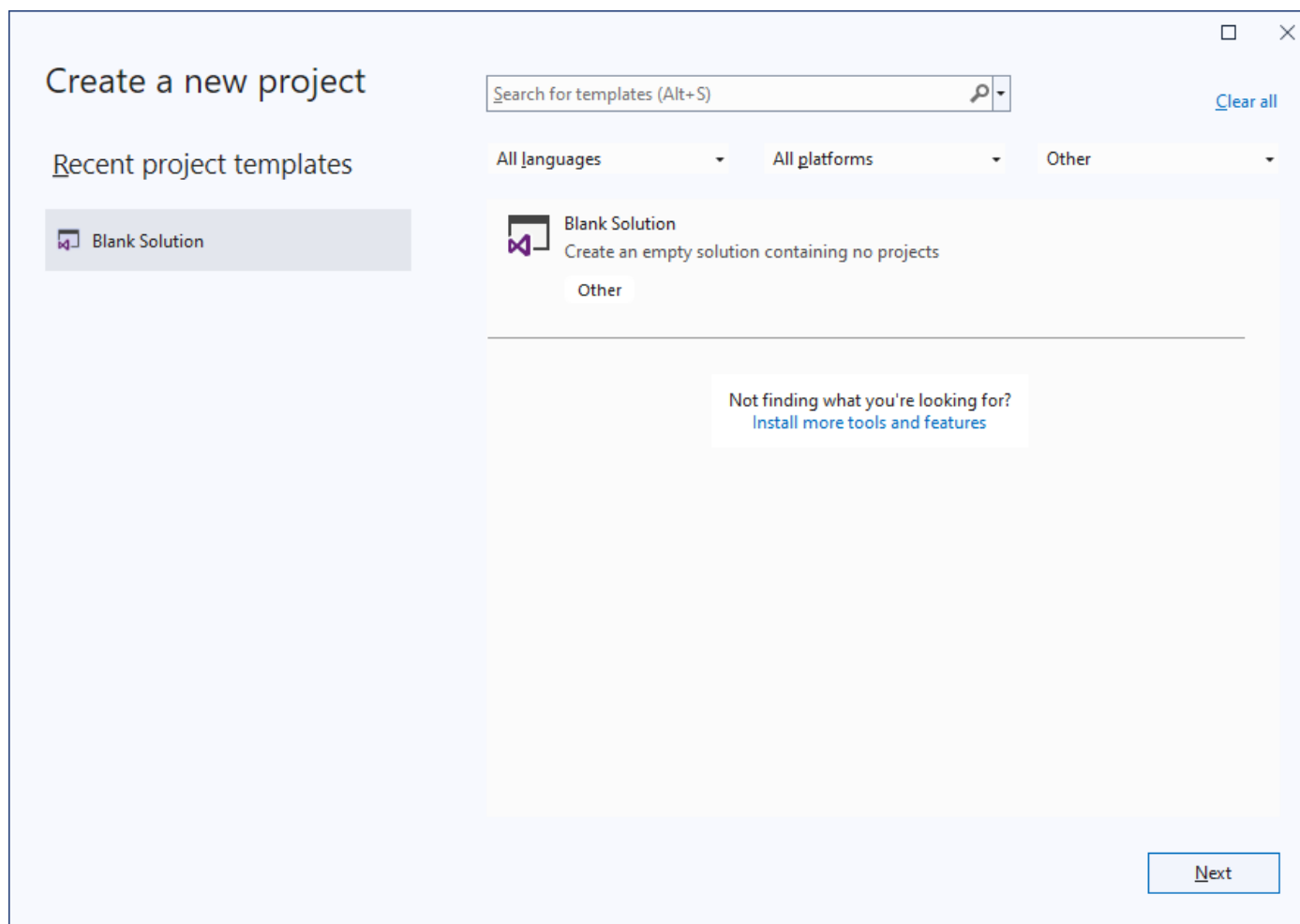
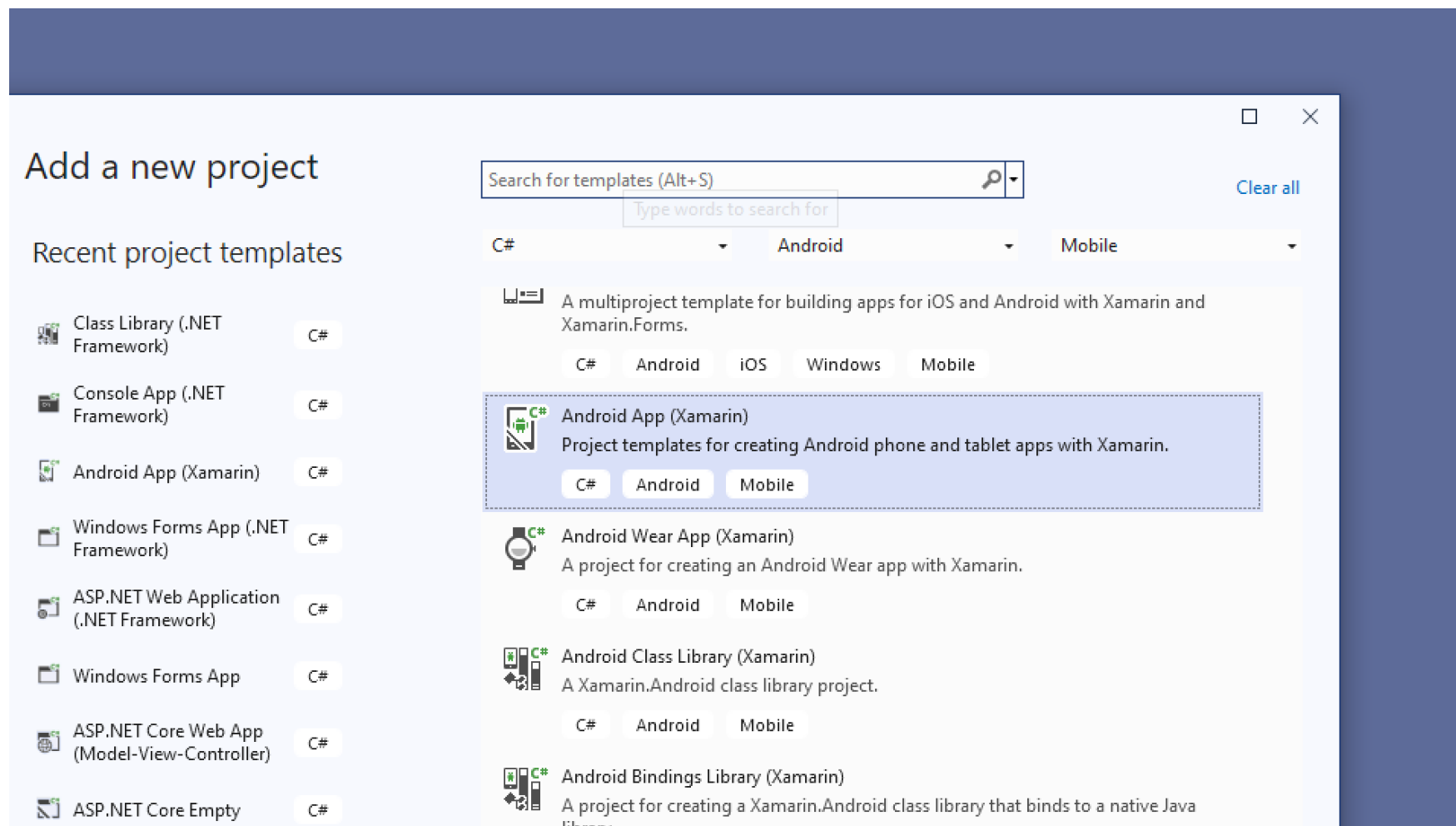


