

Лабораторная работа № 2

Тема: Передача данных между формами в среде Visual Studio на языке C#

Цель работы: Разработка программы с обменом данных между формами в среде Visual Studio на языке C#

Оборудование: IBM – совместимые компьютеры.

Место проведения: Компьютерный класс.

Техника безопасности: См. инструкцию.

Теоретическое описание

Основные свойства форм

С помощью специального окна Properties (Свойства) справа Visual Studio предоставляет нам удобный интерфейс для управления свойствами элемента

Большинство этих свойств оказывает влияние на визуальное отображение формы. Пробежимся по основным свойствам:

- **Name:** устанавливает имя формы - точнее имя класса, который наследуется от класса Form
- **BackColor:** указывает на фоновый цвет формы. Щелкнув на это свойство, мы сможем выбрать тот цвет, который нам подходит из списка предложенных цветов или цветовой палитры
- **BackgroundImage:** указывает на фоновое изображение формы
- **BackgroundImageLayout:** определяет, как изображение, заданное в свойстве BackgroundImage, будет располагаться на форме.
- **ControlBox:** указывает, отображается ли меню формы. В данном случае под меню понимается меню самого верхнего уровня, где находятся иконка приложения, заголовок формы, а также кнопки минимизации формы и крестик. Если данное свойство имеет значение false, то мы не увидим ни иконку, ни крестика, с помощью которого обычно закрывается форма
- **Cursor:** определяет тип курсора, который используется на форме
- **Enabled:** если данное свойство имеет значение false, то она не сможет получать ввод от пользователя, то есть мы не сможем нажать на кнопки, ввести текст в текстовые поля и т.д.
- **Font:** задает шрифт для всей формы и всех помещенных на нее элементов управления. Однако, задав у элементов формы свой шрифт, мы можем тем самым переопределить его
- **ForeColor:** цвет шрифта на форме
- **FormBorderStyle:** указывает, как будет отображаться граница формы и строка заголовка. Устанавливая данное свойство в None можно создавать внешний вид приложения произвольной формы
- **HelpButton:** указывает, отображается ли кнопка справки формы
- **Icon:** задает иконку формы
- **Location:** определяет положение по отношению к верхнему левому углу экрана, если для свойства StartPosition установлено значение Manual
- **MaximizeBox:** указывает, будет ли доступна кнопка максимизации окна в заголовке формы
- **MinimizeBox:** указывает, будет ли доступна кнопка минимизации окна
- **MaximumSize:** задает максимальный размер формы
- **MinimumSize:** задает минимальный размер формы
- **Opacity:** задает прозрачность формы
- **Size:** определяет начальный размер формы

- **StartPosition:** указывает на начальную позицию, с которой форма появляется на экране
- **Text:** определяет заголовок формы
- **TopMost:** если данное свойство имеет значение true, то форма всегда будет находиться поверх других окон
- **Visible:** видима ли форма, если мы хотим скрыть форму от пользователя, то можем задать данному свойству значение false
- **WindowState:** указывает, в каком состоянии форма будет находиться при запуске: в нормальном, максимизированном или минимизированном

Программная настройка свойств

С помощью значений свойств в окне Свойства мы можем изменить по своему усмотрению внешний вид формы, но все то же самое мы можем сделать динамически в коде. Перейдем к коду, для этого нажмем правой кнопкой мыши на форме и выберем в появившемся контекстном меню View Code (Просмотр кода). Перед нами открывается файл кода *Form1.cs*. Изменим его следующим образом:

```
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 HelloApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            Text = "Hello World!";
            this.BackColor = Color.Aquamarine;
            this.Width = 250;
            this.Height = 250;
        }
    }
}
```

Начальное расположение формы

Начальное расположение формы устанавливается с помощью свойства *StartPosition*, которое может принимать одно из следующих значений:

- **Manual:** Положение формы определяется свойством *Location*
- **CenterScreen:** Положение формы в центре экрана
- **WindowsDefaultLocation:** Позиция формы на экране задается системой Windows, а размер определяется свойством *Size*
- **WindowsDefaultBounds:** Начальная позиция и размер формы на экране задается системой Windows
- **CenterParent:** Положение формы устанавливается в центре родительского окна

Ход работы

1. Запустить MS Visual Studio. Создать приложение по шаблону [Windows Forms Application](#).
2. Установить на форму компоненты button и textbox (Рисунок 1)

