2 14

2019 7 18

1

1

18x0123		z1021234	100	S

C# structure

C# r turn 1

r turn

C#

1)
2)
3)

1)

```
struct Student
{
    public String id;
    public String name;
    public String class;
    public int point;
    public char eval;
}
```

2)

```
Student st;
```

3)

```
st.id = "18x0123";
st.name = "                ";
st.class = "z1021234";
st.point = 100;
st.eval   = 'S';
```

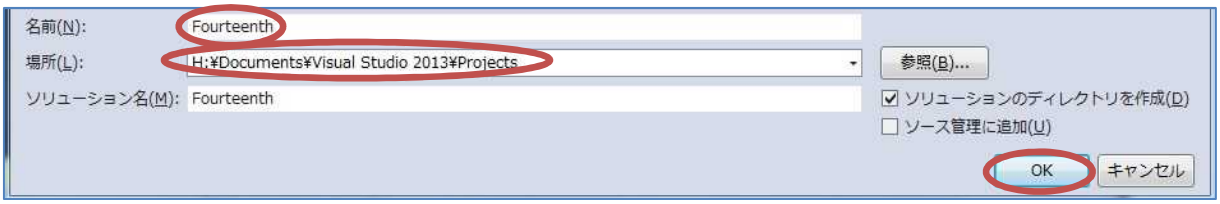
2　14

1)

Visual Studio 2013　[　]　→[　]　→[　]

Visual C#　Windows

Fourteenth　H:¥Documents¥Visual Studio 2013¥Projects　OK　1



2)

14

Form1　3　3　3

Form1　Text

label1　Text

0

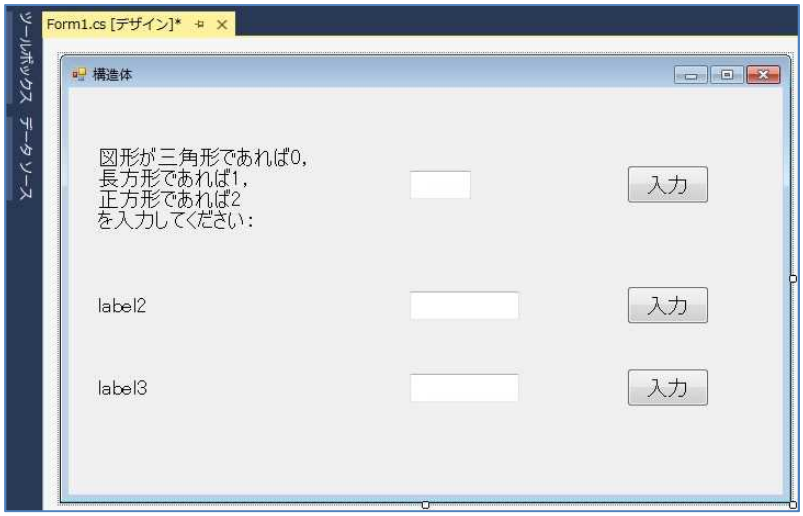
1

2

button1　Text

button2　Text

button3　Text



3)