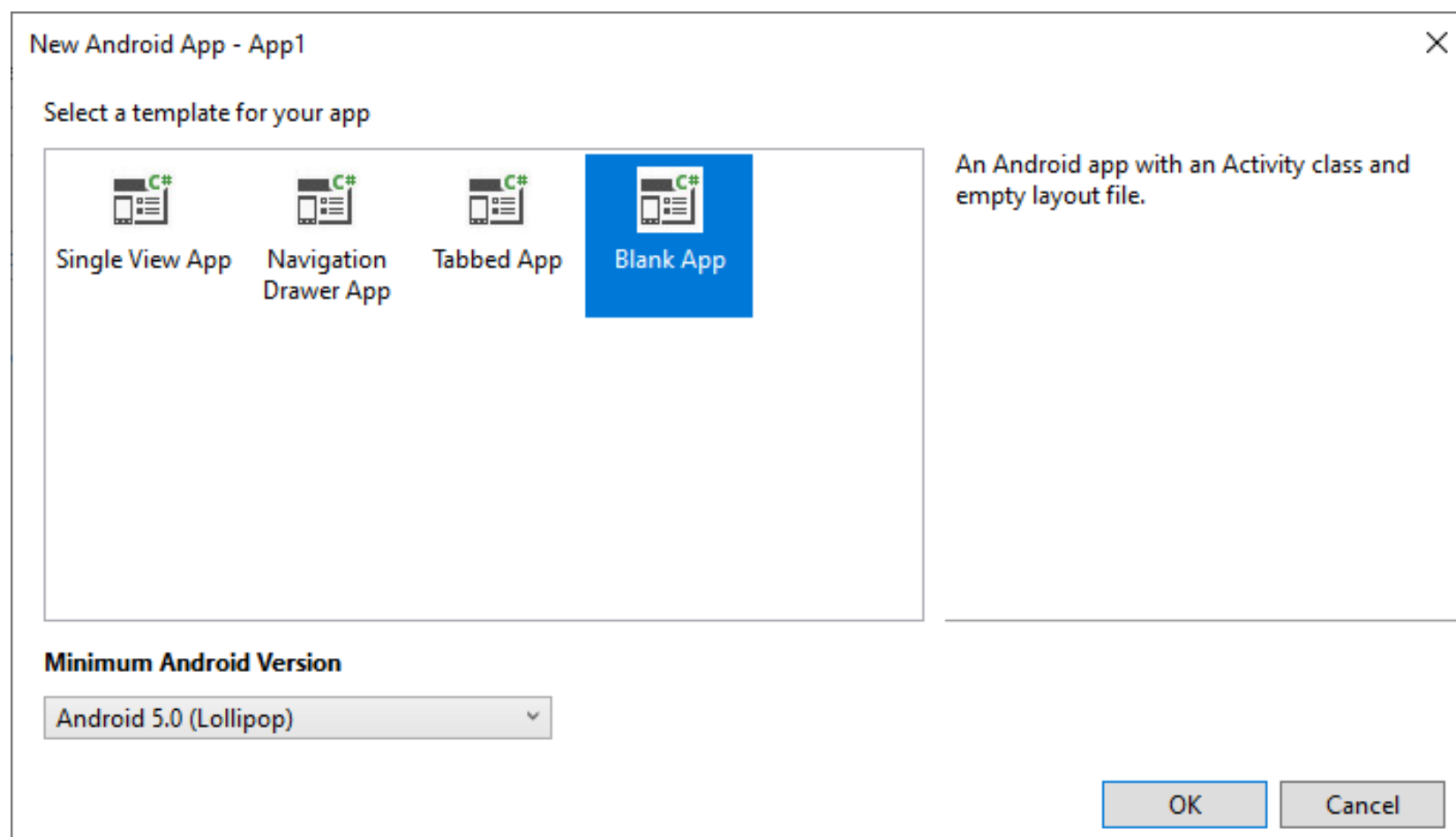
Създаване на мобилно приложение с използване на Model-View-Controller.



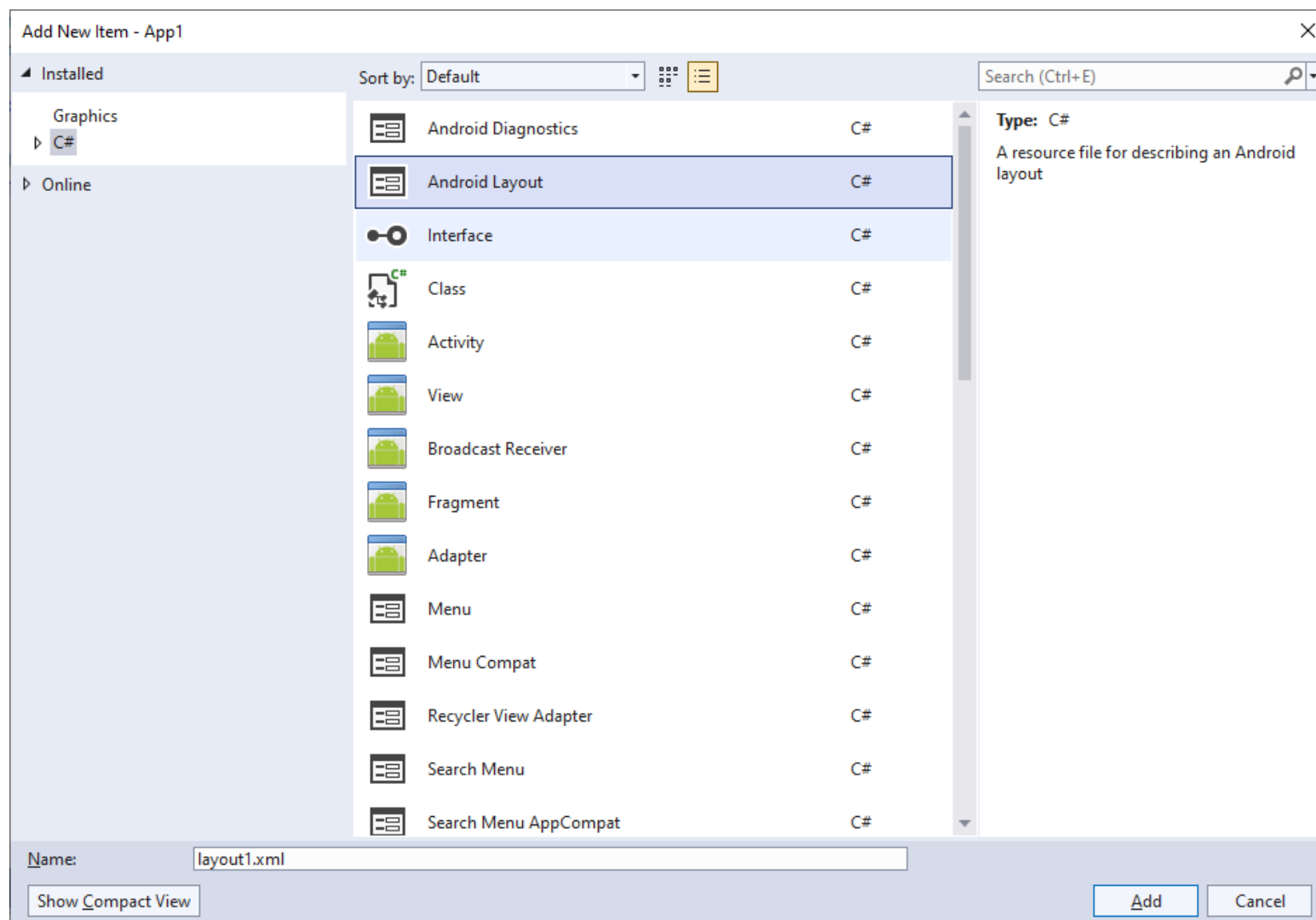
Фиг. 1. Създаване на решение.



