

Протокол №5

Тема:

Работа с обекти и свързани с тях събития

Дата:

Изготвил:

Въпроси и задачи:

Задача 1. Създайте нов проект с графичен интерфейс и го съхранете под името My App.

- 1) От прозореца Properties променете свойството Text на формата в MyApp, и размерите ѝ на 350;200. Променете свойството на формата StartPosition на CenterScreen.
- 2) От кутията с компоненти, с хващане и влачене поставете в средата на формата бутон. От прозореца Properties променете надписа (свойството Text) на бутона в Click Me.
- 3) С двойно щракване върху бутона отворете прозореца с програмния код, където средата автоматично е добавила функцията `private void button1_Click()`, предназначена да обработи събитието Click – щракване върху бутона.

Напишете в тялото на функция

```
MessageBox.Show("Hellow Word!");
```

- 4) От кутията с компоненти, с хващане и влачене поставете във формата втори бутон. От прозореца Properties променете надписа (свойството Text) на бутона в Press Me, Visible = False.
- 5) С двойно щракване върху бутона отворете прозореца с програмния код, където средата автоматично е добавила функцията `private void button2_Click()`, предназначена да обработи събитието Click.

Напишете в тялото на функцията

```
MessageBox.Show("Hellow from button 2!");
```

- 6) Към тялото на функцията `private void button1_Click()`, добавете `button2.Visible = true`.

Компилирайте получената програма и я изпълнете.



Задача 2. Създайте приложение, съдържащо два командни бутона и един етикет. За свойство Text (надпис) на етикета задайте стойност **“Здравей свят!”**, а за двата бутона – съответно **“Покажи”** и **“Скрий”**. При щракване върху бутон **“Покажи”** се визуализира етикета с надпис **“Здравей свят!”**, а при щракване върху бутон **“Скрий”**

етикетът се скрива. В даден момент от изпълнение на задачата трябва да бъде достъпен само единият от тези два бутона, тъй като те изпълняват противоположни действия. При позициониране върху един от двата командни бутона, показалецът на мишката трябва да се променя в стрелка (т.е. във вида Arrow), а при позициониране върху формата – видът на показалеца на мишката трябва да бъде кръст (т.е. Cross). Съхранете проекта с име HideForm.

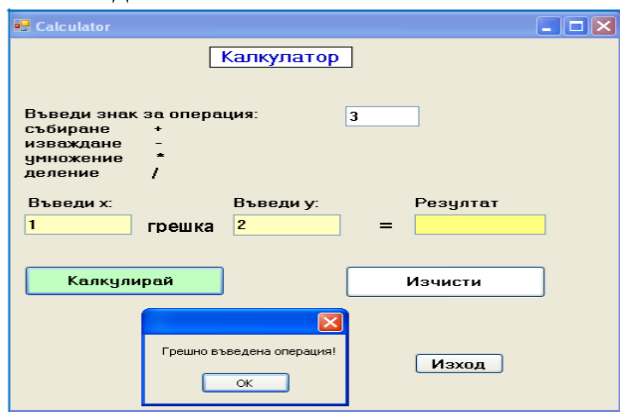
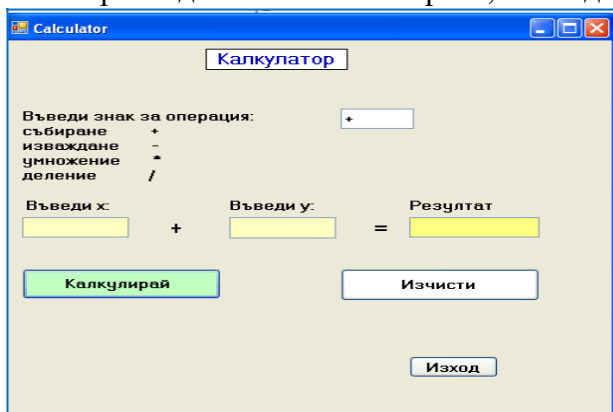
- 1) Създаване на ГПИ на приложението – добавяне на етикет и бутона.
- 2) Настройка на някои свойства на елементите на ГПИ в режим на проектиране.

Елементи на ГПИ	Име на елемент	Свойства
Label	label1	Text="Здравей свят!" ; Font, Bold, 14, BackColor – по избор
Button	buttonOn	Text="Покажи" ; Cursor=Arrow Enabled=false
Button	buttonOff	Text="Скрий" ; Cursor =Arrow Enabled=true
Form	Form1	Cursor =Cross; StartPosition=CenterScreen Text="HideForm"

- 3) Добавяне на програмнен код към елементите.

```
private void buttonOn_Click(object sender, EventArgs e)
{ label1.Visible = true; buttonOn.Enabled = false; buttonOff.Enabled = true; } private
void buttonOff_Click(object sender, EventArgs e)
{ label1.Visible = false; buttonOn.Enabled = true; buttonOff.Enabled = false; }
```

Задача 3. Като използвате създаденото приложение добавете нови компоненти – етикети, бутона и текстови кутии, за да реализирате прост калкулатор, който да извършва действията – събиране, изваждане, умножение и деление.



