Министерство образования и науки Российской

**Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 2**

**по дисциплине**

**«Программирование компьютерной графики»**

**По теме: «Разработка оконного приложения»**

Выполнила:

ст. гр. ПРИ-118

Левченко В.П

Принял:

Жигалов И.Е

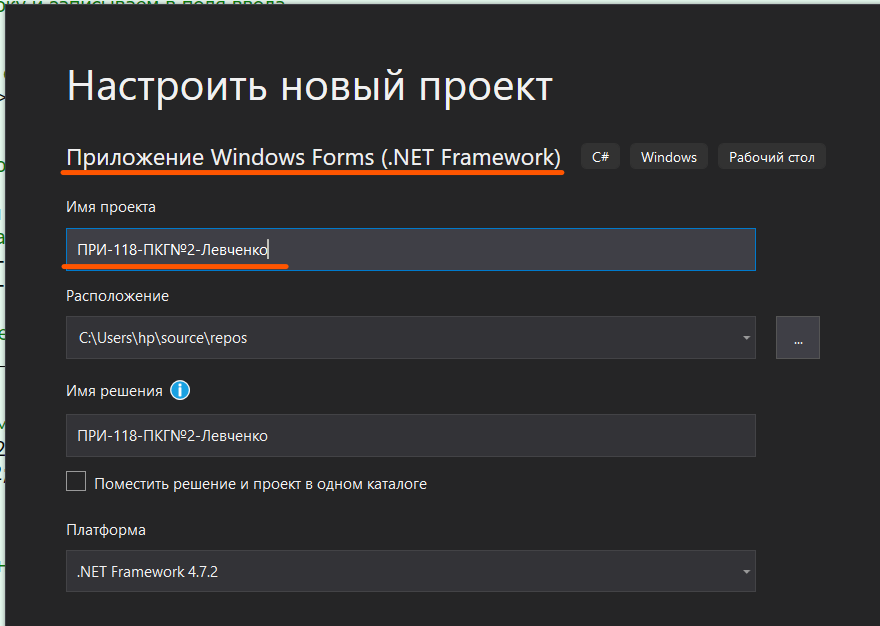
Владимир, 2021

**Цель работы**

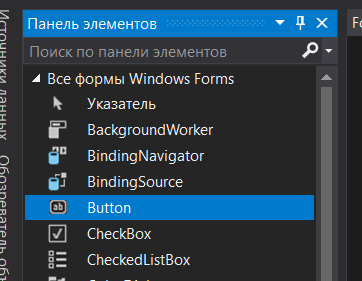
Освоение работы с оконными элементами в среде Microsoft Visual C# для ОС Windows с использованием Microsoft .net Framework. Получение практических навыков построения оконных приложений в Visual C#.

**Выполнение работ**

Для начала создадим новый проект Windows Forms.

****Рисунок 1. Создание нового проекта

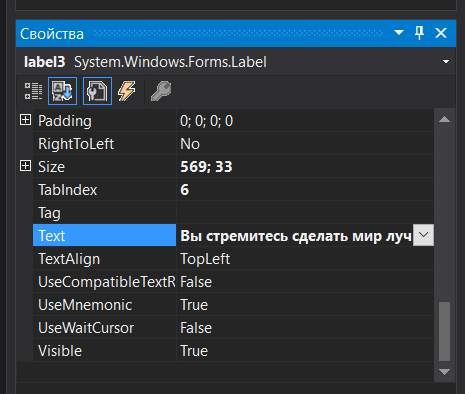
При открытии проекта, с левой стороны есть панель элементов.

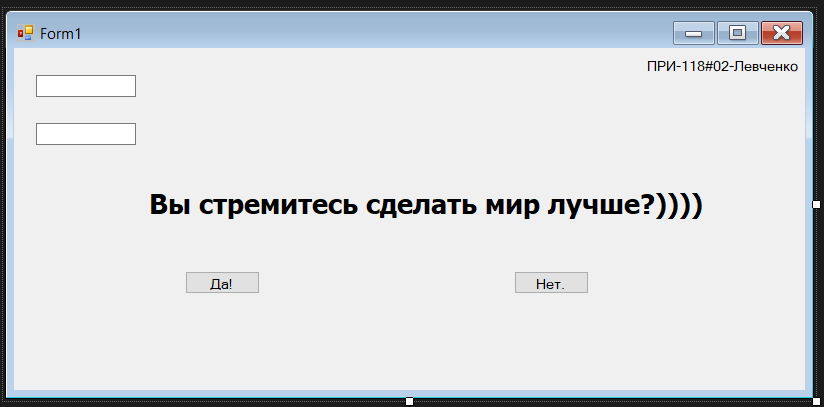
****Рисунок 2. Панель элементов

Добавим на форму элементы, которые понадобятся для дальнейшей работы.

