Пример создания и вызова новой формы приложения в C#

Содержание

Условие задания

Разработать демонстрационное приложение, осуществляющее вызов из главной формы второстепенной формы по схеме, изображенной на рис. 1. Приложение реализует взаимодействие между различными формами, которыми могут быть диалоговые окна любой сложности.

В главной форме Form1 разместить:

- элемент управления типа Label для вывода результата возврата из второстепенной формы;
- элемент управления типа Button для вызова второстепенной формы.

Во второстепенной форме Form2 разместить:

- элемент управления типа Label для вывода заголовка формы;
- два элемента управления типа Button для обеспечения подтверждения или неподтверждения выбора (действия) во второстепенной форме.

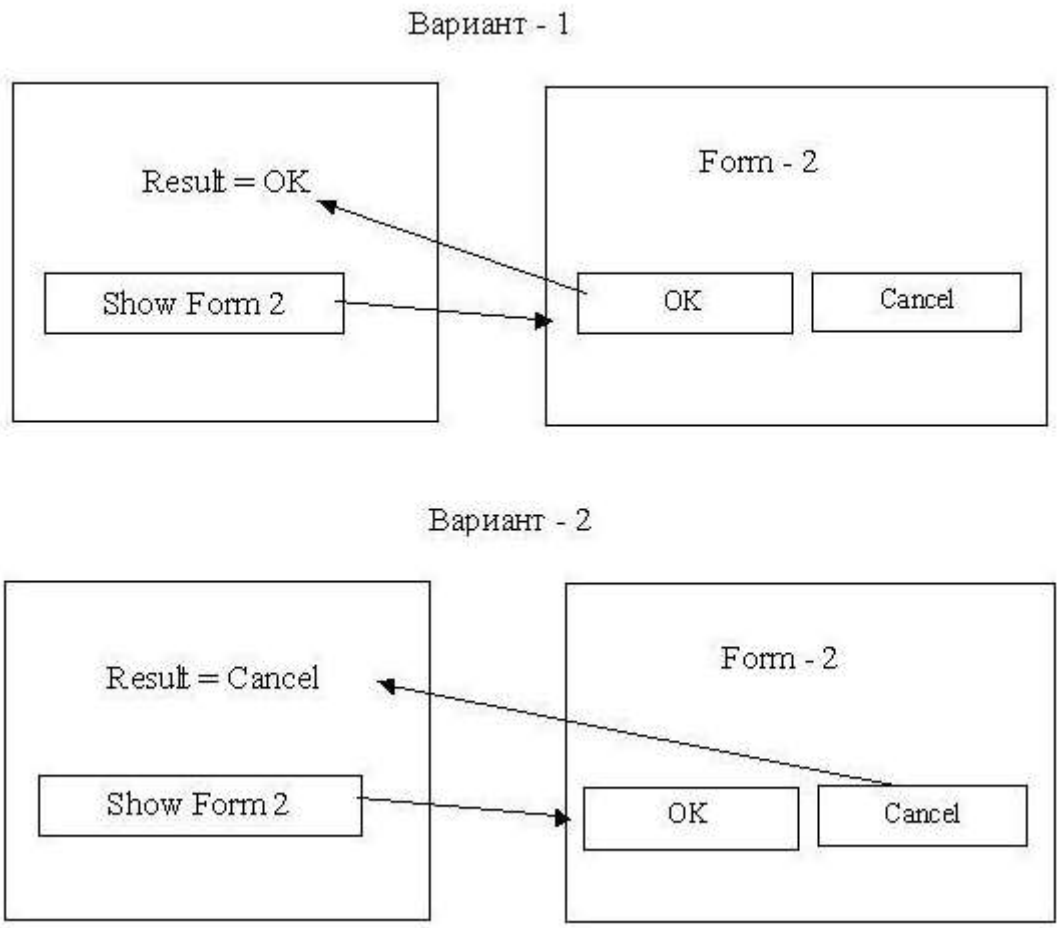

Рис. 1. Схема взаимодействия между формами

Выполнение

1. Создать приложение типа Windows Forms Application

Запустить Microsoft Visual Studio 2010. Пример создания нового приложения типа Windows Forms Application подробно описывается **здесь**.

