```
 1  using SplashKitSDK;
 2  using System.IO;
 3
 4  namespace ShapeDrawer
 5  {
 6      public class Drawing
 7      {
 8          private readonly List<Shape> _shapes;
 9          private Color _background;
10
11          public Color Background
12          {
13              get => _background;
14              set => _background = value;
15          }
16
17          public Drawing(Color background)
18          {
19              _shapes = new List<Shape>();
20              _background = background;
21          }
22
23          public Drawing() : this (Color.White)
24          {
25              _shapes = new List<Shape>();
26              _background  = Color.White;
27          }
28
29          public int ShapeCount
30          {
31              get => _shapes.Count;
32          }
33
34          public void AddShape(Shape shape)
35          {
36              _shapes.Add(shape);
37          }
38
39          public void RemoveShape(Shape shape)
40          {
41              _= _shapes?.Remove(shape);
42          }
43
44          public void Draw()
45          {
46              SplashKit.ClearScreen(_background);
47              for (int i = 0; i < _shapes.Count; i++)
48              {
49                  if (_shapes[i].Selected)
```

```
50                  {
51                      _shapes[i].DrawOutline();
52                  }
53                  _shapes[i].Draw();
54              }
55          }
56
57          public void SelectShapesAt(Point2D point)
58          {
59              foreach (Shape s in _shapes)
60              {
61                  s.Selected = s.IsAt(point);
62              }
63          }
64
65          public List<Shape> SelectedShapes
66          {
67              get
68              {
69                  List<Shape> result = new List<Shape>();
70                  foreach (Shape s in _shapes)
71                  {
72                      if (s.Selected)
73                      {
74                          result.Add(s);
75                      }
76                  }
77                  return result;
78              }
79          }
80
81          public void Save(string filename)
82          {
83              StreamWriter writer;
84
85              writer = new StreamWriter(filename);
86              try
87              {
88                  writer.WriteColor(Background);
89                  writer.WriteLine(ShapeCount);
90
91                  foreach (Shape s in _shapes)
92                  {
93                      s.SaveTo(writer);
94                  }
95              }
96              finally
97              {
98                  writer.Close();
```

```
 99              }
100          }
101
102          public void Load(string filename)
103          {
104              StreamReader reader = new StreamReader(filename);
105              Shape s;
106              try
107              {
108                  Background = reader.ReadColor();
109                  int count = reader.ReadInteger();
110                  _shapes.Clear();
111
112                  for (int i = 0; i < count; i++)
113                  {
114                      string kind = reader.ReadLine();
115
116                      switch (kind)
117                      {
118                          case "Rectangle":
119                              s = new MyRectangle();
120                              break;
121
122                          case "Circle":
123                              s = new MyCircle();
124                              break;
125
126                          case "Line":
127                              s = new MyLine();
128                              break;
129
130                          default:
131                              throw new InvalidDataException("Unknown shape
                        kind: " + kind);
132                      }
133
134                      s.LoadFrom(reader);
135                      _shapes.Add(s);
136                  }
137              }
138              finally
139              {
140                  reader.Close();
141              }
142          }
143      }
144  }
145
```