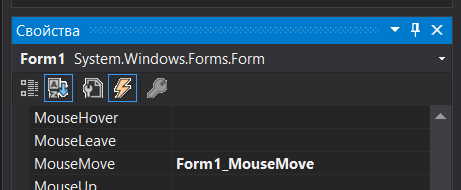
****Рисунок 3. Элементы для приложения

Теперь придадим рабочий вид форме и элементам, добавим подписи, обозначим кнопки.

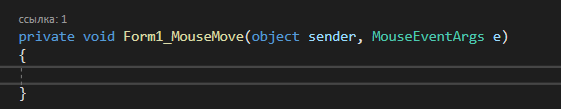
****Рисунок 4. Изменение label

****Рисунок 5. Результат изменения всех элементов

Определим перемещение указателя мыши по форме с помощью события MouseMove, для этого в свойствах формы перейдём на вкладку «События».

****Рисунок 6. Добавление события

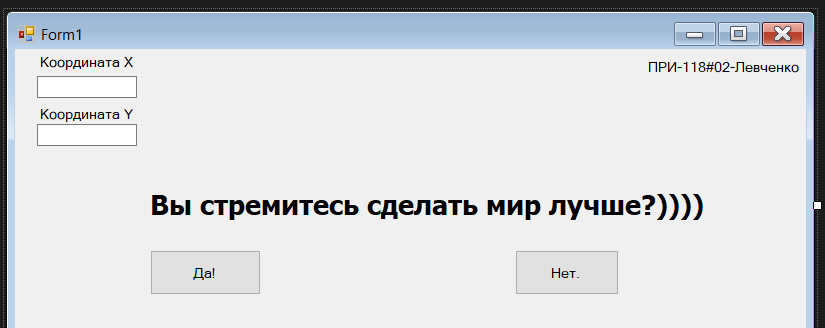
Теперь в коде появится новый обработчик события:

****Рисунок 7. Обработчик события перемещения курсора

Добавим в обработчик код, который будет определять координаты перемещения курсора x и y и будет записывать их в лейблы в форме:

textBox1.Text = e.X.ToString();

textBox2.Text = e.Y.ToString();

****Рисунок 8. Итоговый вид формы

Теперь добавим события для кнопок «Да!» и «Нет.»:

Для кнопки «Да!»:

private void button1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Ой, да не ври)");

Application.Exit();

}

Для кнопки «Нет.»:

private void button2\_Click(object sender, EventArgs e)

{

MessageBox.Show("Оно и видно", "Ну вы поглядите...", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

Добавление новых элементов управления на форму:

Объявим экземпляр класса Random (rnd), с помощью которого мы будем генерировать случайные числа.

Random rnd = new Random();

Point tmp\_location;

Объявим еще несколько переменных, часть из которых сразу будет инициализирована.

tmp\_location объявляется для того, что бы в будущем временно хранить текущее положение окна.

int \_w = System.Windows.Forms.SystemInformation.PrimaryMonitorSize.Width;

int \_h = System.Windows.Forms.SystemInformation.PrimaryMonitorSize.Height;

Код функции:

Для начала определим координаты, при приближении к которым надо будет сменить расположение окна:

if (e.X > 75 && e.X < 190 && e.Y > 150 && e.Y < 210)

{

// запоминаем текущее положение окна

tmp\_location = this.Location;

// генерируем перемещения по осям X и Y и прибавляем их к хранимому значению текущего положения окна

// числа генерируются в диапазоне от -100 до 100.

tmp\_location.X += rnd.Next(-100, 100);

tmp\_location.Y += rnd.Next(-100, 100);

// если окно вышло за пределы экрана по одной из осей

if (tmp\_location.X < 0 || tmp\_location.X > (\_w - this.Width / 2) || tmp\_location.Y < 0 || tmp\_location.Y > (\_h - this.Height / 2))

{

// новыми координатами станет центр окна

tmp\_location.X = \_w / 2;