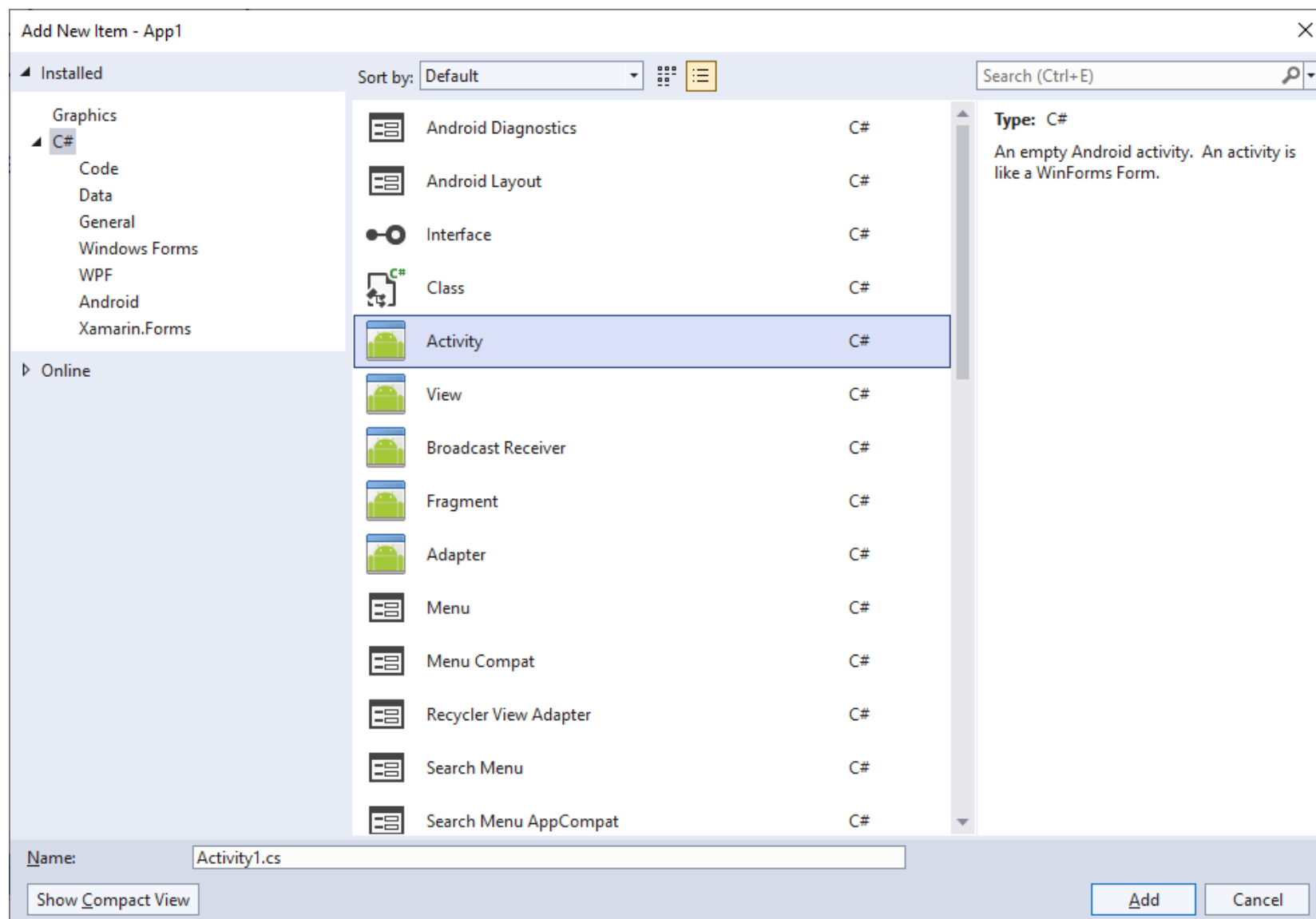
Фиг. 2. Създаване на мобилен проект.



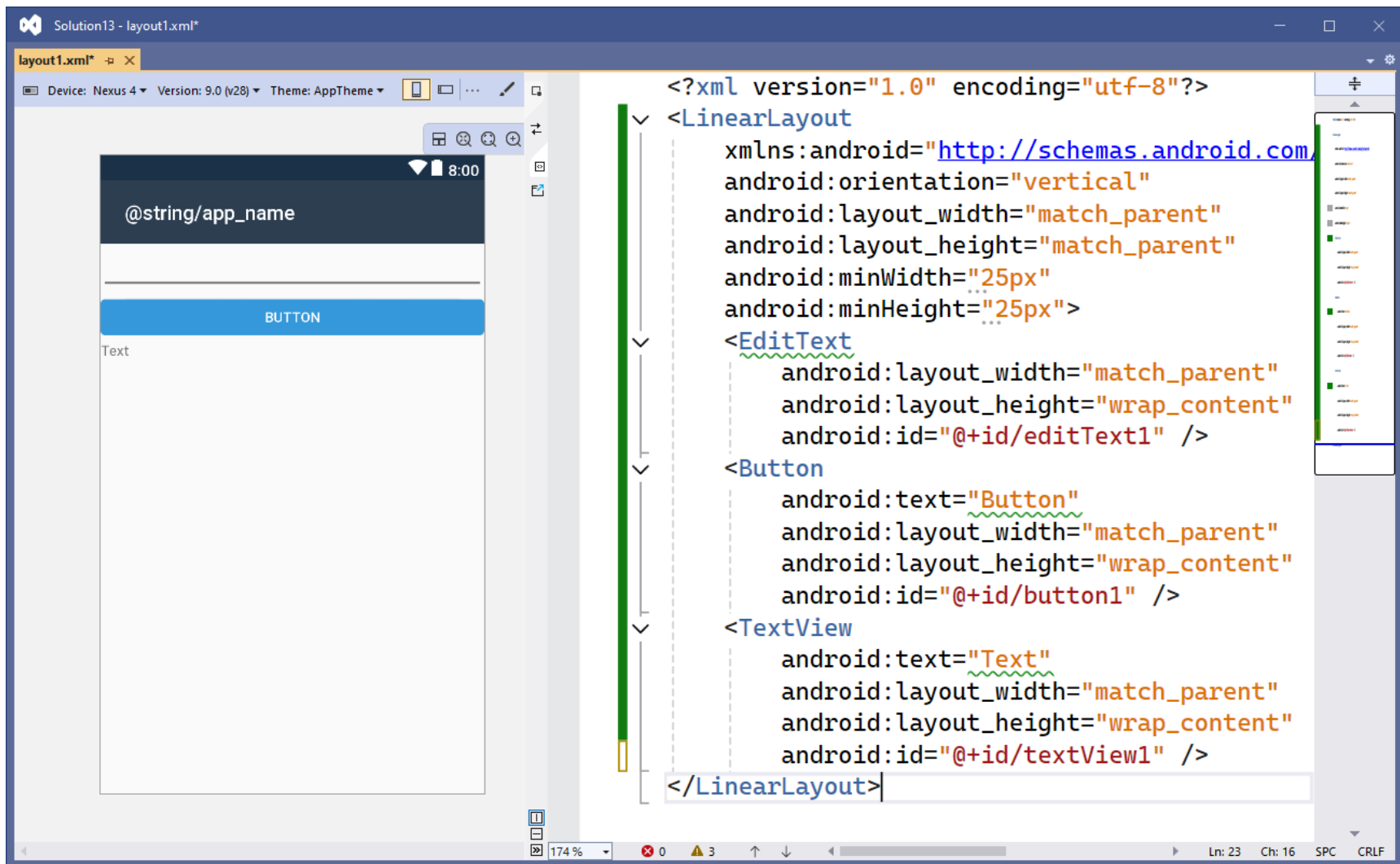
Фиг. 3. Избор на шаблон на проект.



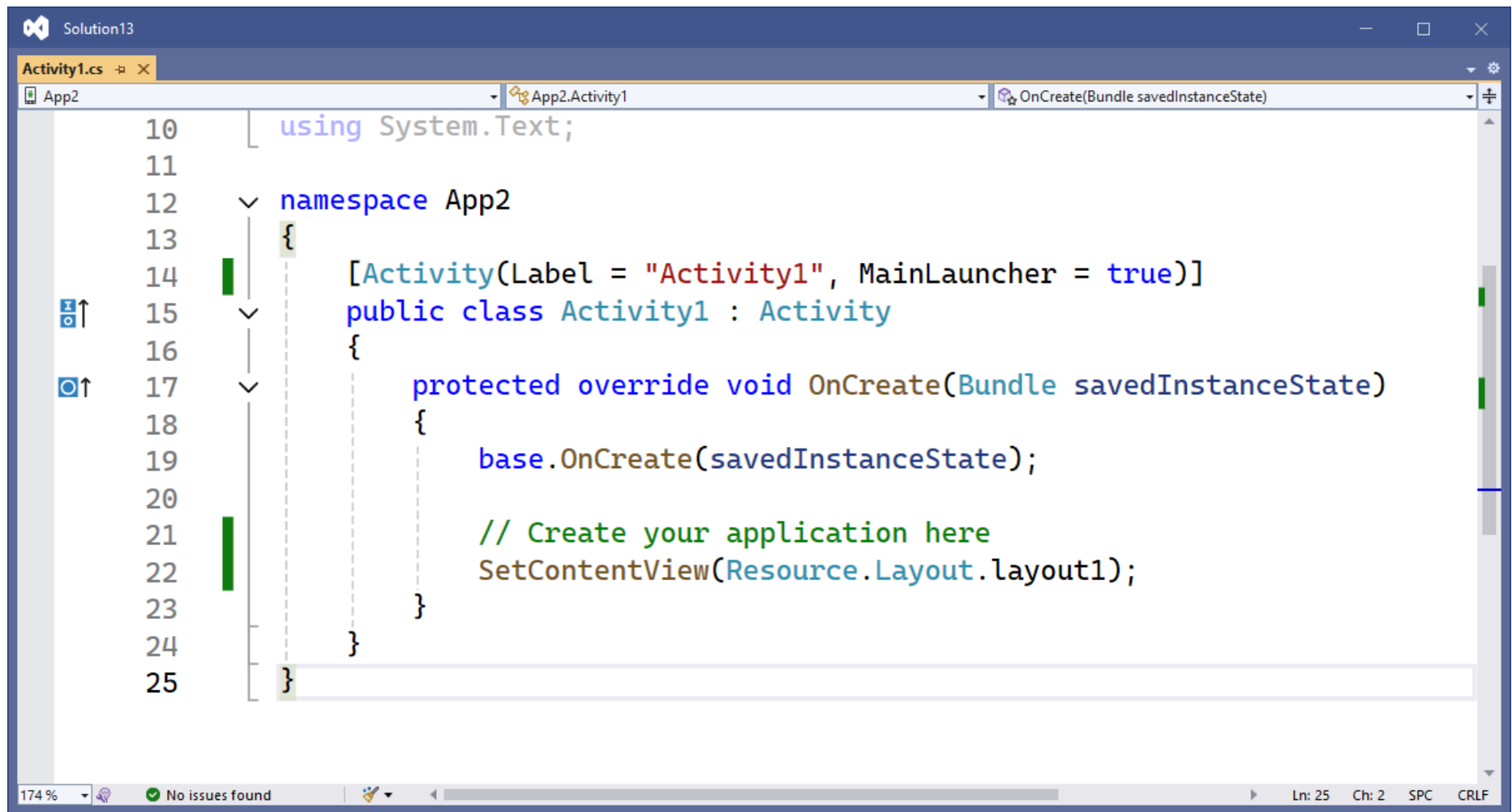
Фиг. 4. Добавяне на изглед, като слой.