Сохранить проект в произвольной папке.

После создания приложения у нас есть одна форма. Программный код формы размещен в файле «Form1.cs» (рис. 2).

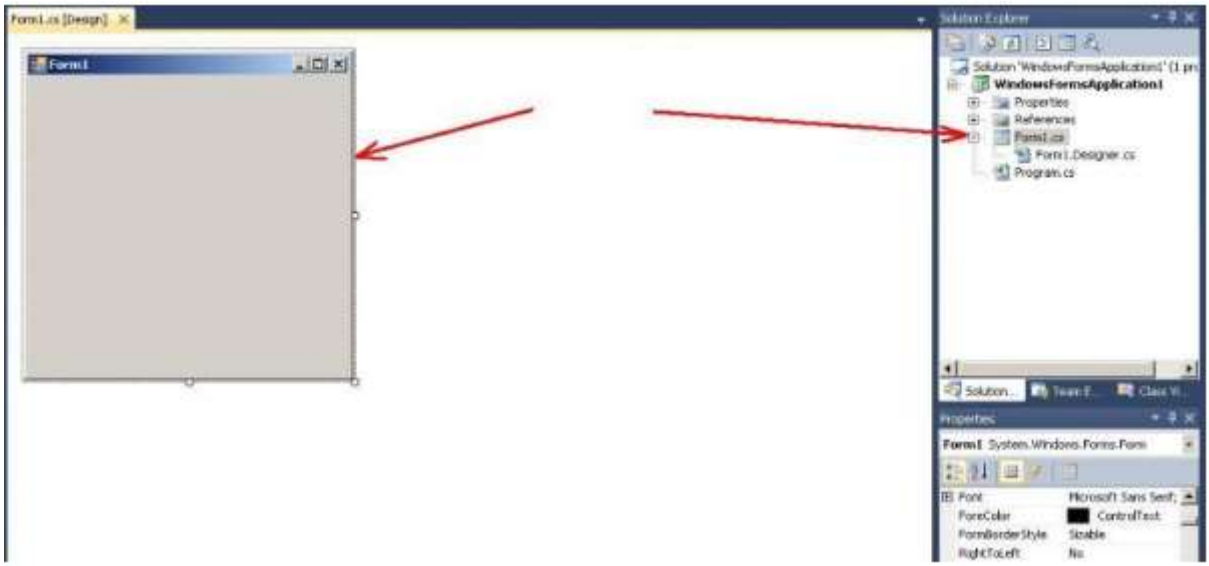

Рис. 2. Главная форма приложения Form1

2. Разработка главной формы приложения

Из палитры элементов управления Toolbox выносим на форму:

- элемент управления типа Button;
- элемент управления типа Label.

Автоматически создаются два объекта-переменные с именами button1 и label1.
В элементе управления типа Button свойство Text установить в значение «Show Form 2«.
В элементе управления типа Label свойство Text установить в значение «Result = «.
После внесенных изменений главная форма Form1 приложения будет иметь вид как показано на рисунке 3.

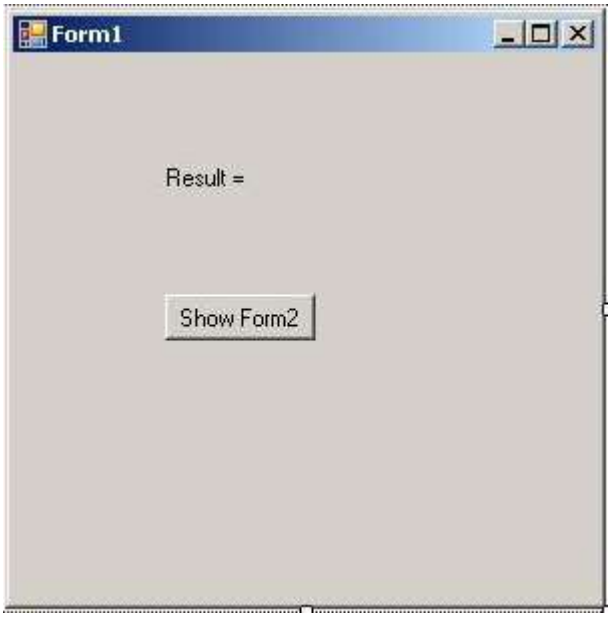

Рис. 3. Главная форма приложения после внесенных изменений

3. Создание второстепенной формы

Для создания второстепенной формы в C# можно воспользоваться несколькими способами.