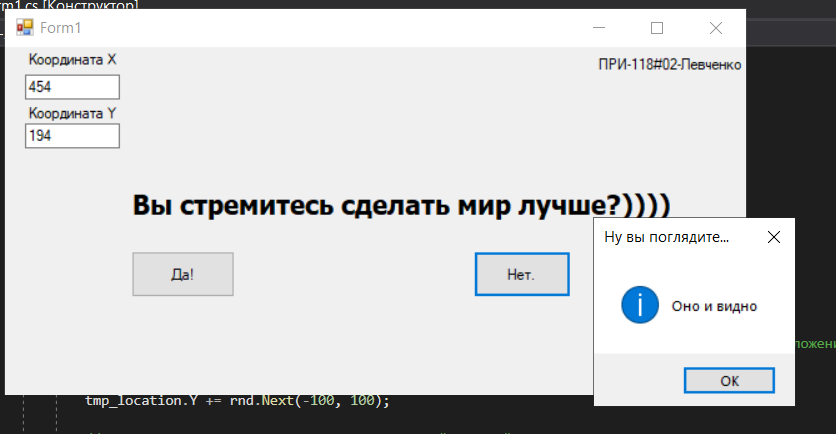
tmp\_location.Y = \_h / 2;

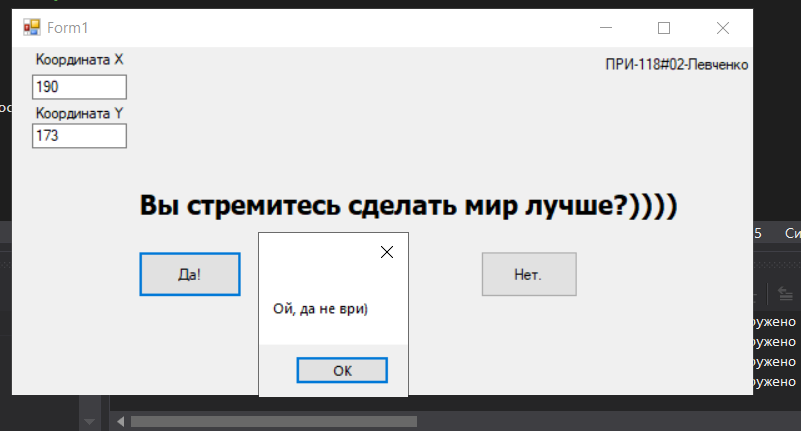
}

// обновляем положение окна на новое сгенерированное

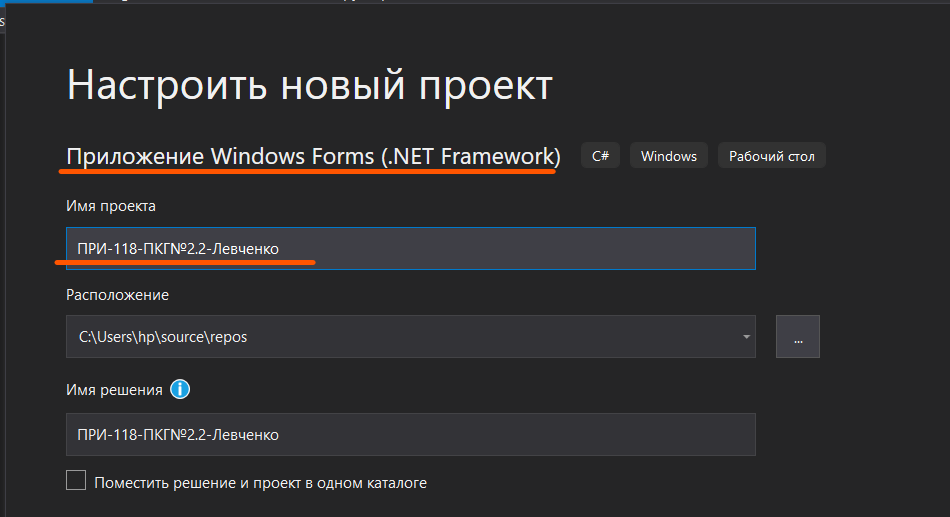
this.Location = tmp\_location;

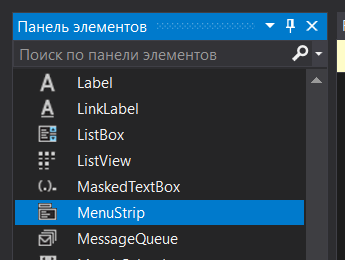
}

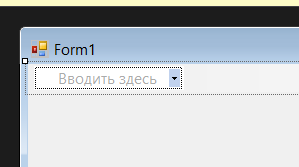
****Рисунок 9. Результат нажатия на кнопку «нет.»