Фиг. 5. Добавяне на Прозорец, като Activity.



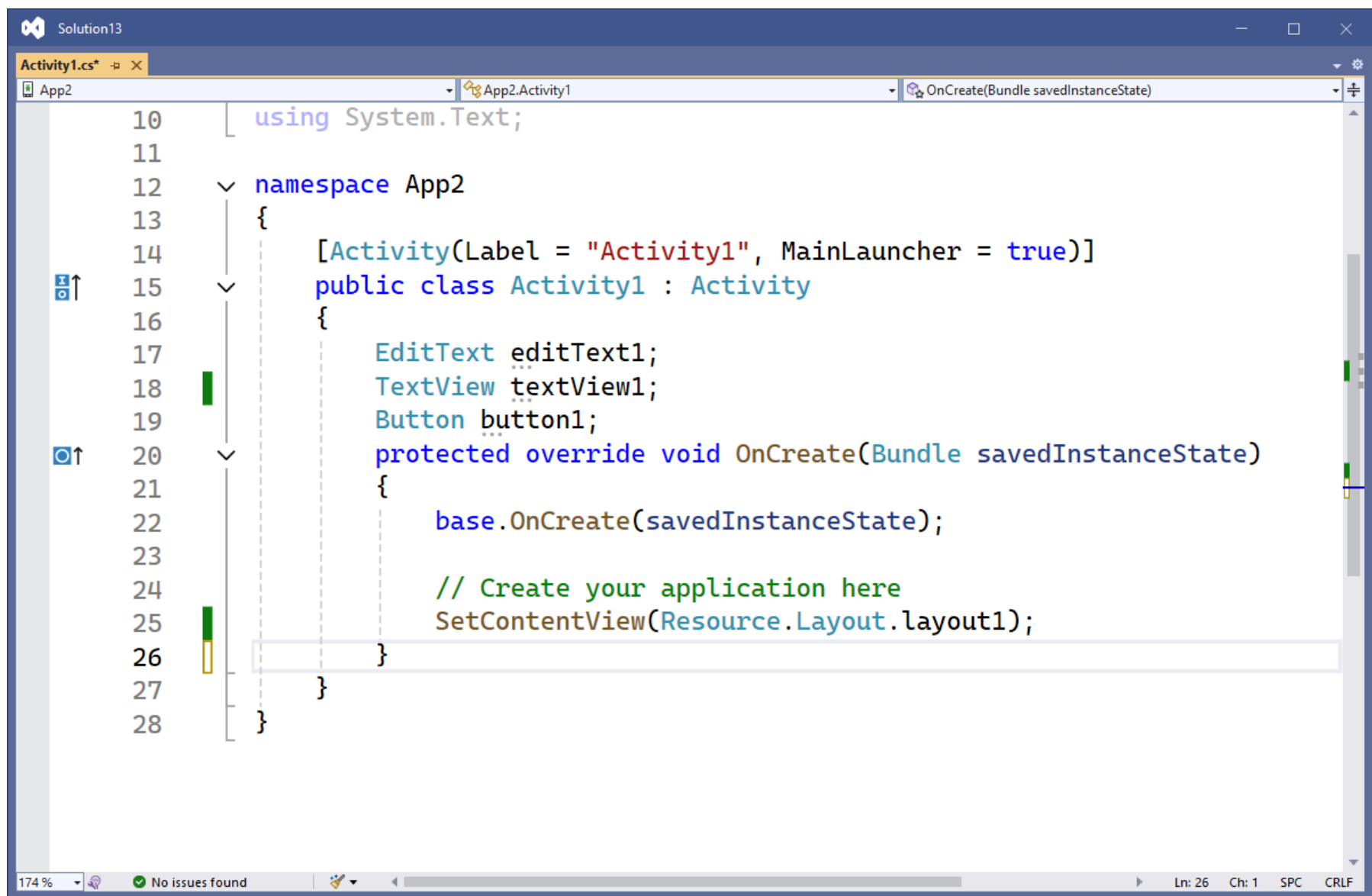
Фиг. 6. Добавяне на компоненти към изглед.



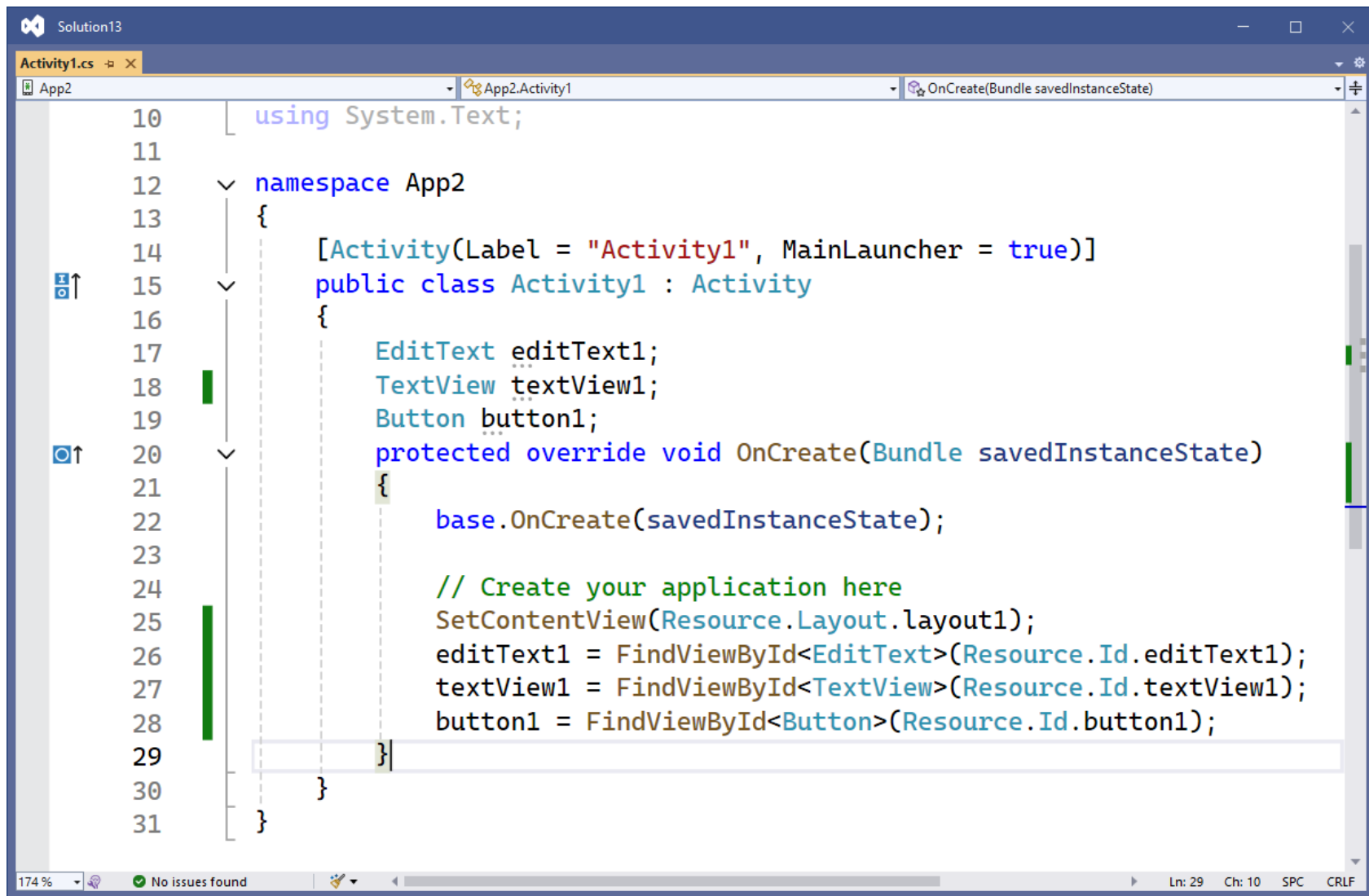
```
10 using System.Text;
11
12 namespace App2
13 {
14     [Activity(Label = "Activity1", MainLauncher = true)]
15     public class Activity1 : Activity
16     {
17         protected override void OnCreate(Bundle savedInstanceState)
18         {
19             base.OnCreate(savedInstanceState);
20
21             // Create your application here
22             SetContentView(Resource.Layout.layout1);
23         }
24     }
25 }
```