Способ 1.
Для добавления формы №2 в проект этим способом нужно вызвать команду (рис. 4)

```
Project -> Add Windows Form...
```

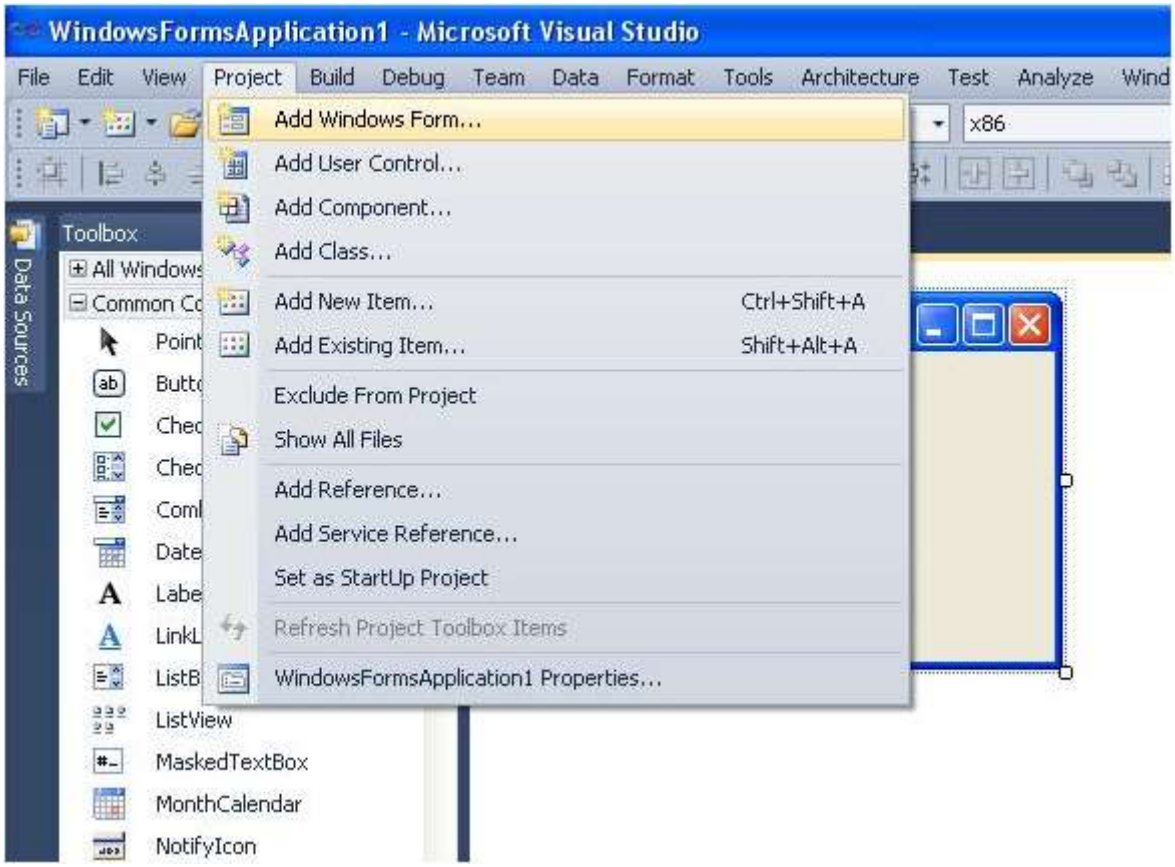

Рис. 4. Команда «Add Windows Form…» для добавления новой формы в проект

В результате откроется окно «Add New Item — Windows Forms Application1«. В этом окне выбираем элемент «Windows Form» (рис. 5). Оставляем имя формы как «Form2.cs«.

