(object sender, EventArgs e)
        {
            try
            {
                // create a new instance of Circle Class with passing Parameterized constructor
                Circle objCircle = new Circle(float.Parse(txtRadius.Text));
                //calculate The Perimeter Of Circle
                objCircle.CalculatePerimeter();
                //Display the Value of Calculated Perimeter of Circle using Perimeter Property
                txtCirclePerimeter.Text = Convert.ToString(objCircle.Perimeter);
            }
            catch (Exception ee)
            {
                MessageBox.Show(ee.Message);
            }
        }

        private void btnCalculateRectangleArea_Click(object sender, EventArgs e)
        {
            try
            {
                // create a new instance of Circle Class
                Rectangle objRect = new Rectangle();
            }
        }
    }
}

```