174 % No issues found Ln: 25 Ch: 2 SPC CRLF

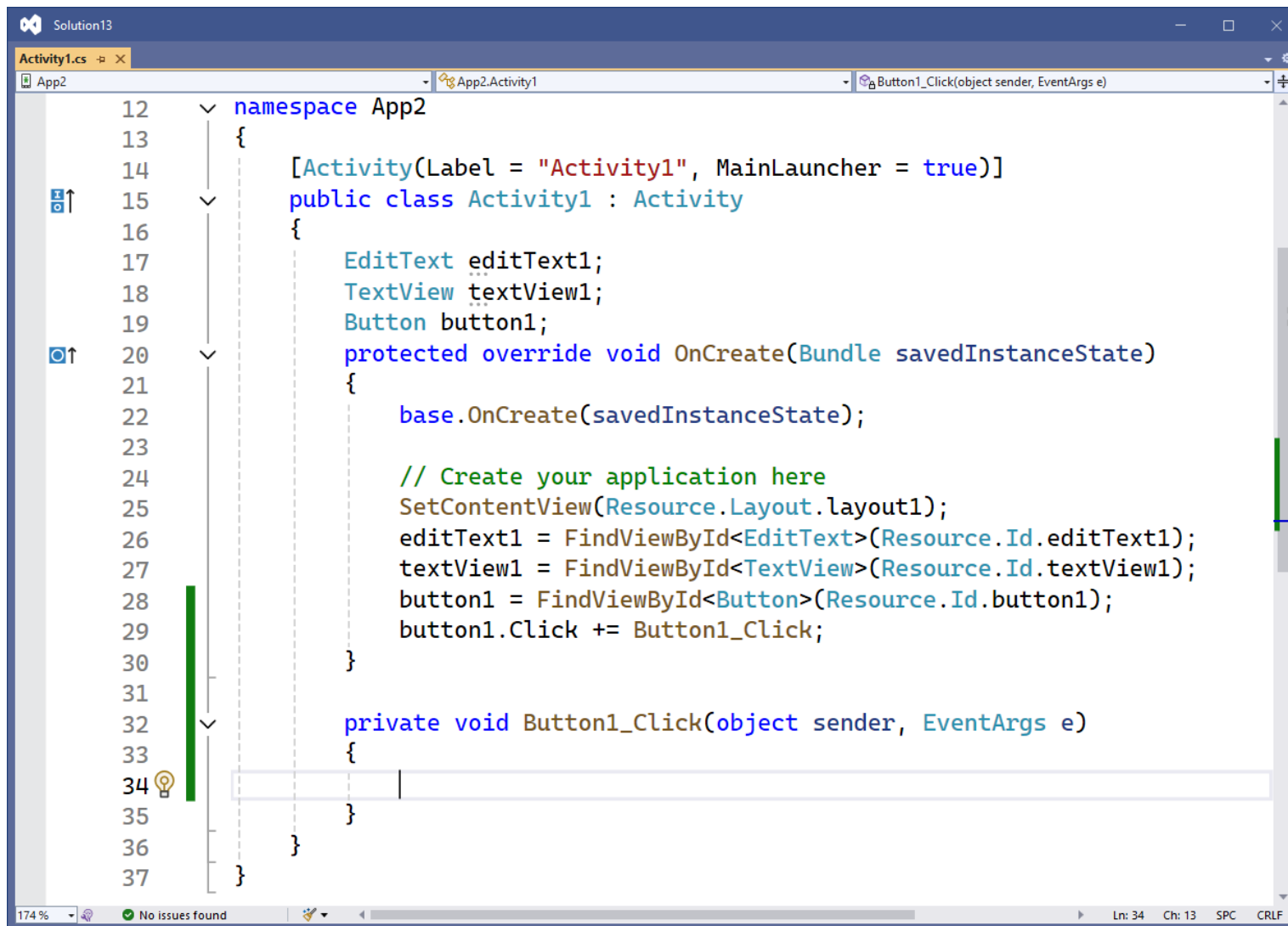
Фиг. 7. Свързване на Изглед и Активити.



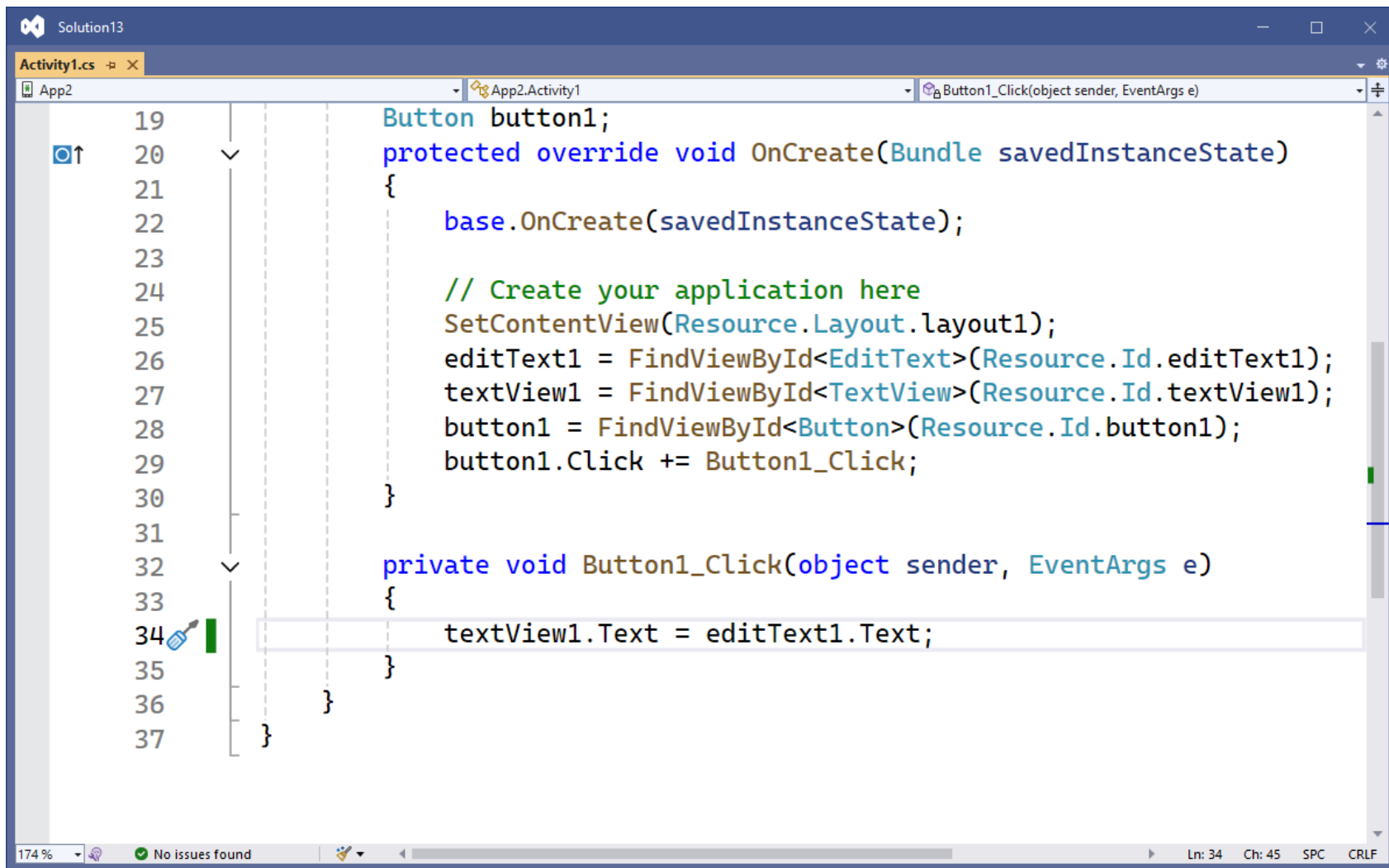
Фиг. 8. Добавяне на компоненти към Activity.