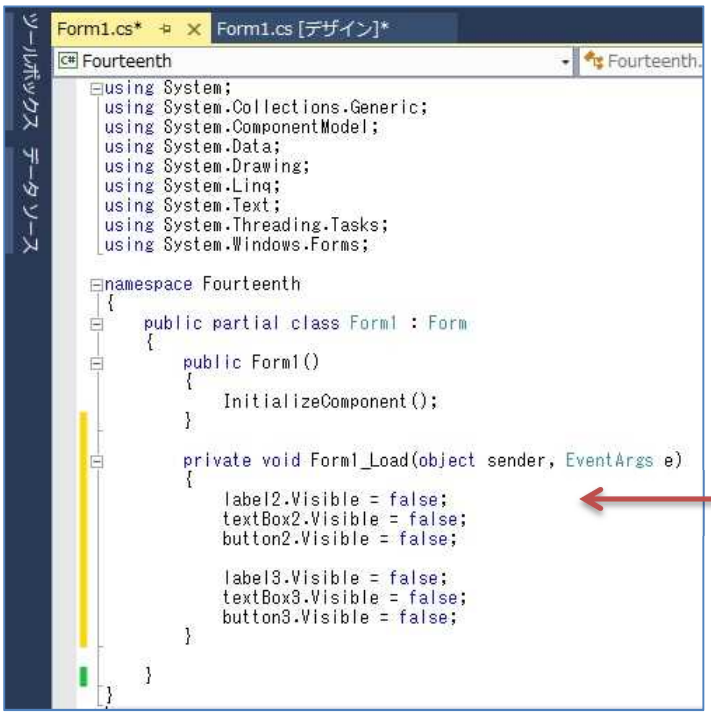
Form1

Form1.cs　Form1_Load　Form1

label2　textBox2　button2　label3　textBox3

button3　Visible　false

```
private void Form1_Load(object sender, EventArgs e)
{
    label2.Visible = false;
    textBox2.Visible = false;
    button2.Visible = false;

    label3.Visible = false;
    textBox3.Visible = false;
    button3.Visible = false;
}
```