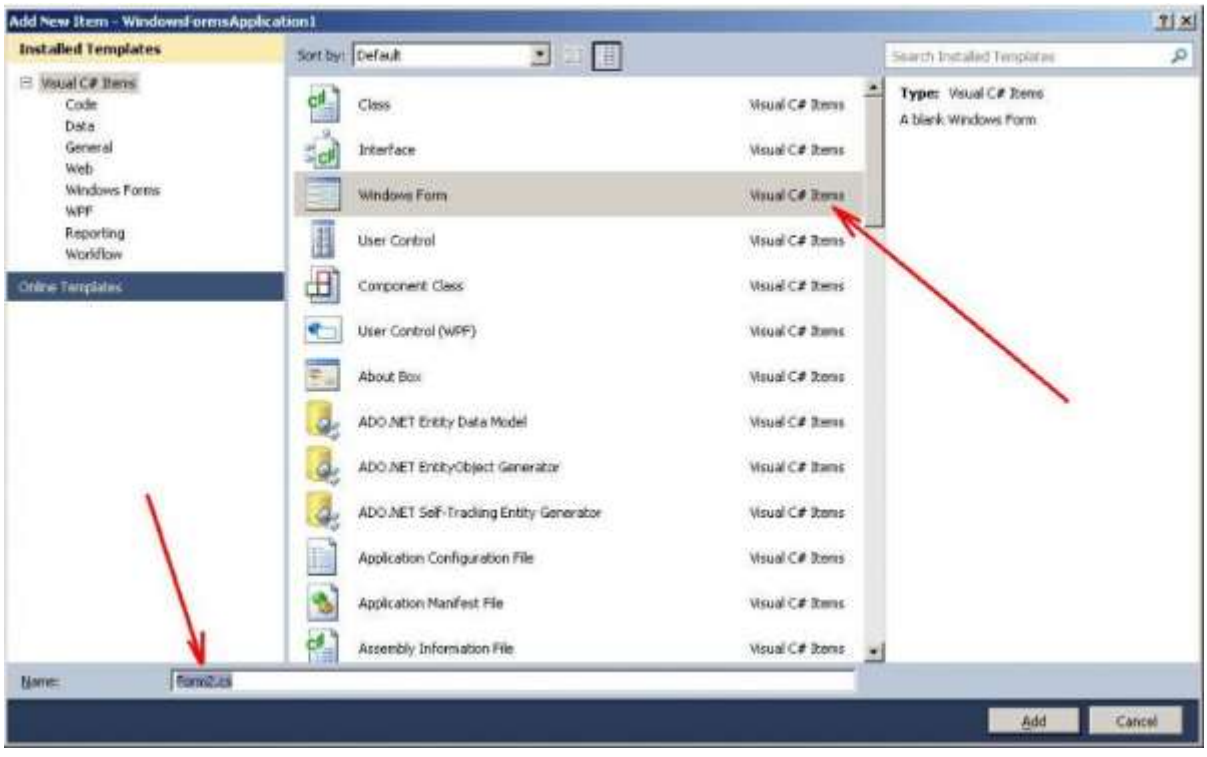

Рис. 5. Окно добавления новой формы к проекту

После нажатия на кнопке «Add» новая форма будет добавлена к проекту (рис. 6).

Рис. 6. Новосозданная форма Form2 и файл «Form2.cs«

Способ 2.

Также новую форму можно добавить к проекту с помощью соответствующей команды из контекстного меню (рис. 7). Последовательность действий следующая:

- в Solution Explorer сделать клик правой кнопкой «мышки» на названии приложения WindowsFormsApplication1;
- выбрать подменю Add;
- в подменю Add выбрать команду «Windows Form…«.