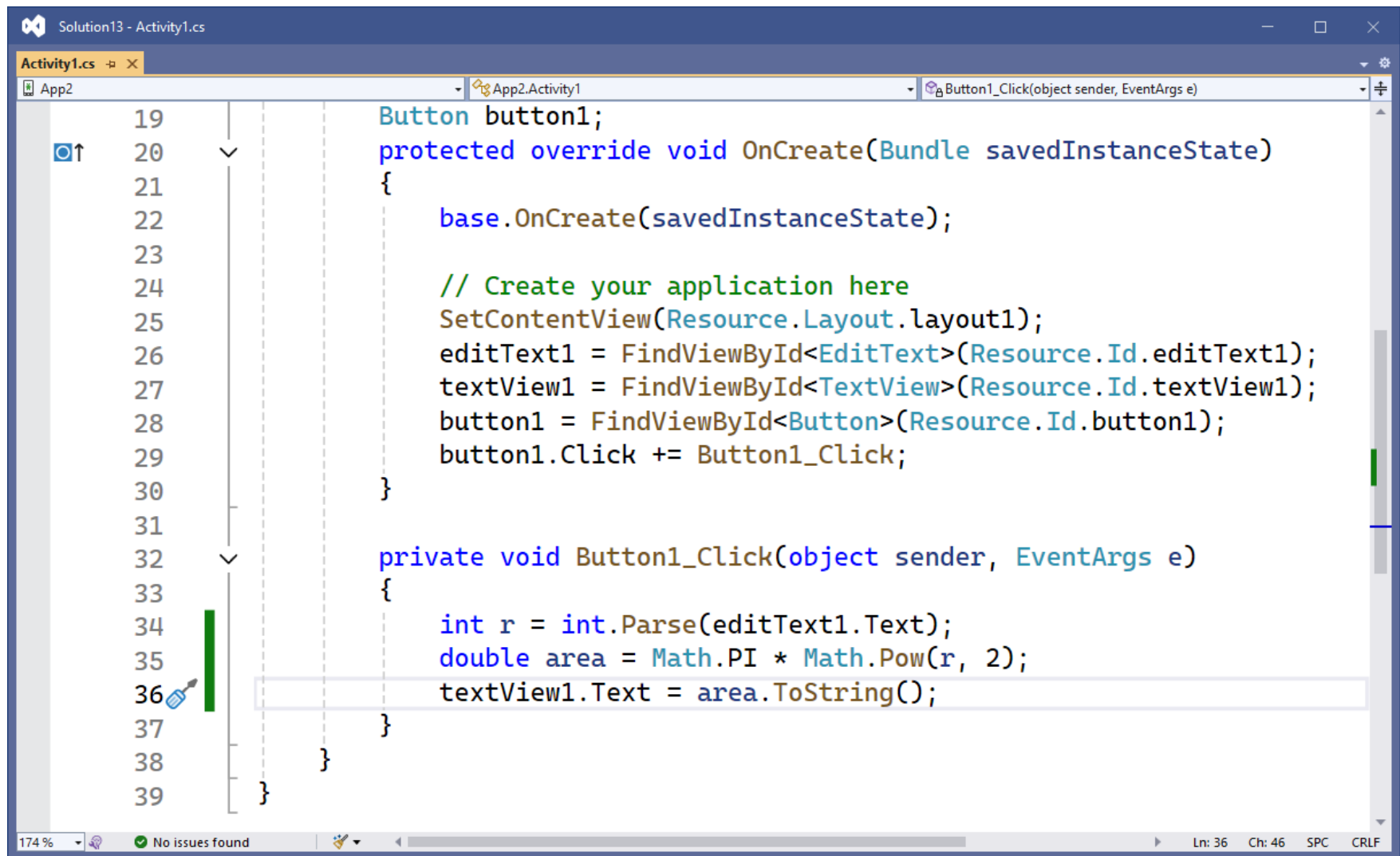
Фиг. 9. Инициализиране на компонентите от изгледа.



Фиг. 10. Добавяне на събитие към бутона.



Фиг. 11. Добавяне на код към събитие.

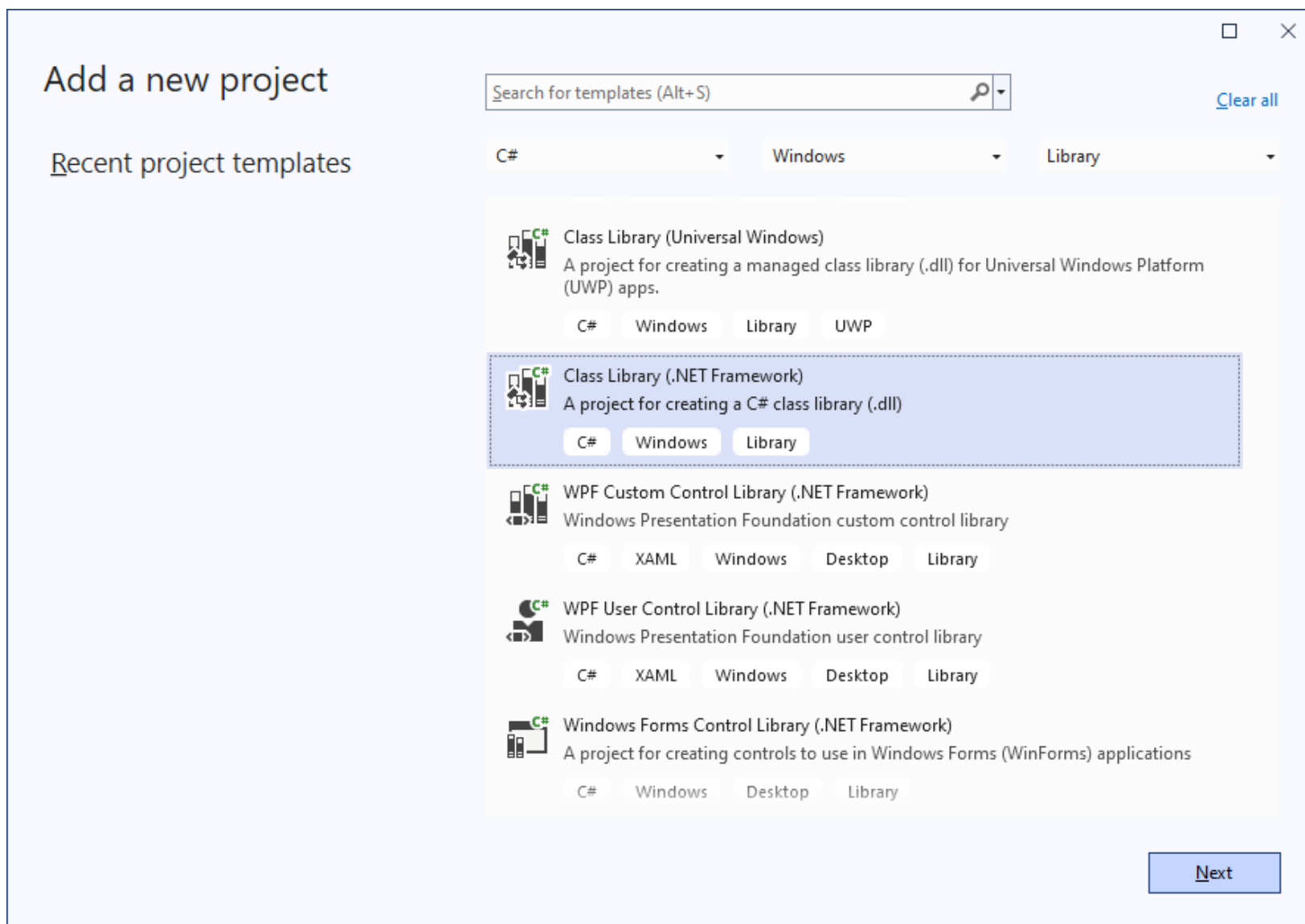


```
Solution13 - Activity1.cs
Activity1.cs
App2
App2.Activity1
Button1_Click(object sender, EventArgs e)

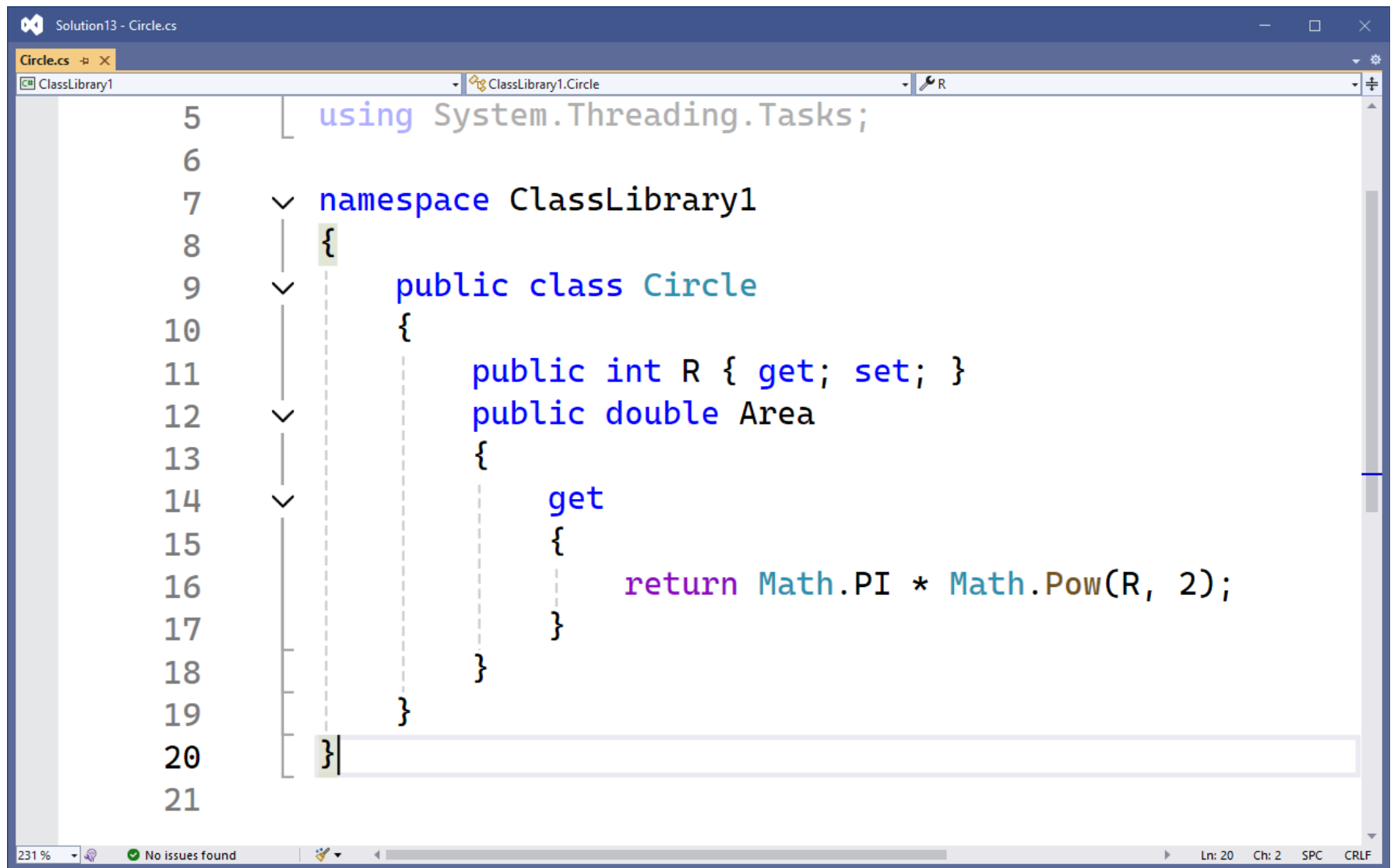
19 Button button1;
20 protected override void onCreate(Bundle savedInstanceState)
21 {
22     base.onCreate(savedInstanceState);
23
24     // Create your application here
25     setContentView(Resource.Layout.layout1);
26     editText1 = FindViewById<EditText>(Resource.Id.editText1);
27     textView1 = FindViewById<TextView>(Resource.Id.textView1);
28     button1 = FindViewById<Button>(Resource.Id.button1);
29     button1.Click += Button1_Click;
30 }
31
32 private void Button1_Click(object sender, EventArgs e)
33 {
34     int r = int.Parse(editText1.Text);
35     double area = Math.PI * Math.Pow(r, 2);
36     textView1.Text = area.ToString();
37 }
38 }
39 }
```