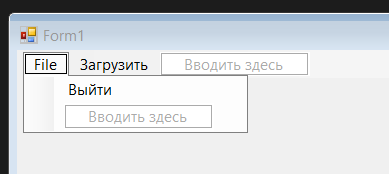
****Рисунок 10. Результат нажатия на кнопку «Да!»

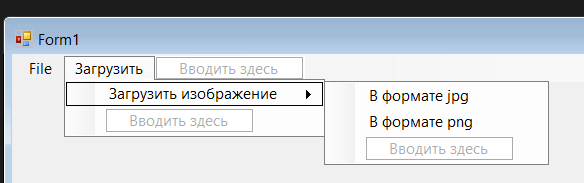
**Создание второго оконного приложения**

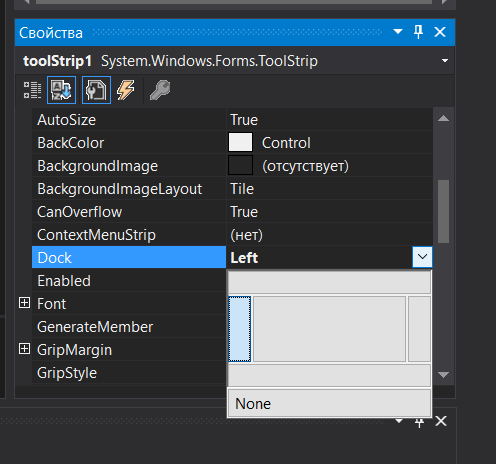
  
Рисунок 11. Создание нового проекта

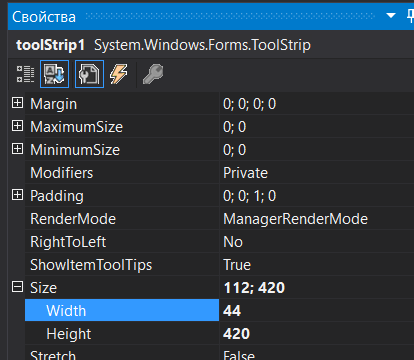
  
Рисунок 12. Добавляем на форму элемент

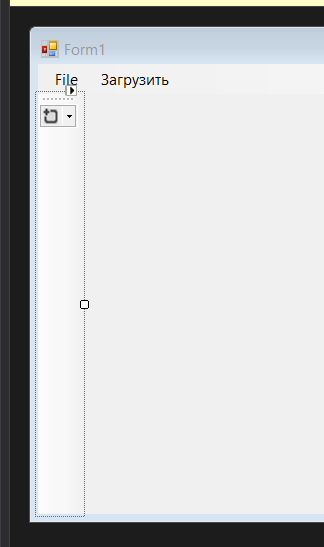
  
Рисунок 13. Результат добавления

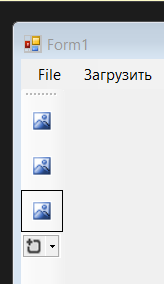
  
Рисунок 14. Добавление ещё нескольких элементов

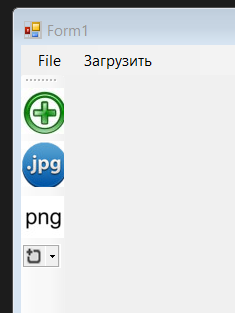
  
Рисунок 15. Окончательный вариант верхнего меню

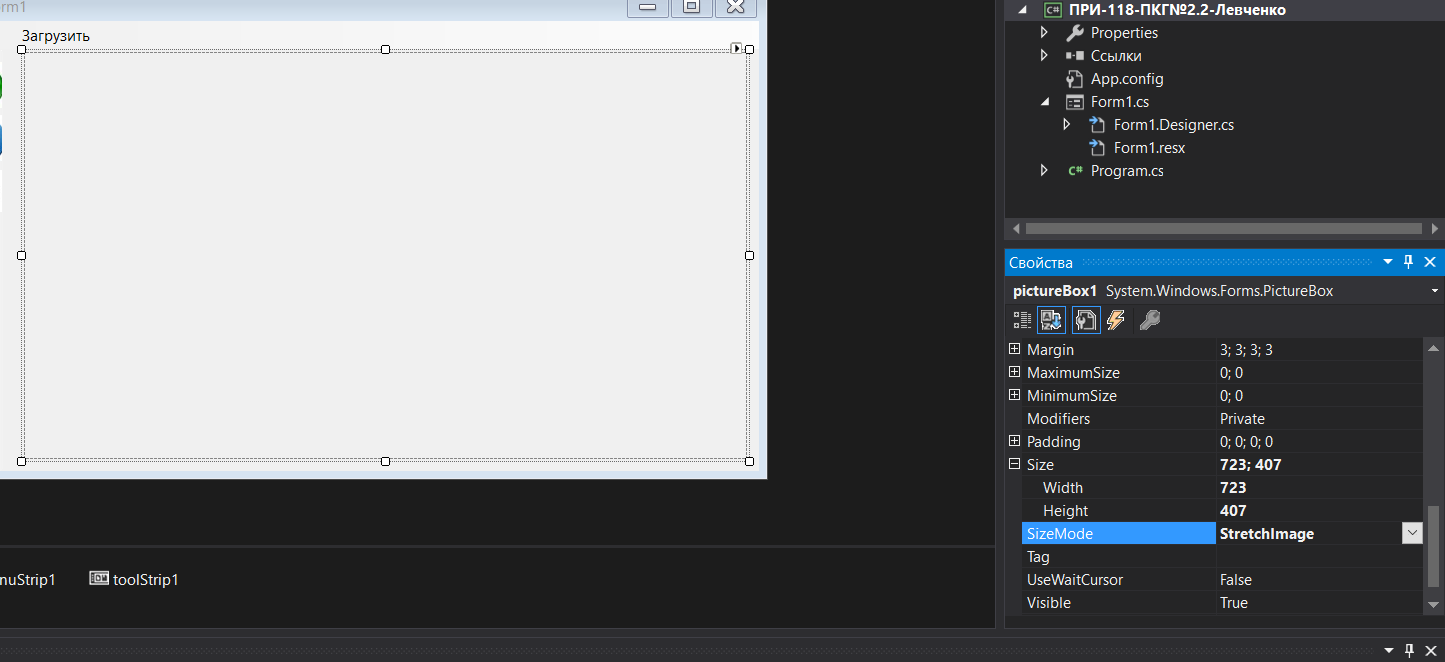
  
Рисунок 16. Изменение привязки элемента ToolStrip

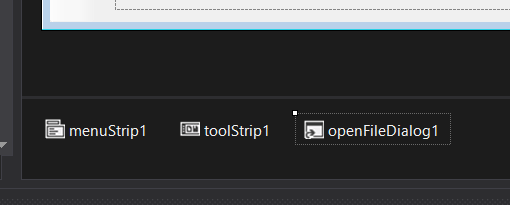
  
Рисунок 17. Изменение размера (50; 50)

  
Рисунок 18. Результат изменения

  
Рисунок 19. Добавление кнопок

  
Рисунок 20. Добавление картинок на кнопки

  
Рисунок 21. Изменение размера

  
Рисунок 22. Элементы

Обработчики событий:

//jpeg

private void toolStripButton2\_Click(object sender, EventArgs e)

{

}

//png

private void toolStripButton3\_Click(object sender, EventArgs e)

{

}

//d формате jpg