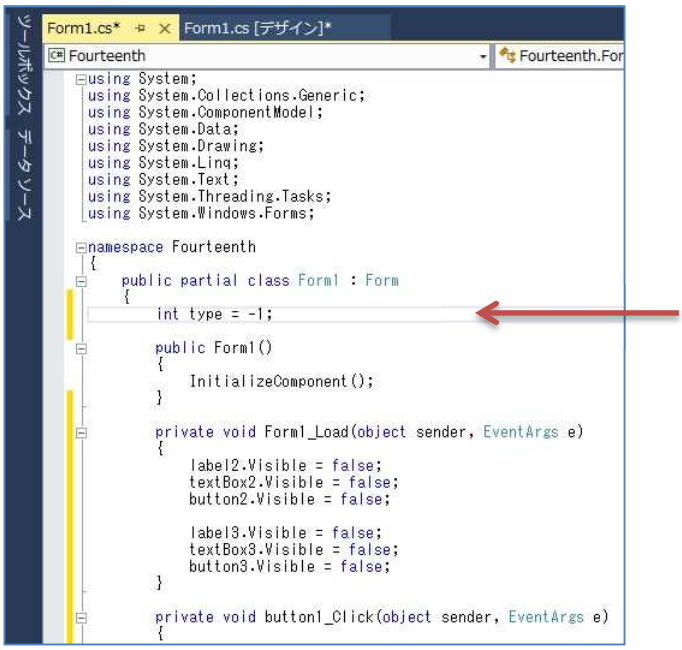


Form1

button1

Form1.cs

```
int type = -1;
```

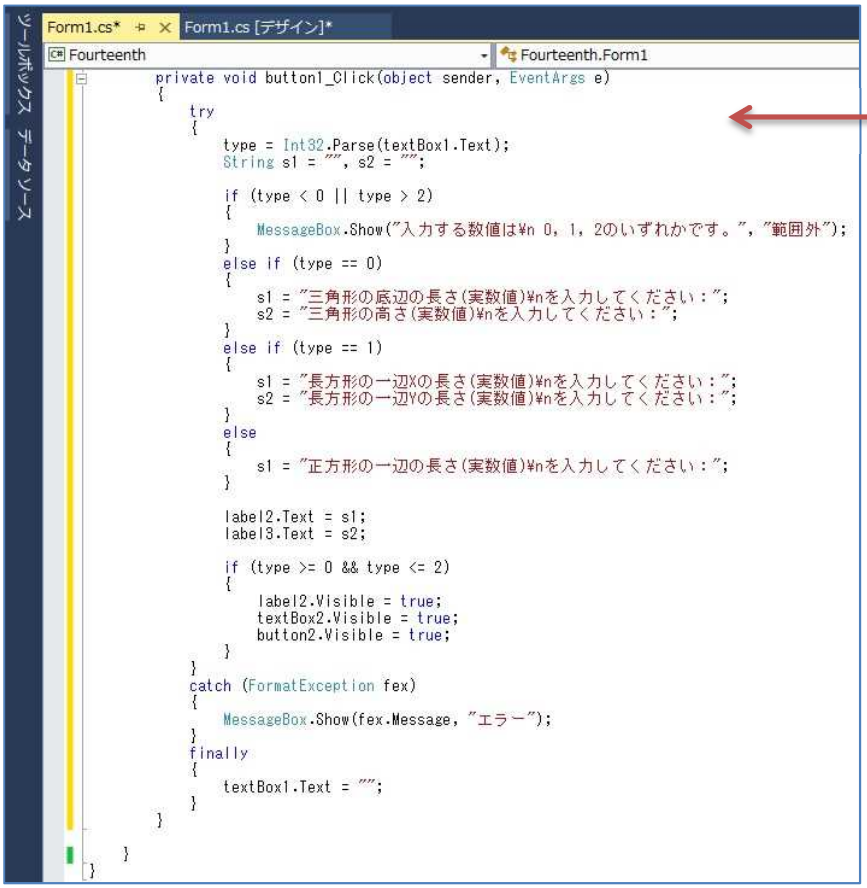


2 14

button1_Click

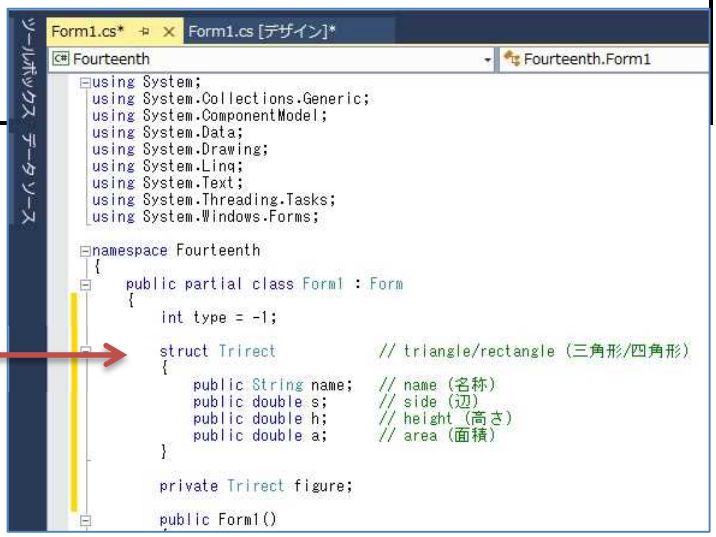
```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        type = Int32.Parse(textBox1.Text);
        String s1 = "", s2 = "";

        if (type < 0 || type > 2)
        {
            MessageBox.Show("                \n 0  1  2                ", "      ");
        }
        else if (type == 0)
        {
            s1 = "                    (       )\n                      ";
            s2 = "             (       )\n                      ";
        }
        else if (type == 1)
        {
            s1 = "            X         (       )\n                      ";
            s2 = "            Y         (       )\n                      ";
        }
        else
        {
            s1 = "                    (       )\n                      ";
        }

        label2.Text = s1;
        label3.Text = s2;

        if (type >= 0 && type <= 2)
        {
            label2.Visible = true;
            textBox2.Visible = true;
            button2.Visible = true;
        }
    }
    catch (FormatException ex)
    {
        MessageBox.Show(ex.Message, "      ");
    }
    finally
    {
        textBox1.Text = "";
    }
}
```

