```csharp
1  using SplashKitSDK;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Runtime.InteropServices;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace ShapeDrawer
10 {
11     public class MyRectangle : Shape
12     {
13         private int _width;
14         private int _height;
15
16         public MyRectangle(Color color, float x, float y, int width, int
             height) : base(color)
17         {
18             X = x;
19             Y = y;
20             _width = width;
21             _height = height;
22         }
23
24         public MyRectangle() : this (Color.Green, 0.0f, 0.0f, 100, 100)
25         {
26             Color = Color.Green;
27             X = 0.0f;
28             Y = 0.0f;
29             _width = 100;
30             _height = 100;
31         }
32         public int Width
33         {
34             get => _width;
35             set => _width = value;
36         }
37
38         public int Height
39         {
40             get => _height;
41             set => _height = value;
42         }
43
44         public override void DrawOutline()
45         {
46             SplashKit.FillRectangle(Color.Black, X - 2, Y - 2, _width + 4,
                 _height + 4);
47         }
```

```csharp
48
49          public override void Draw()
50          {
51              if (Selected)
52              {
53                  DrawOutline();
54              }
55              SplashKit.FillRectangle(Color, X, Y, _width, _height);
56          }
57
58          public override bool IsAt(Point2D point)
59          {
60              if (point.X >= X && point.X <= X + _width)
61              {
62                  if (point.Y >= Y && point.Y <= Y + _height)
63                  {
64                      return true;
65                  }
66              }
67              return false;
68          }
69
70          public override void SaveTo(StreamWriter writer)
71          {
72              writer.WriteLine("Rectangle");
73              base.SaveTo(writer);
74              writer.WriteLine(Width);
75              writer.WriteLine(Height);
76          }
77
78          public override void LoadFrom(StreamReader reader)
79          {
80              base.LoadFrom(reader);
81              Width = reader.ReadInteger();
82              Height = reader.ReadInteger();
83          }
84      }
85  }
86
```