```csharp
1  using SplashKitSDK;
2  using System;
3  using System.IO;
4
5  namespace MyGame
6  {
7      public class MyRectangle:Shape
8      {
9          private int _width;
10         private int _height;
11         public MyRectangle() : this(Color.Green, 0.0f, 0.0f, 173, 173)
12         {
13
14         }
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 int Width
25         {
26             get
27             {
28                 return _width;
29             }
30             set
31             {
32                 _width = value;
33             }
34         }
35
36         public int Height
37         {
38             get
39             {
40                 return _height;
41             }
42             set
43             {
44                 _height = value;
45             }
46         }
47
48         public override void Draw()
```

```csharp
49          {
50              if (Selected)
51              {
52                  DrawOutline();
53              }
54              SplashKit.FillRectangle(Color, X, Y, _width, _height);
55          }
56
57          public override void DrawOutline()
58          {
59              SplashKit.FillRectangle(Color.Black , X - (5 + 3), Y - (5 + 3), ⤶
                    _width + 2 * (5 + 3), _height + 2 * (5 + 3));
60          }
61
62          public override bool IsAt(Point2D pt)
63          {
64              if (pt.X > X && pt.X < X + _width && pt.Y > Y && pt.Y < Y +      ⤶
                  _height)
65              {
66                  return true;
67              }
68              else
69              {
70                  return false;
71              }
72          }
73
74          public override void SaveTo(StreamWriter writer)
75          {
76              writer.WriteLine("Rectangle");
77              base.SaveTo(writer);
78              writer.WriteLine(_width);
79              writer.WriteLine(_height);
80          }
81
82          public override void LoadFrom(StreamReader reader)
83          {
84              base.LoadFrom(reader);
85              _width = reader.ReadInteger();
86              _height = reader.ReadInteger();
87          }
88      }
89 }
90
```