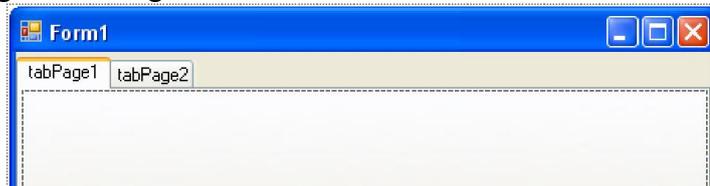


Панель вкладок TabControl в C# и платформе .NET. Создаем программу - тест.

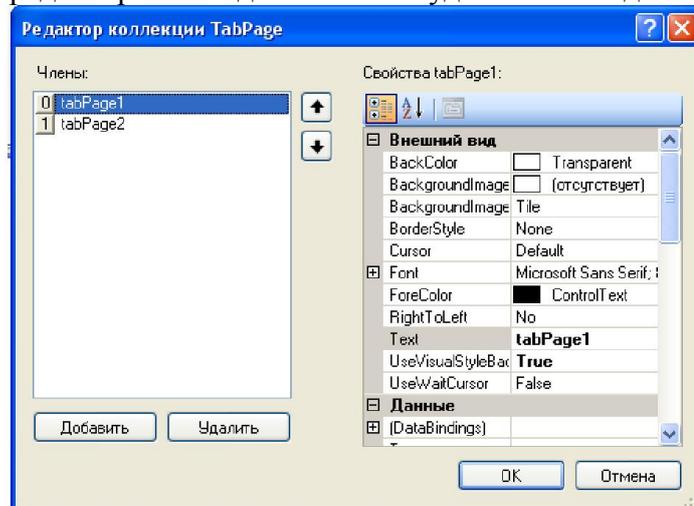
Чтобы настроить вкладки элемента TabControl используем свойство TabPages. При переносе элемента TabControl с панели инструментов на форму по умолчанию создаются две вкладки - tabPage1 и tabPage2.



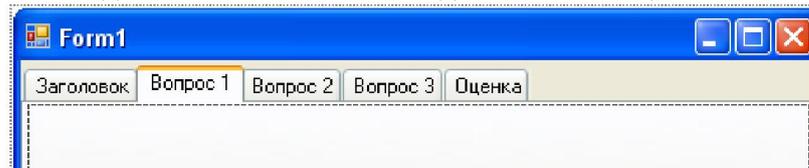
Изменим их отображение с помощью свойства TabPages:



Нам откроется окно редактирования/добавления и удаления вкладок:



Каждая вкладка представляет своего рода панель, на которую мы можем добавить другие элементы управления, а также заголовок, с помощью которого мы можем переключаться по вкладкам. Текст заголовка задается с помощью свойства Text.



На основе этих вкладок создадим простейшую программу тест.

Опишем глобальную переменную для хранения количества правильных ответов.

```
public partial class Form1 : Form
{
```

```
    int b = 0;
    public Form1()
```

Добавим команду, вывода формы в центре экрана.

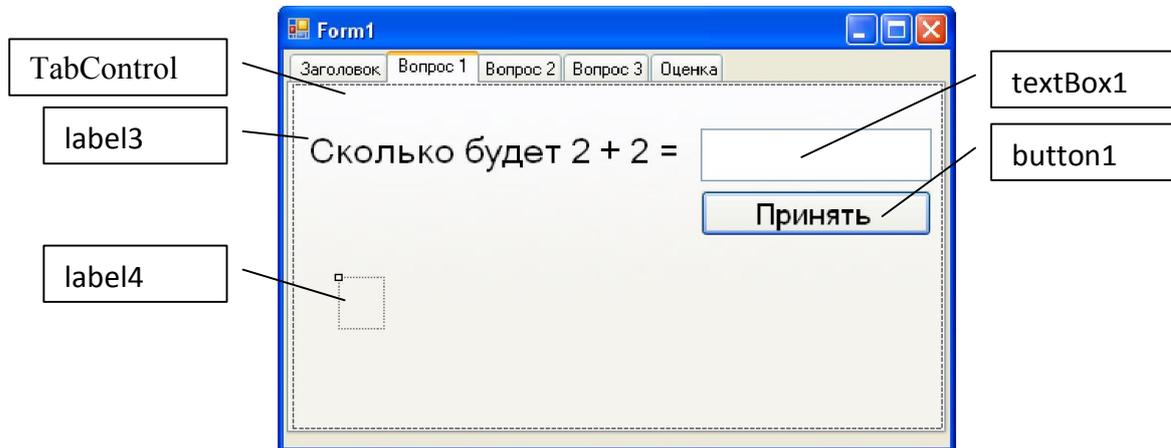
```
public Form1()
{
    InitializeComponent();
    this.StartPosition = FormStartPosition.CenterScreen;
}
}
```