2　　14



Form1　　button2　　button3　　Form1.cs

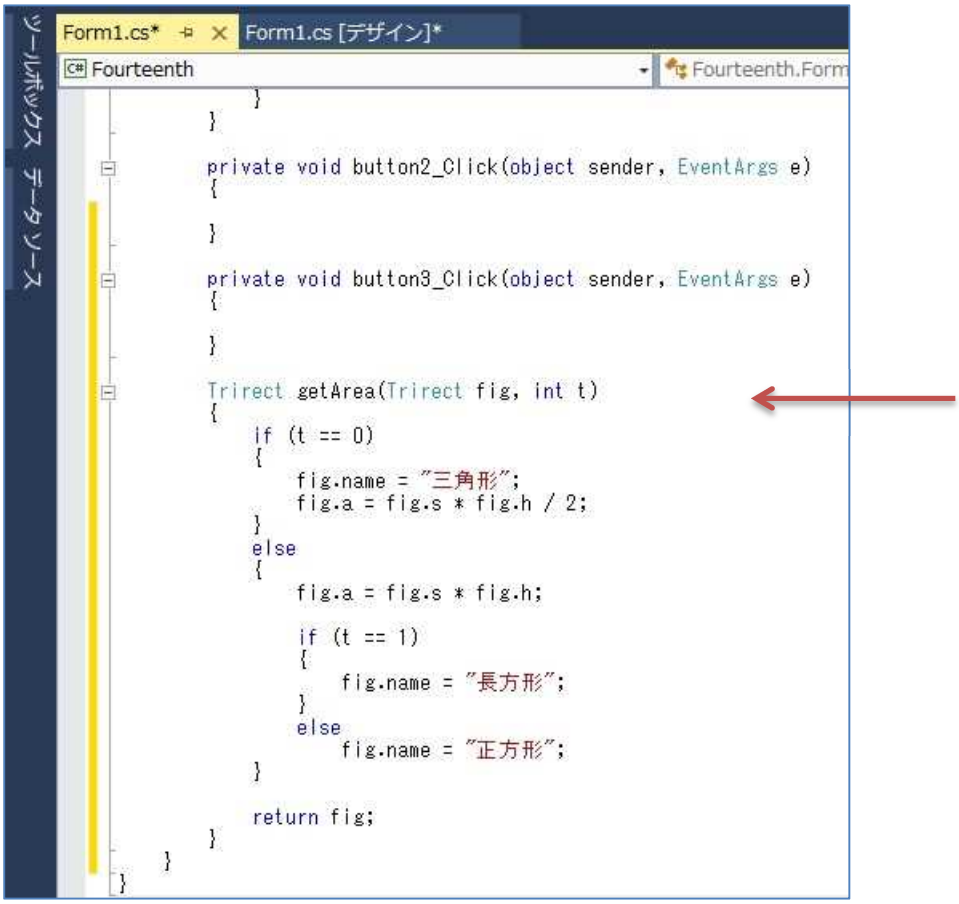
```
struct Trirect              //triangle/rectangle (　　/　　)
 {
    public String name;     // name (　　)
    public double s;        //side (　)
    public double h;        // height (　　)
    public double a;        //area (　　)
}

private Trirect figure;
```



2　　14

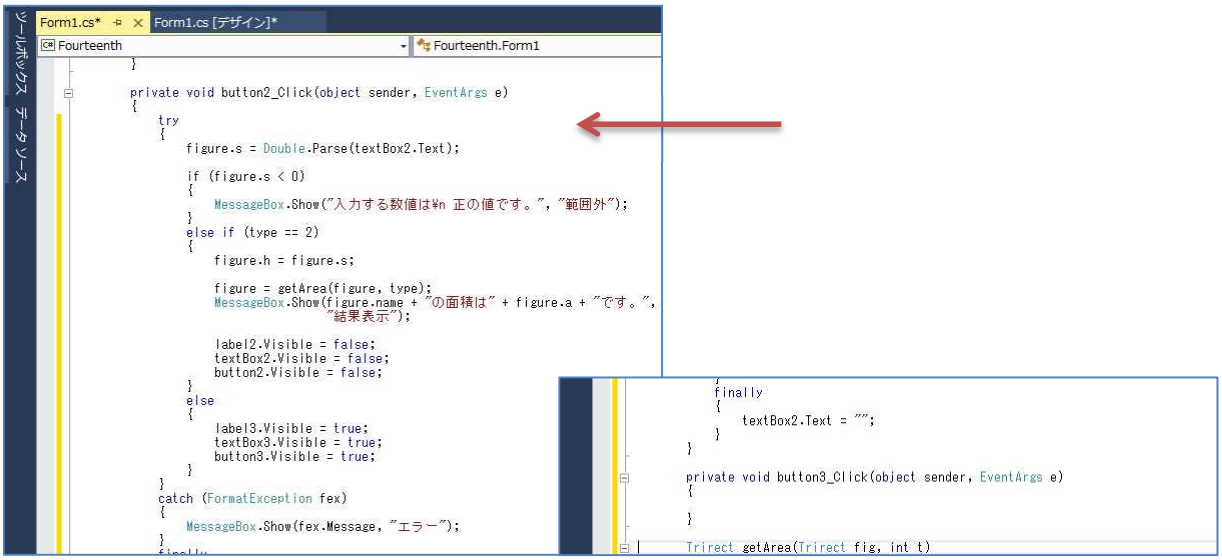
getArea

```
Tirect getArea(Tirect fig,int t)
{
  if (t == 0)
   {
     fig.name = "三角形";
     fig.a = fig.s * fig.h / 2;
   }
   else
   {
     fig.a = fig.s * fig.h;

     if (t == 1)
      {
        fig.name = "長方形";
      }
      else
        fig.name = "正方形";
   }

   return fig;
}
```