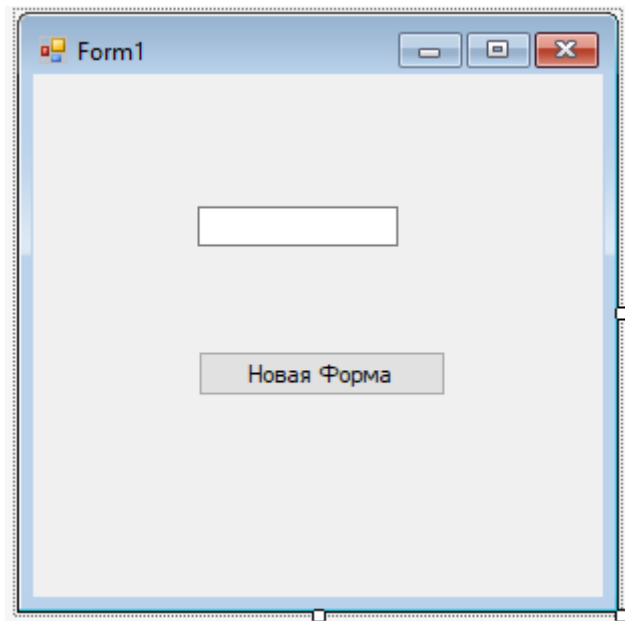


Рисунок 1

3. Добавить в приложение вторую форму (Проект – Добавить форму – Форма Windows Forms).
4. Для открытия второй формы выполним следующие действия. Перейти к первой форме и щелкнуть дважды на кнопке – откроется код с описанием события Click. Ввести в код следующие данные

```
namespace WindowsFormsAppForms
{
    public partial class Form1 : Form
    {
        Form2 frm2;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DataBank.Text = textBox1.Text;
            frm2 = new Form2();
            frm2.Show();
        }
    }
}
```

5. Проверить работу приложения. При нажатии на кнопку первой формы откроется вторая форма.
6. На второй форме установить компонент Label и задать в его свойстве Text следующую строку
`label1.Text = "Загружена Вторая форма";`
7. Запустить приложение и проверить его работу.

8. Для передачи данных между формами создадим новый класс. Для этого в Обозревателе решений выбрать в контекстном меню Добавить – Класс... (Рисунок2) Задать ему имя DataBank.cs . Откроется окно с кодом описания класса, в котором сделать класс статичным.

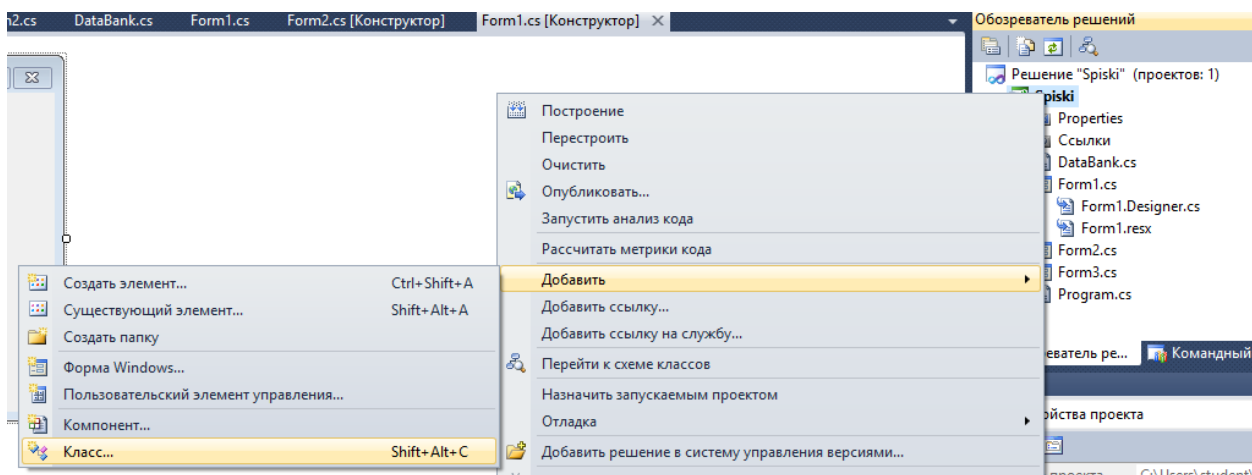


Рисунок 2

9. Введем для него следующий код, определив в нем переменную text

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Spiski
{
    static class DataBank
    {
        public static string Text;
    }
}
```

10. Для кнопки первой формы допишем текст

```
private void button1_Click(object sender, EventArgs e)
{
    DataBank.Text = textBox1.Text;
    frm2 = new Form2();
    frm2.Show();
}
```

11. На второй форме в событии Load написать код, который будет отображаться при загрузке формы

```
private void Form2_Load(object sender, EventArgs e)
{
    label1.Text = DataBank.Text;
}
```

Задания

1. Настроить начальное расположение форм по центру экрана.
2. Добавить третью форму и установить код для ее открытия и передачи данных.