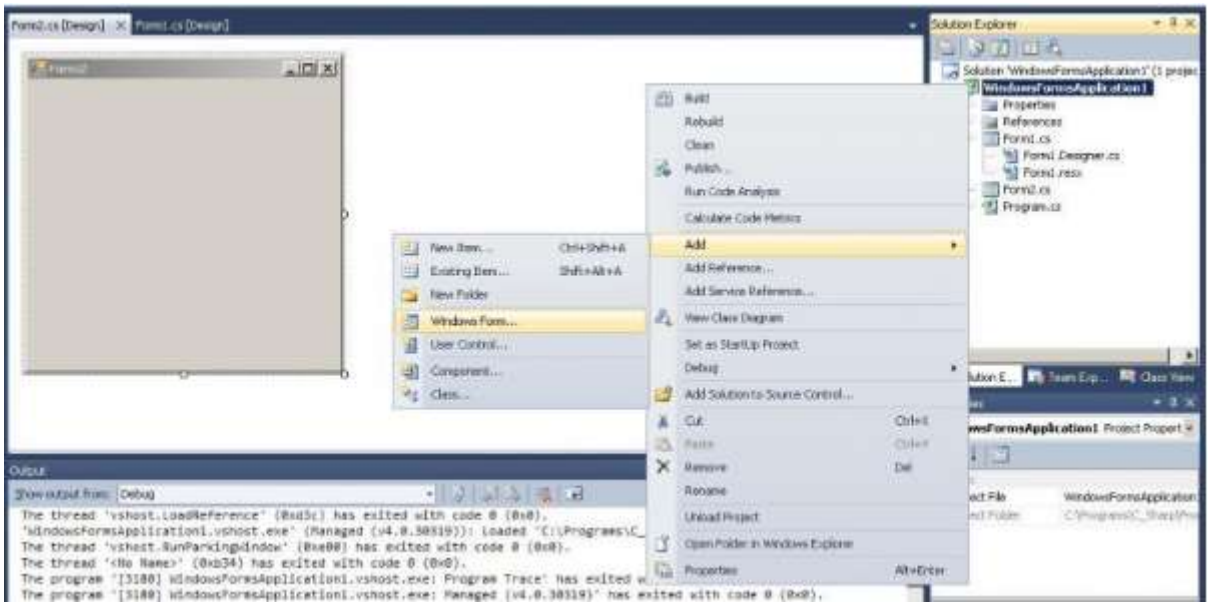

Рис. 7. Добавление новой формы из Solution Explorer

В результате откроется точно такое же окно как на рисунке 5.

4. Разработка второстепенной формы

Следующим шагом есть разработка второстепенной формы. Используя средства панели инструментов Toolbox создаем второстепенную форму Form2 как показано на рисунке 8. Такое построение формы соответствует условию задачи. Таким же образом, на Form2 имеем элементы управления label1, butto1, button2.

Рис. 8. Второстепенная форма Form2

5. Программирование событий клика на кнопках OK и Cancel формы Form2

Программируем событие клика на кнопке OK. Подробный пример программирования события клика на кнопке OK описывается **здесь**.

В программный код обработчика события button1_Click() (кнопка «OK«) вписываем следующую строку:

```
this.DialogResult = DialogResult.OK;
```

это значит, что результат возврата из формы Form2 есть «OK«.
Точно так же в обработчике события button2_Click вписываем:

```
this.DialogResult = DialogResult.Cancel;
```

это значит выбор кнопки «Cancel» (button2).