В тесте мы будем использовать три типа вопросов (точнее три типа вариантов ответа на вопрос)

1. **Ответ вводится в виде числа, слова.** При создании серьезных проектов надо учитывать, что слово пользователь может вводить разными регистрами, поэтому варианты ответа можно включить через связку Or. Например, так:

```
if (textBox1.Text=="Кот" || textBox1.Text=="кот" ||
    textBox1.Text=="КОТ") ...
```

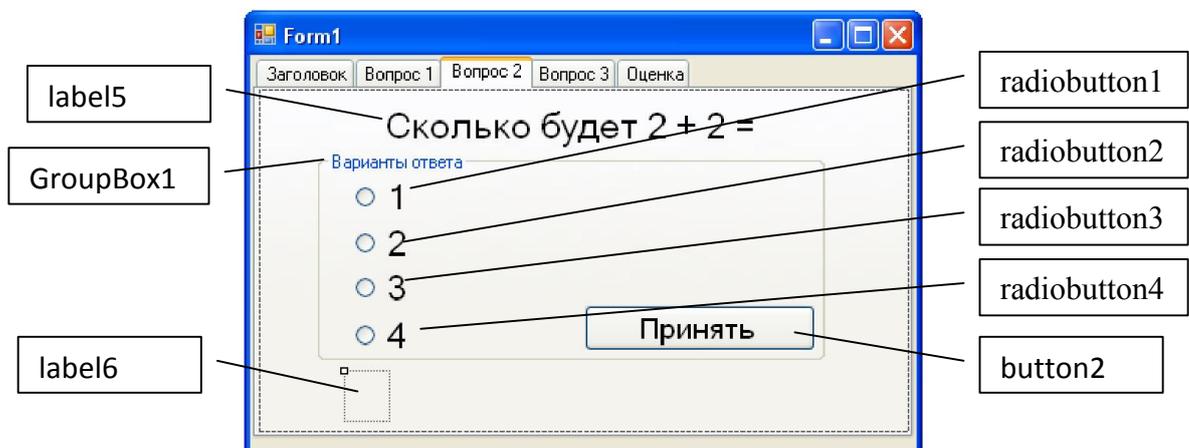
Размести на вкладке следующие элементы. Номера элементов могут быть другие. Все зависит от количества ранее установленных элементов.



Процедура «Принять» будет иметь следующий вид:

```
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text=="4")
    {
        label4.Text="Правильно";
        b = b + 1;
    }
    else label4.Text="Неправильно";
    // прячем кнопку что бы исключить повторный ввод
    // (угадывание ответа)
    button1.Visible=false;
}
}
```

2. Выбирается один вопрос при помощи радиокнопок.



Процедура «Принять» будет иметь следующий вид:

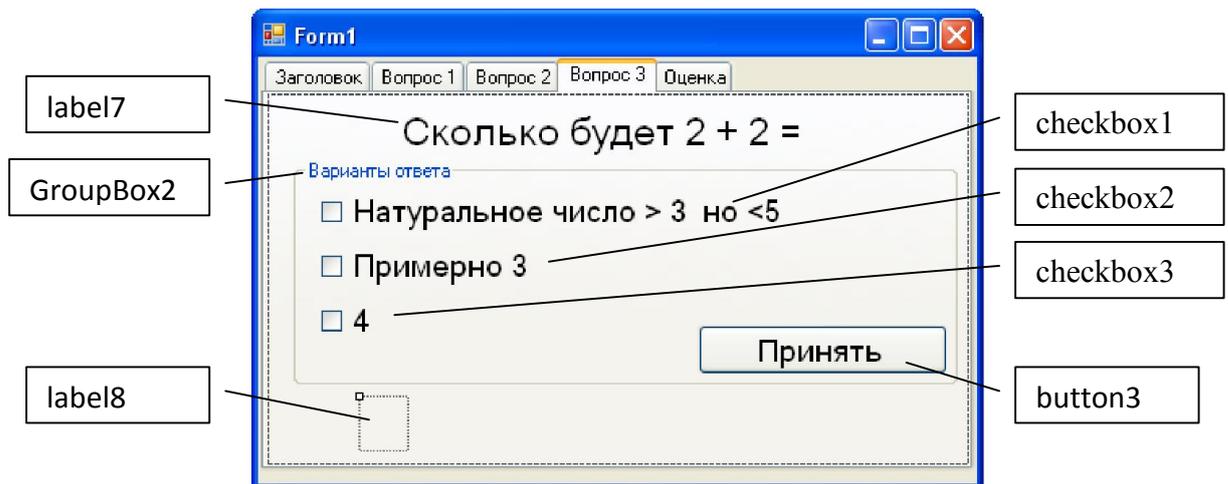
```
private void button2_Click(object sender, EventArgs e)
{
```

```

if (radioButton4.Checked==true)
{
    label6.Text="Правильно";
    b=b+1;
}
else label6.Text="Неправильно";
button2.Visible=false;
}