private void вФорматеJpgToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

//в формате png

private void вФорматеPngToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

Функция добавления изображения:

Image **MemForImage**;  
  
// функция загрузки изображения  
private void LoadImage(bool jpg)  
{

// директория, которая будет выбрана как начальная в окне для выбора файла  
openFileDialog1.InitialDirectory = "c:\\";  
  
// если будем выбирать jpg файлы  
if (jpg)  
{

// устанавливаем формат файлов для загрузки - jpg  
openFileDialog1.Filter = "image (JPEG) files (\*.jpg)|\*.jpg|All files (\*.\*)|\*.\*";

}  
else  
{

// устанавливаем формат файлов для загрузки - png  
openFileDialog1.Filter = "image (PNG) files (\*.png)|\*.png|All files (\*.\*)|\*.\*";

}  
  
// если открытие окна выбора файла завершилось выбором файла и нажатием кнопки ОК  
if(openFileDialog1.ShowDialog() == DialogResult.OK)  
{

try // безопасная попытка  
{

// пытаемся загрузить файл с именем openFileDialog1.FileName - выбранный пользователем файл.  
MemForImage = Image.FromFile(openFileDialog1.FileName);  
// устанавливаем картинку в поле элемента PictureBox  
pictureBox1.Image = MemForImage;

