

Visual Studio IDE showing the ShapeDrawer application. The code in the ShapeDrawer.cs file is as follows:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 {
28     _background = value;
29 }
30
31
32 1 reference
33 public Drawing() : this(Color.White)
34 {
35     // other steps could go here...
36 }
37
38 0 references
39 public int ShapeCount
```

The Shape Drawer window displays a collection of green squares arranged in a pattern. The Diagnostic Tools window on the right shows the following metrics:

- Events: 0 of 0
- Process Memory (MB): 50
- CPU (% of all processors): 100

The Error List and Call Stack windows are empty.

Visual Studio IDE showing the ShapeDrawer application. The code in the ShapeDrawer.cs file is as follows:

```
1
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using System.Collections.Generic;
8 using SplashScreen;
9
10 namespace ShapeDrawer
11 {
12     0 references
13     public class Drawing
14     {
15         private readonly List<Shape> _shapes;
16         private Color _background;
17         1 reference
18         public Drawing(Color background)
19         {
20             _shapes = new List<Shape>();
21             _background = background;
22         }
23         1 reference
24         public Color Background
25         {
26             get
27             {
28                 return _background;
29             }
30             set
31             {
32                 _background = value;
33             }
34         }
35     }
36
37 1 reference
38 public Drawing() : this(Color.White)
39 {
40     // other steps could go here...
41 }
42
43 0 references
44 public int ShapeCount
```

The Shape Drawer window displays a collection of green squares arranged in a pattern. The Diagnostic Tools window on the right shows the following metrics:

- Events: 0 of 0
- Process Memory (MB): 50
- CPU (% of all processors): 100

The Error List and Call Stack windows are empty.

Visual Studio Code interface showing a C# project named "ShapeDrawer". The code in the "ShapeDrawer.Drawing" file defines a class for drawing shapes on a background.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Collections.Generic;
7 using SplashScreen;
8
9 namespace ShapeDrawer
10 {
11     [Serializable]
12     public class Drawing
13     {
14         private readonly List<Shape> _shapes;
15         private Color _background;
16         [Reference]
17         public Drawing(Color background)
18         {
19             _shapes = new List<Shape>();
20             _background = background;
21         }
22         [Reference]
23         public Color Background
24         {
25             get
26             {
27                 return _background;
28             }
29             set
30             {
31                 _background = value;
32             }
33         }
34         [Reference]
35         public Drawing() : this(Color.White)
36         {
37             // other steps could go here...
38         }
39     }
40 }
41
42 [Reference]
43 public int ShapeCount
```

The "Shape Drawer" window displays a collection of green rectangular shapes arranged in a pattern on a white background.

The "Diagnostic Tools" panel on the right shows a summary of the application's performance:

- Events: 0
- Process Memory (MB): 10
- CPU (% of all processors): 0

The "Error List" panel at the bottom shows no errors or warnings.

The Windows taskbar at the bottom indicates the system is ready, with a temperature of 28°C and a date of 24/9/2023.