```
 1  using SplashKitSDK;
 2  using System;
 3  using System.Collections.Generic;
 4  using System.Linq;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7
 8  namespace _4._1._1
 9  {
10      public class Drawing
11      {
12          private readonly List<Shape> _shapes;
13          private SplashKitSDK.Color _background;
14          private SplashKitSDK.Color Color;
15
16          public Drawing(SplashKitSDK.Color background)
17          {
18              _background = background;
19              _shapes = new List<Shape>();
20          }
21          public SplashKitSDK.Color background
22          {
23              get
24              {
25                  return _background;
26              }
27              set
28              {
29                  _background = value;
30              }
31          }
32          public Drawing() : this(SplashKitSDK.Color.White)
33          {
34
35          }
36          public List<Shape> SelectedShapes
37          {
38              get
39              {
40                  var result = new List<Shape>(); ;
41                  foreach (var shape in _shapes)
42                  {
43                      if (shape.Selected)
44                      {
45                          result.Add(shape);
46                      }
47                  }
48                  return result;
49              }
```

```
50          }
51          public int ShapeCount
52          {
53              get { return _shapes.Count(); }
54          }
55          public void AddShape(Shape shape)
56          {
57              _shapes.Add(shape);
58          }
59          public void RemoveShape(Shape shape)
60          {
61              _shapes.Remove(shape);
62          }
63          public void Draw()
64          {
65              SplashKit.ClearScreen(_background);
66              foreach (Shape shape in _shapes)
67              {
68                  shape.Draw();
69              }
70          }
71
72          public void SelectShapesAt(Point2D pt)
73          {
74              foreach (var shape in _shapes)
75              {
76                  if (shape.IsAt(pt))
77                  {
78                      shape.Selected = true;
79                  }
80                  else { shape.Selected = false; }
81              }
82          }
83
84      }
85  }
86
87
```