}  
catch (Exception ex) // если попытка загрузки не удалась  
{

// выводим сообщение с причиной ощибки  
MessageBox.Show( "Не удалось загрузить файл: " + ex.Message);

}

}

Добавляем код в обработчики событий:

//jpeg

private void toolStripButton2\_Click(object sender, EventArgs e)

{

// загружаем jpg файлы

LoadImage(true);

}

//png

private void toolStripButton3\_Click(object sender, EventArgs e)

{

// загружаем png файлы

LoadImage(false);

}

//d формате jpg

private void вФорматеJpgToolStripMenuItem\_Click(object sender, EventArgs e)

{

// загружаем jpg файлы

LoadImage(true);

}

//в формате png

private void вФорматеPngToolStripMenuItem\_Click(object sender, EventArgs e)

{

// загружаем png файлы

LoadImage(false);

}

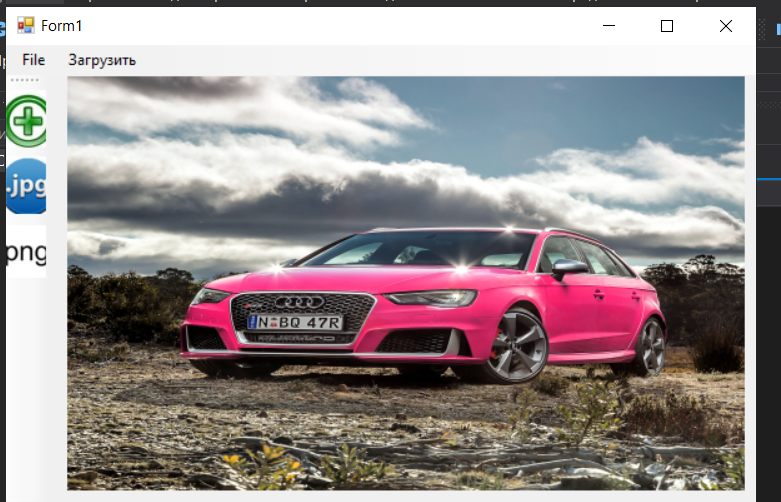
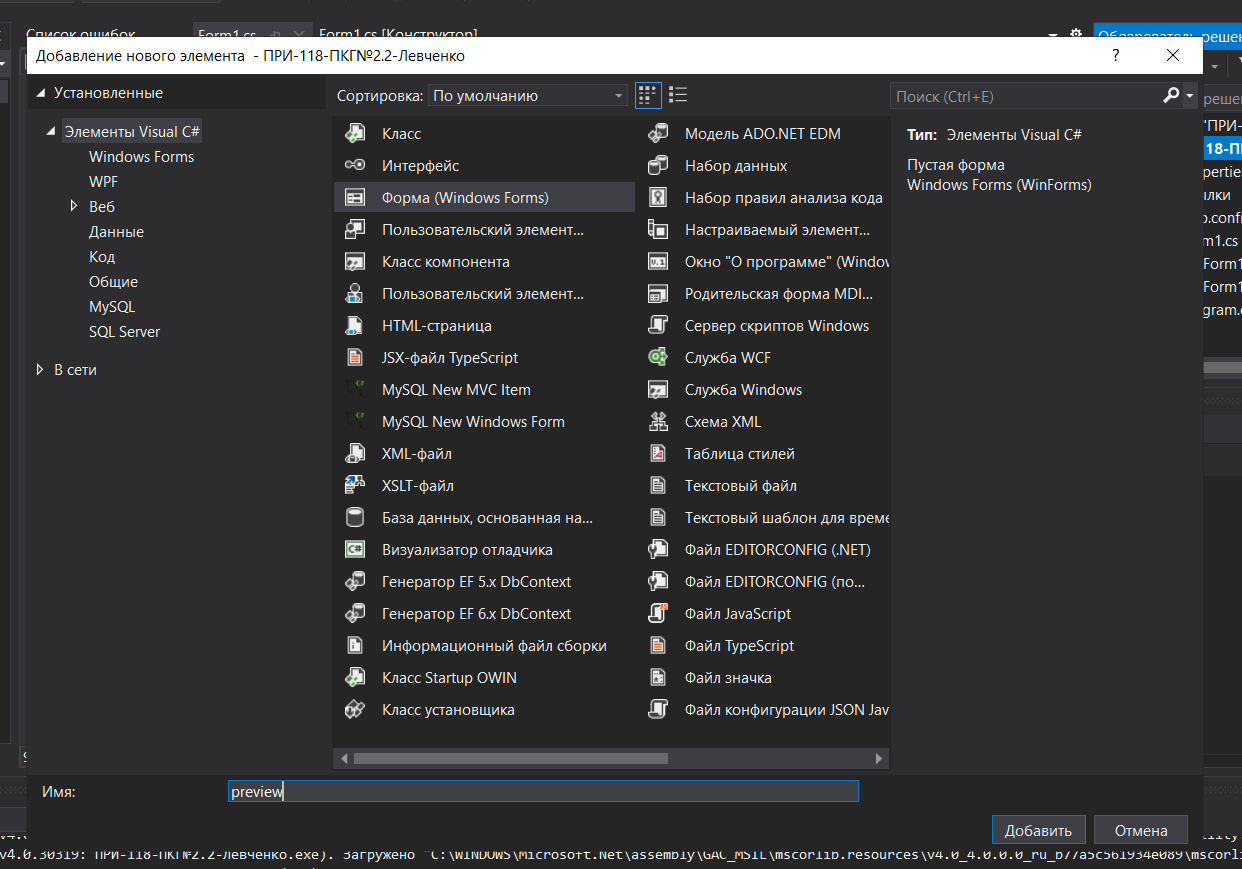