button2_Click

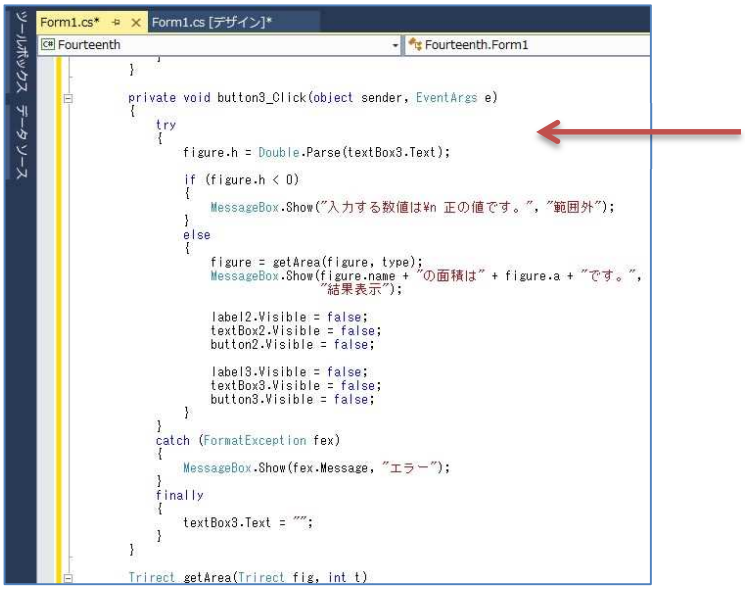
```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        figure.s = Double.Parse(textBox2.Text);

        if (figure.s < 0)
        {
            MessageBox.Show("入力する数値は¥n 正の値です。", "範囲外");
        }
        else if (type == 2)
        {
            figure.h = figure.s;

            figure = getArea(figure, type);
            MessageBox.Show(figure.name + "の面積は" + figure.a + "です。",
                            "結果表示");

            label2.Visible = false;
            textBox2.Visible = false;
            button2.Visible = false;
        }
        else
        {
            label3.Visible = true;
            textBox3.Visible = true;
            button3.Visible = true;
        }
    }
    catch (FormatException fex)
    {
        MessageBox.Show(fex.Message, "エラー");
    }
    finally
    {
        textBox2.Text = "";
    }
}
```



2　　14

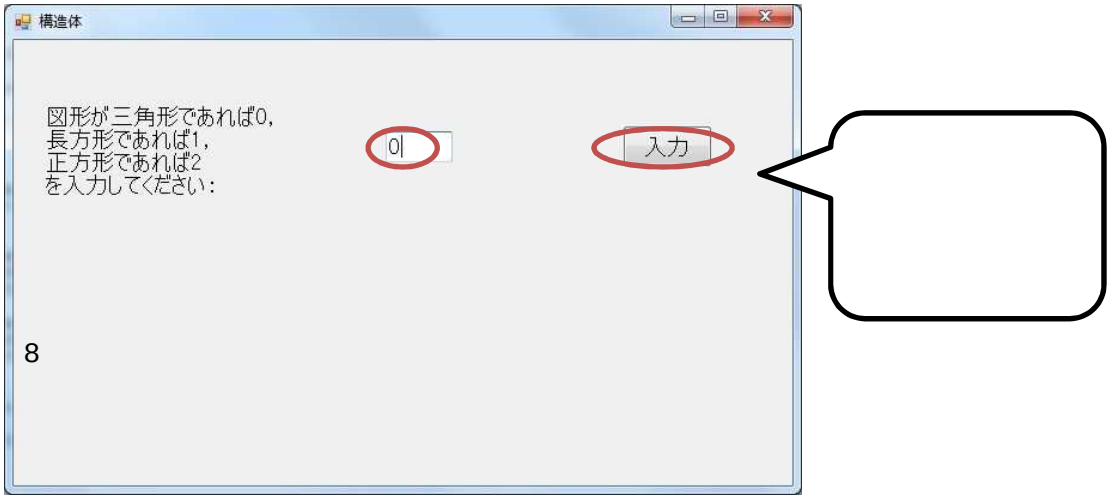
button3_Click

```
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        figure.h = Double.Parse(textBox3.Text);

        if (figure.h < 0)
        {
            MessageBox.Show("入力する数値は¥n 正の値です。", "範囲外");
        }
        else
        {
            figure = getArea(figure, type);
            MessageBox.Show(figure.name + "の面積は" + figure.a + "です。",
                            "結果表示");

            label2.Visible = false;
            textBox2.Visible = false;
            button2.Visible = false;

            label3.Visible = false;
            textBox3.Visible = false;
            button3.Visible = false;
        }
    }
    catch (FormatException fex)
    {
        MessageBox.Show(fex.Message, "エラー");
    }
    finally
    {
        textBox3.Text = "";
    }
}
```

