

Протокол №8

Тема:

Прихващане и обработка на грешки

Дата:

Изготвил:

Въпроси и задачи:

Оператор **try ... catch ... finally**

В езика C# с оператора `try ... catch ... finally` може да се обработват изключения (exception), т.е. ситуации, които ако не бъдат обработени, ще предизвикат аварийно спиране на програмата.

Синтаксис:

`try` { < оператори, при изпълнение на които се очаква изключение > }

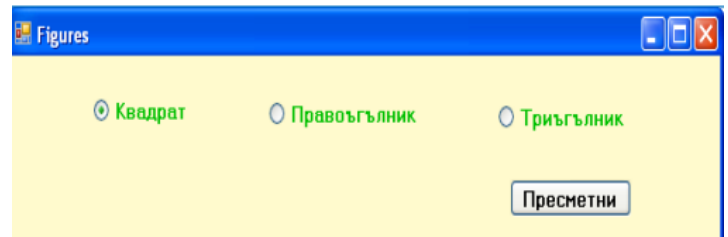
`catch` (< вид на изключението >)

{ < оператори, които ще се изпълнят при възникне на изключение > }

`finally` { < оператори, които ще се изпълнят винаги > }

Когато програмата достигне този оператор, се опитва да изпълни операторите в блока `try`. Ако изключението се случи, се изпълнява кодът след `catch` (хващам, улавям), а след това кодът след `finally` (накрая). Ако очакваното изключение не се случи, се изпълнява само кодът след `finally`. Частите `catch` и `finally` не са задължителни, може да липсват.

Задача. 1 Създайте приложение, което да пресмята лице и периметър на квадрат, правоъгълник и триъгълник. В зависимост от вида на фигурата да се изберат необходимите входни данни. При първоначалният интерфейс да се предостави възможност за избор на вида на фигурата.



- 1) Проектирайте дизайна на формата, както е показано на фигурата.
- 2) Програмирайте функционалността на приложението.

```
private void buttonOut_Click(object sender, EventArgs e)
{
    FormFigures.ActiveForm.Close();
}
```

```
private void radioButKv_Click(object sender, EventArgs e)
{
    textBoxA.Visible = true; labelA.Visible = true;
    textBoxB.Visible = false; labelB.Visible = false;
    textBoxC.Visible = false; labelC.Visible = false;
}

private void radioButPr_Click(object sender, EventArgs e)
{
    textBoxA.Visible = true; labelA.Visible = true;
    textBoxB.Visible = true; labelB.Visible = true;
    textBoxC.Visible = false; labelC.Visible = false;
}

private void radioButTr_Click(object sender, EventArgs e)
{
    textBoxA.Visible = true; labelA.Visible = true;
    textBoxB.Visible = true; labelB.Visible = true;
    textBoxC.Visible = true; labelC.Visible = true;
}
```

```
private void buttonRez_Click(object sender, EventArgs e)
{
    int a, b, c;
    try { a = int.Parse(textBoxA.Text); }
    catch (FormatException)
    {
        a = 0;
        textBoxA.Text = "0";
    }
    try { b = int.Parse(textBoxB.Text); }
    catch (FormatException)
```

```
{
    b = 0;
    textBoxB.Text = "0";
}
try { c = int.Parse(textBoxC.Text); }
catch (FormatException)
{
    c = 0;
    textBoxC.Text = "0";
}
double S, P, p;
if (radioButKv.Checked)
{ S = a * a; P = 4 * a; }
else if (radioButPr.Checked)
{ S = a * b; P = 2 * (a + b); }
else
{ P = a+b+c; p=P/2;
  S= Math.Sqrt(p * (p - a) * (p - b) * (p - c)); }
labelPerRez.Text = P.ToString();
labelSRez.Text = S.ToString();
}
private void buttonClear_Click(object sender, EventArgs e)
{
    textBoxA.Text = "";
    textBoxB.Text = "";
    textBoxC.Text = "";
    labelPerRez.Text = " 0";
    labelSRez.Text = " 0";
    textBoxA.Focus();
}
```

- 3) Тествайте програмата с различни входни данни.
- 4) Забелязахте ли нередности или изключения, които не са обработени?
- 5) Опишете резултатите от тестването и предложете средства за отстраняване на нередности.