174 % No issues found Ln: 36 Ch: 46 SPC CRLF

Фиг. 12. Добавяне на по-сложен код към събитие.



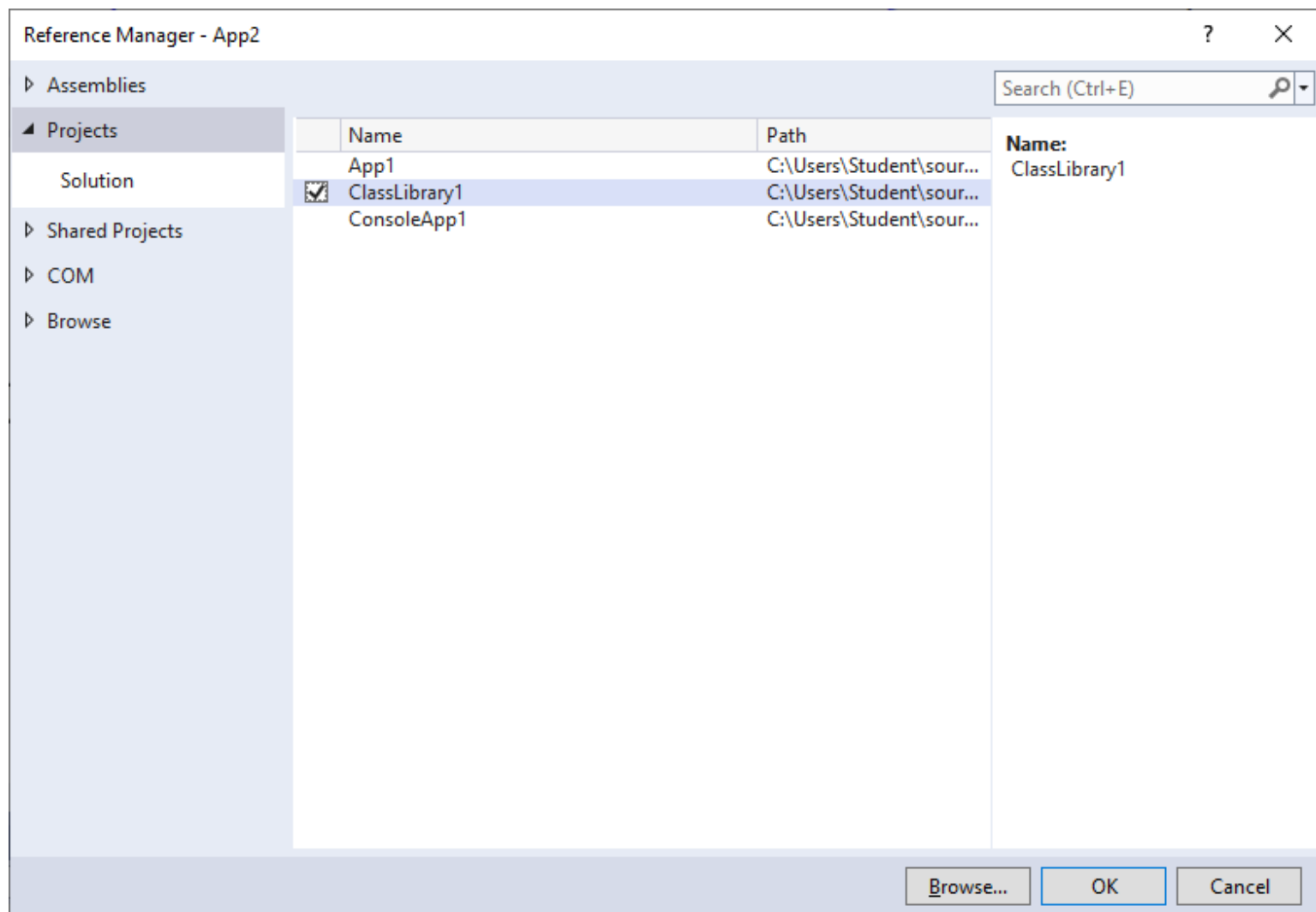
Фиг. 13. Добавяне на библиотека.