```

3. Выбираем несколько ответов.



Процедура «Принять» будет иметь следующий вид:

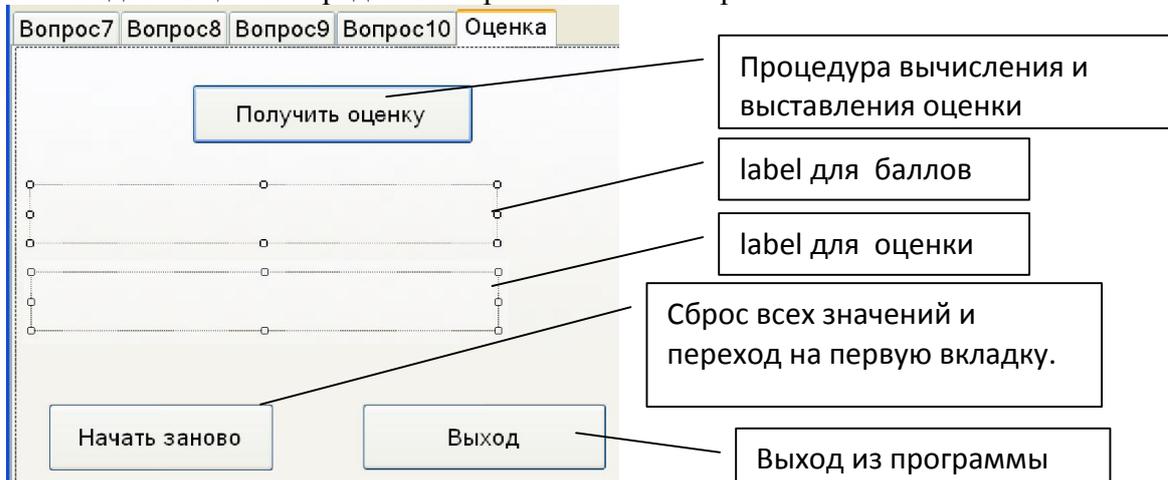
```

private void button3_Click(object sender, EventArgs e)
{
    if (checkBox1.Checked==true &&
        checkBox3.Checked==true &&
        checkBox2.Checked==false)
    {
        label18.Text = "Правильно";
        b = b + 1;
    }
    else label18.Text = "Неправильно";
    button3.Visible = false;
}

```

Так как галочки можно поставит на всех checkbox, то проверяем их все, не поставлено ли лишних галочек.

На вкладке «Оценка» предлагаем три возможных варианта:



Процедура «Получить» оценку может выглядеть так:

```
label9.Text="Набрано баллов= " + Convert.ToString(b) ;;
if (b==10) label10.Text="Оценка 5 (отлично)";
if (b==9 || b==8 || b==7) label10.Text="Оценка 4 (хорошо)";
if (b==6 || b==5) label10.Text="Оценка 3 (удовлетворительно)";
if (b==4 || b==3) label10.Text="Оценка 2 (плохо)";
if (b==2 || b==1) label10.Text="Оценка 1 (все ужасно)";
```

Переход в начало потребует очистки всех введенных пользователем данных:

```
// переходим на первую вкладку
tabcontrol1.SelectedIndex=0;
// показываем все кнопки «Проверить» обратно.
button1.Visible=true;
button2.Visible=true;
.
.
.
button10.Visible=true;
// скрываем надписи правильно неправильно
label20.text= "";
.
.
.
label2.text= "";
// убираем все галочки кнопочки и надписи в полях ответа
checkbox1.checked=false;
.
.
.
checkbox12.checked=false;
textBox1.Text= "";
.
.
.
textBox3.Text= "";
radioButton1.checked=false;
.
.
.
radioButton14.checked=false;
// обнуляем баллы и изменяем соответствующие надписи
b=0;
label21.Text="Набрано баллов= ";
label22.Text="";
```

В вопросе можно использовать картинки. Пример оформления вопроса.