

```
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Windows.Forms;
5
6 namespace Lab5
7 {
8     class OperationPanel : Panel
9     {
10         public enum Operation { Sum, Sub, Mult, Div }
11         private Operation _currentOperation;
12         private readonly TextBox[] _textBoxes = new TextBox[3];
13         private readonly Label[] _labels = new Label[3];
14         private Label _labelOperator, _labelEquals;
15         private Button _buttonOperation;
16         public Operation OperationType
17         {
18             set
19             {
20                 _currentOperation = value;
21                 UpdateOperationText();
22             }
23             get { return _currentOperation; }
24         }
25
26         public OperationPanel()
27         {
28             InitializeComponent();
29         }
30
31         private void InitializeComponent()
32         {
33             Size = new Size(550, 150);
34             BackColor = Color.WhiteSmoke;
35             AutoSize = true;
36             AutoSizeMode = AutoSizeMode.GrowAndShrink;
37             List<string> labelTexts = new List<string> { "العدد", "النتائج", "العدد الثاني" };
38
39             for (int i = 0; i < 3; i++)
40             {
41                 _labels[i] = CreateLabel(labelTexts[i], new Point((110 + 100) * i + 10, 13));
42                 _textBoxes[i] = CreateTextBox(new Point((80 + 120) * i + 10, 50));
43                 Controls.Add(_labels[i]);
44                 Controls.Add(_textBoxes[i]);
45             }
46         }
47     }
48 }
```

```
47         _labelOperator = new Label { Text = "+", Location = new Point(350, 51) };
48         _labelEquals = new Label { Text = "=", Location = new Point(150, 51) };
49         _buttonOperation = new Button { Text = "جمع", Location = new Point(380, 94), Size = new Size(100, 30) };
50         _buttonOperation.Click += OnButtonOperationClick;
51         Controls.AddRange(new Control[] { _labelOperator, _labelEquals, _buttonOperation });
52     }
53
54     private Label CreateLabel(string text, Point location)
55     {
56         return new Label
57         {
58             Size = new Size(100, 23),
59             Location = location,
60             Text = text
61         };
62     }
63
64     private TextBox CreateTextBox(Point location)
65     {
66         return new TextBox
67         {
68             Size = new Size(120, 20),
69             Location = location
70         };
71     }
72
73     private void UpdateOperationText()
74     {
75         switch (_currentOperation)
76         {
77             case Operation.Sum:
78                 _labelOperator.Text = "+";
79                 _buttonOperation.Text = "جمع";
80                 break;
81             case Operation.Sub:
82                 _labelOperator.Text = "-";
83                 _buttonOperation.Text = "طرح";
84                 break;
85             case Operation.Mult:
86                 _labelOperator.Text = "*";
87                 _buttonOperation.Text = "ضرب";
88                 break;
89             case Operation.Div:
90                 _labelOperator.Text = "/";
91                 _buttonOperation.Text = "قسمة";
```

```
92         break;
93     }
94 }
95
96 private void OnButtonOperationClick(object sender, EventArgs e)
97 {
98     if (ValidateInput())
99     {
100         float n1 = float.Parse(_textBoxes[2].Text.Trim());
101         float n2 = float.Parse(_textBoxes[1].Text.Trim());
102         float result = CalculateResult(n1, n2);
103         _textBoxes[0].Text = result.ToString();
104     }
105     else
106     {
107         MessageBox.Show("ادخل رقم..");
108     }
109 }
110
111 private bool ValidateInput()
112 {
113     return !string.IsNullOrEmpty(_textBoxes[2].Text) && !
114         string.IsNullOrEmpty(_textBoxes[1].Text);
115 }
116
117 private float CalculateResult(float n1, float n2)
118 {
119     switch (_currentOperation)
120     {
121         case Operation.Sum: return n1 + n2;
122         case Operation.Sub: return n1 - n2;
123         case Operation.Mult: return n1 * n2;
124         case Operation.Div: return n2 != 0 ? n1 / n2 : 0;
125         default: return 0;
126     }
127 }
128
129 public double TextN1
130 {
131     get { return double.Parse(_textBoxes[2].Text); }
132     set { _textBoxes[2].Text = value.ToString(); }
133 }
134
135 public double TextN2
136 {
137     get { return double.Parse(_textBoxes[1].Text); }
138     set { _textBoxes[1].Text = value.ToString(); }
139 }
```

---

```
140         public double TextN3 => double.Parse(_textBoxes[0].Text);
141     }
142 }
143
```