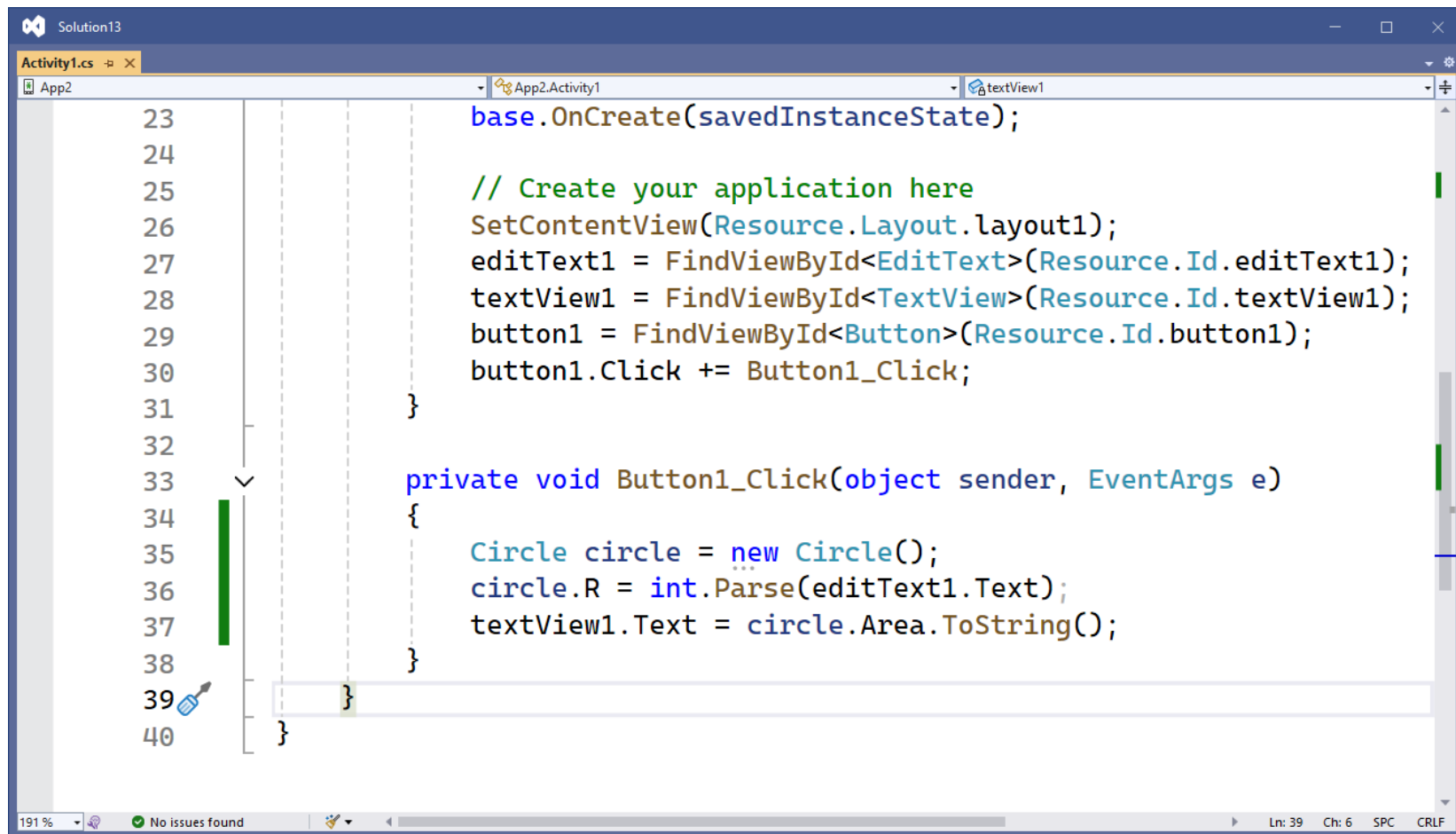
```
5  using System.Threading.Tasks;
6
7  namespace ClassLibrary1
8  {
9      public class Circle
10     {
11         public int R { get; set; }
12         public double Area
13         {
14             get
15             {
16                 return Math.PI * Math.Pow(R, 2);
17             }
18         }
19     }
20 }
21
```

The screenshot shows a Visual Studio window with the title 'Solution13 - Circle.cs'. The file explorer on the left shows 'ClassLibrary1' and 'ClassLibrary1.Circle'. The code editor displays the following C# code: Line 5: 'using System.Threading.Tasks;'. Line 7: 'namespace ClassLibrary1'. Line 8: '{'. Line 9: 'public class Circle'. Line 10: '{'. Line 11: 'public int R { get; set; }'. Line 12: 'public double Area'. Line 13: '{'. Line 14: 'get'. Line 15: '{'. Line 16: 'return Math.PI * Math.Pow(R, 2);'. Line 17: '}'. Line 18: '}'. Line 19: '}'. Line 20: '}'. Line 21: The cursor is at the end of line 20. The status bar at the bottom shows '231 %', 'No issues found', and 'Ln: 20 Ch: 2 SPC CRLF'.

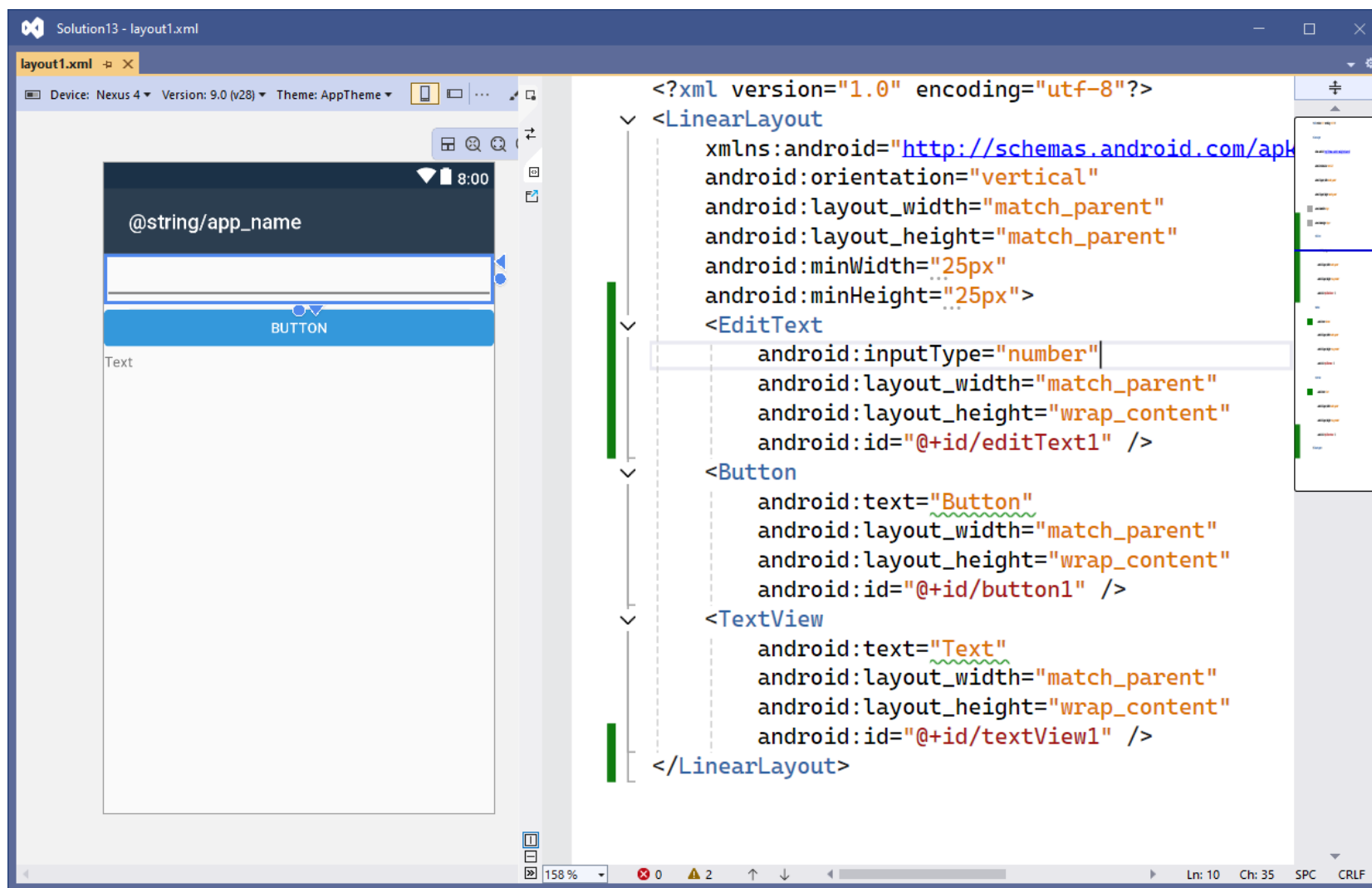
Фиг. 14. Добавяне на модел в библиотеката.



Фиг. 15. Добавяне референция към библиотека в проект.



Фиг. 16. Използване на модел от библиотека в събитие.



Фиг. 17. Добавяне на контрол на входните данни (където е курсора).