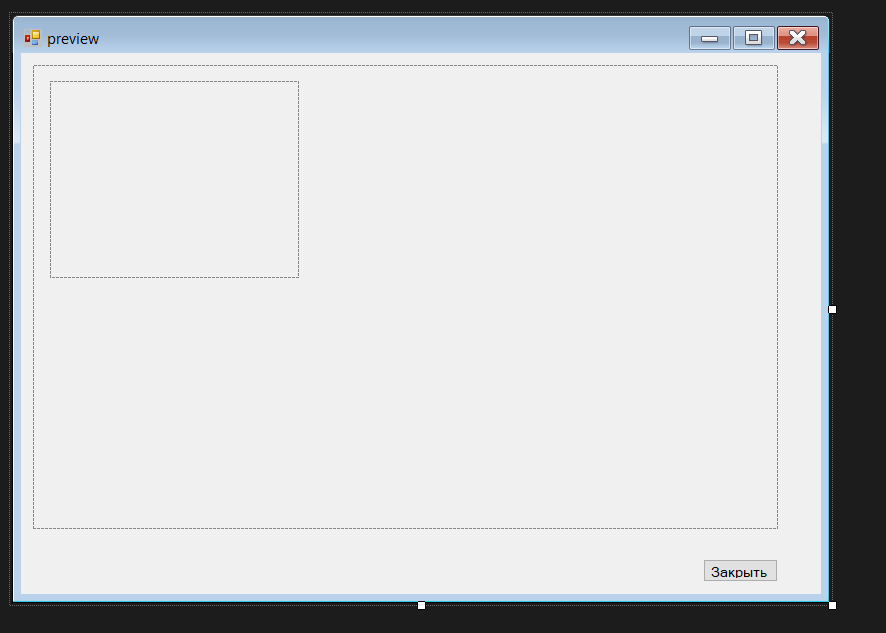
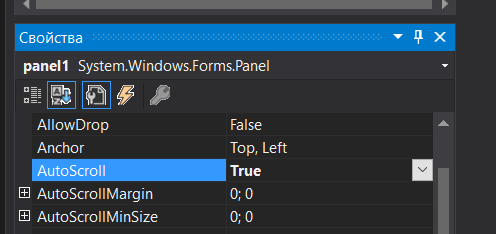
  
Рисунок 23. Результат добавления изображения jpg

  
Рисунок 24. Результат добавления изображения png

  
Рисунок 25. Добавляем новую форму

  
Рисунок 26. Добавляем элементы

  
Рисунок 27. Изменяем свойство автоскролла

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ПРИ\_118\_ПКГ\_2.\_2\_Левченко

{

public partial class preview : Form

{

Image ToView;

public preview(Image view)

{

ToView = view;

InitializeComponent();

}

private void preview\_Load(object sender, EventArgs e)

{

// если объект, хранящий изображение неравен null

if (ToView != null)

{

// устанавливаем новые размеры элемента pictureBox1,

// равные ширине (ToView.Width) и высоте (ToView.Height) загружаемого изображения.

pictureBox1.Size = new Size(ToView.Width, ToView.Height);