Примерен програмен код към елементите за реализирана функционалност

```
private void button1CalcSum_Click(object sender, EventArgs e)
{
    int rezultat = 0, x, y;
    x = int.Parse(textBox1Number1.Text);
    y = int.Parse(textBox2Number2.Text);
    switch (label1Sight.Text)
    {
        case "+": rezultat = x + y; break;
        case "-": rezultat = x - y; break;
        case "*": rezultat = x * y; break;
        case "/": rezultat = x / y; break;
        default:
            { labelBug.Font = new Font(labelBug.Font,
                FontStyle.Bold);
              MessageBox.Show(labelBug.Text); break; }
    }
    textBox3Sum.Text = rezultat.ToString();
}

private void button2Clear_Click(object sender, EventArgs e)
{
    textBox1Number1.Text = "";
    textBox2Number2.Text = "";
    textBox3Sum.Text = "";
    textBoxZnak.Text = "";
    label1Sight.Text = "";
    textBoxZnak.Focus();
}

private void textBoxZnak_TextChanged(object sender, EventArgs e)
{
    if (textBoxZnak.Text == "+") label1Sight.Text =
        "+"; else if (textBoxZnak.Text == "-")
        label1Sight.Text = "-"; else if (textBoxZnak.Text ==
        "*") label1Sight.Text = "*"; else if
        (textBoxZnak.Text == "/") label1Sight.Text = "/";
    else label1Sight.Text = "грешка";
    label1Sight.Font = new Font(label1Sight.Font,
        FontStyle.Bold); textBox1Number1.Focus();
}

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

Диалогови кутии

При разработката на Windows Forms приложения често пъти се налага да извеждаме диалогови кутии с някакви съобщения или с някакъв въпрос. Нека разгледаме стандартните средства за такива ситуации.

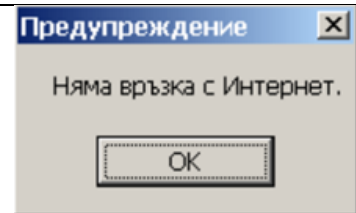
Стандартни диалогови кутии

Класът **MessageBox** ни позволява да извеждаме стандартни диалогови кутии, съдържащи текст, бутони и икони:

съобщения към потребителя
въпросителни диалози

Показването на диалогова кутия се извършва чрез извикване на статичния метод **Show(...)** на класа **MessageBox**. Следният код, например, ще покаже диалогова кутия със заглавие **"Предупреждение"** и текст **"Няма връзка с интернет"**:

**MessageBox.Show ("Няма връзка с Интернет.",
"Предупреждение");**

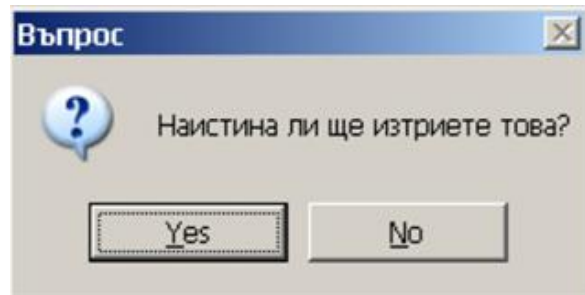


Нека разгледаме още един пример за стандартна диалогова кутия с малко повече функционалност:

bool confirmed =

**MessageBox.Show("Наистина ли ще изтриете това?", "Въпрос",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes;**

Този код ще покаже диалогова кутия със заглавие **"Въпрос"** и текст **"Наистина ли ще изтриете това?"**. Преди текста ще има икона с въпросителен знак в нея, а под него – бутони **Yes** и **No**. Ако потребителят натисне **Yes**, променливата **confirmed** ще има стойност **true**, в противен случай ще има стойност **false**. Ето как изглежда диалоговата кутия от примера:



Радио-бутон (RadioButton)

Радио-бутон е бутон, който има две състояния – активно и неактивно. Изобразява се с кръг, като в средата на активния бутон има точка. Обикновено се използват група от радио-бутони, като във всеки момент е активен само един. Най-важното свойство на радио-бутон е **Checked**. То е от булев тип и показва дали бутонът е натиснат или не. Друго негово свойство е **Text**.