После внесенных изменений листинг программного кода файла «Form2.cs» будет иметь следующий вид:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
public partial class Form2 : Form
{
      public Form2()
      {
      InitializeComponent();
      }
      private void button1_Click(object sender, EventArgs e)
            {
      this.DialogResult = DialogResult.OK;
      }
      private void button2_Click(object sender, EventArgs e)
            {
      this.DialogResult = DialogResult.Cancel;
      }
      }
      }
```

Переход к программному коду формы Form2 (файл «Form2.cs«) можно осуществить с помощью Solution Explorer. Для этого в Solution Explorer вызываем контекстное меню для формы Form2 и из этого меню выбираем команду «View Code» (рис. 9).

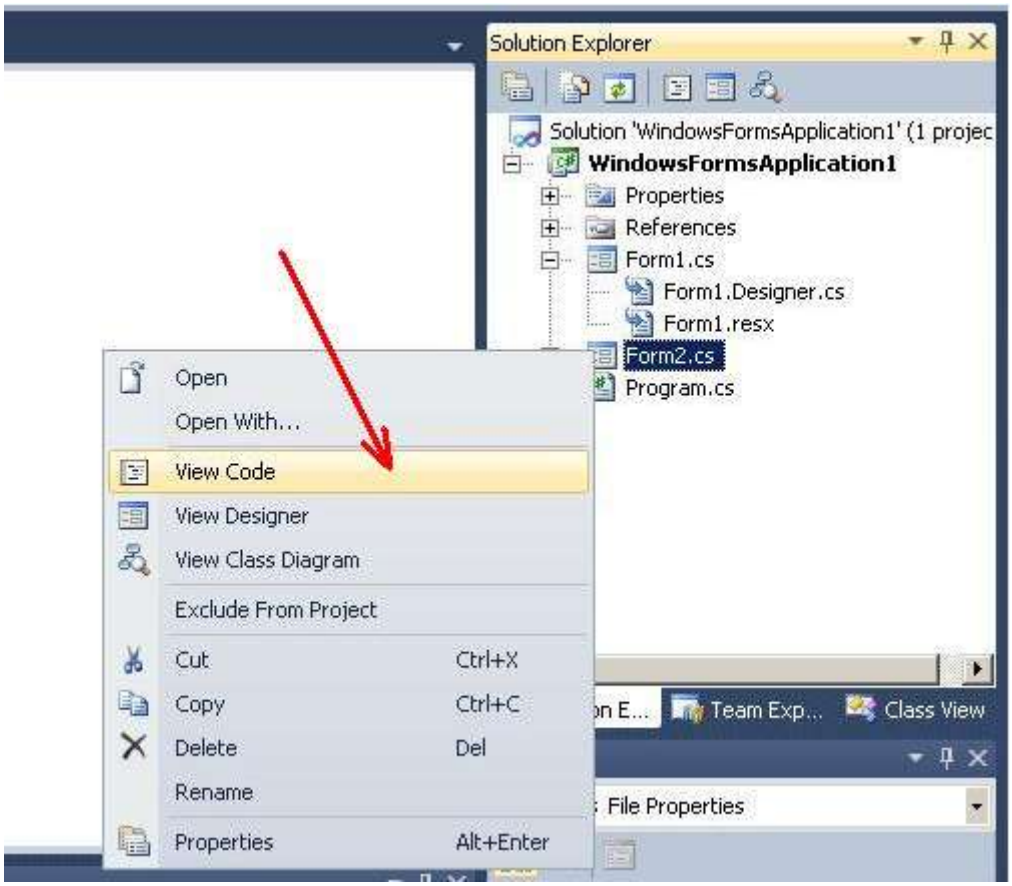

Рис. 9. Команда «View Code» для перехода в режим программного кода

6. Вызов формы Form2 из главной формы приложения

Согласно с условием задачи, для вызова Form2 из Form1 нужно запрограммировать событие клика на кнопке «Show Form 2«.

Программный код обработчика события будет иметь следующий вид:

```
...
private void button1_Click(object sender, EventArgs e)
{      Form2 f = new Form2(); // создаем объект типа Form2
if (f.ShowDialog() == DialogResult.OK) // вызов диалогового окна
формы Form2
{
label1.Text = "Result = OK!";
}
else
{
label1.Text = "Result = Cancel!";
}
```

```
}
...
```

В листинге, приведенном выше, сначала создается экземпляр класса типа Form2. В операторе условного перехода if осуществляется вызов диалогового окна формы Form2 с помощью строки

```
f.ShowDialog();
```

Функция ShowDialog() выводит окно формы и держит его открытым до тех пор, пока пользователь не сделает какой-либо выбор. После выбора пользователем той или иной команды, окно закрывается с кодом возврата. Происходит проверка кода возврата с известными константами класса DialogResult. После проверки выводится сообщение о действии, выбранном пользователем в Form2 (элемент управления label2).

Листинг всего программного кода формы Form1 следующий

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
Form2 f = new Form2();
if (f.ShowDialog() == DialogResult.OK)
{
label1.Text = "Result = OK!";
}
else
{
```

```
label1.Text = "Result = Cancel!";
}
}
}
}
```

7. Выполнение приложения

После выполненных действий можно выполнять приложение и исследовать его работу.