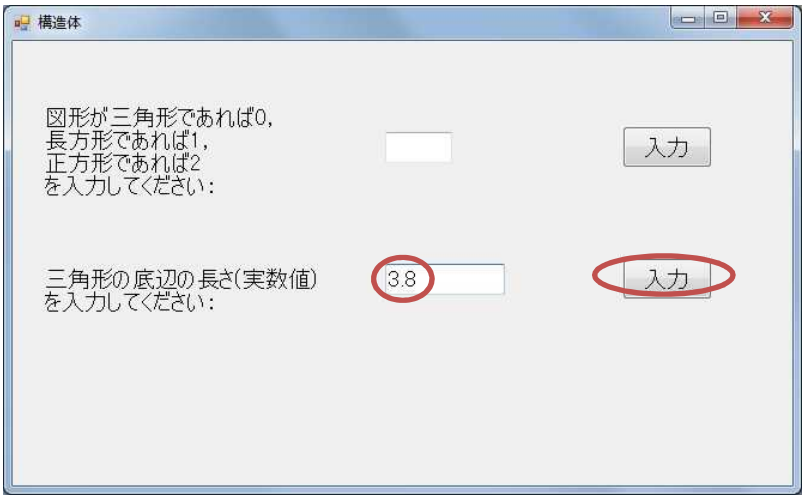


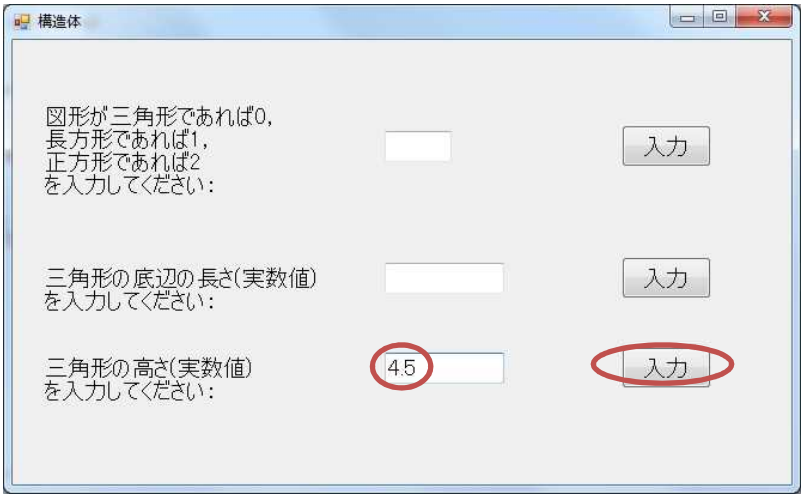
2　　14

4)


構造体
図形が三角形であれば0,
長方形であれば1,
正方形であれば2
を入力してください:
0
入力

8


構造体
図形が三角形であれば0,
長方形であれば1,
正方形であれば2
を入力してください:
入力
三角形の底辺の長さ(実数値)
を入力してください:
3.8
入力


構造体
図形が三角形であれば0,
長方形であれば1,
正方形であれば2
を入力してください:
入力
三角形の底辺の長さ(実数値)
を入力してください:
入力
三角形の高さ(実数値)
を入力してください:
4.5
入力


結果表示
三角形の面積は8.55です。
OK

2　　14

H:¥Documents¥Visual Studio 2013¥Projects¥Fourteenth¥Fourteenth

1) Form1.Designer.cs
2) Form1.cs
3) Questions_14th.txt