// устанавливаем изображение для отображения в элементе pictureBox1

pictureBox1.Image = ToView;

}

}

private void button1\_Click(object sender, EventArgs e)

{

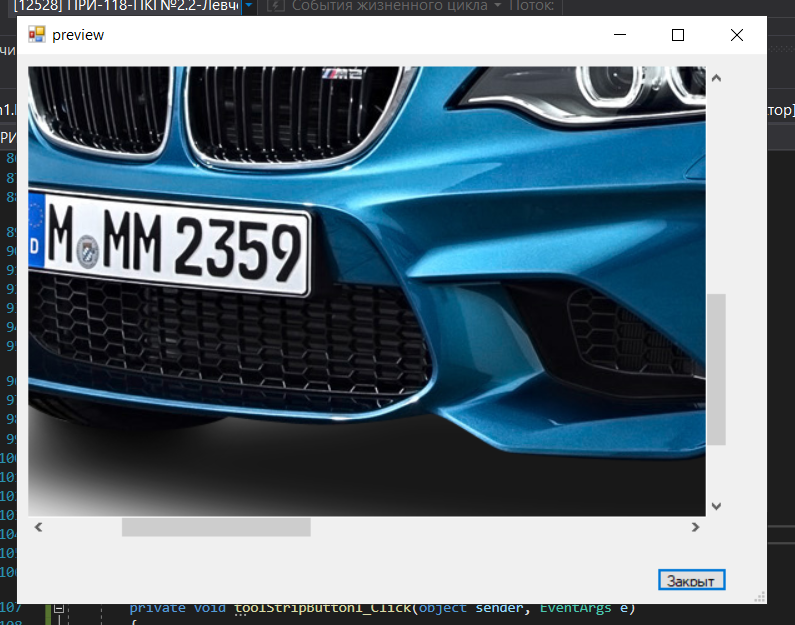
// закрываем диалоговое окно

Close();

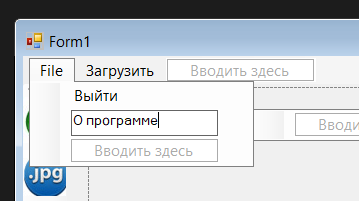
}

}

}

  
Рисунок 28. Результат открытия дополнительного окна

Аналогично добавляем ещё одно дополнительное окно, добавляем в меню новый пункт о программе, затем добавляем обработчик события.

  
Рисунок 29. Новый пункт

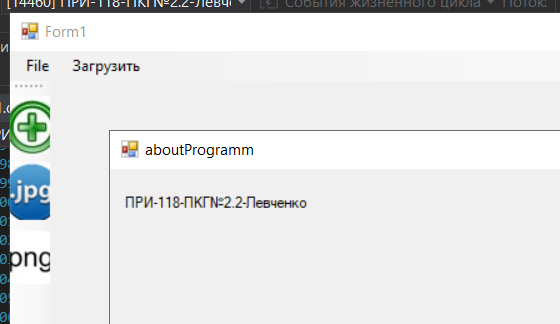
private void оПрограммеToolStripMenuItem\_Click(object sender, EventArgs e)

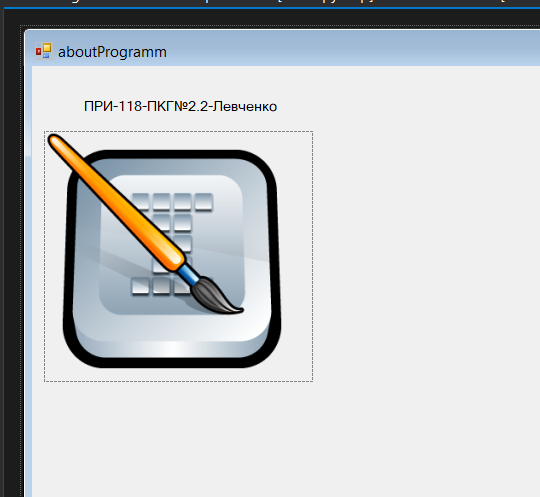
{

Form about = new aboutProgramm();

about.ShowDialog();

}

  
Рисунок 30. Результат добавления информации о программе

  
Рисунок 31. Добавление логотипа с помощью элемента PictureBox

**Вывод**

Освоена работа с оконными элементами в среде Microsoft Visual C# для ОС Windows с использованием Microsoft .net Framework. Получены практические навыки построения оконных приложений в Visual C#.