

## Тьюторы

## О курсе

Информация  
Глоссарий  
Дипломы  
Вопросы и ответы  
Студенты  
Рейтинг выпускников  
Мнения  
Учебные программы

## План занятий

Экзамен экстерном

Самостоятельная работа 1

Самостоятельная работа 2

Самостоятельная работа 1

Самостоятельная работа 3

Самостоятельная работа 4

Самостоятельная работа 2

Самостоятельная работа 5

Самостоятельная работа 6

Самостоятельная работа 3

Самостоятельная работа 7

Самостоятельная работа 4

**Самостоятельная работа 8**

Упражнение 1. Прим...  
Создание заготовки...  
Разработка дизайна...  
Программирование э...  
Объект Application  
Упражнение 2. Ока...  
Создание стартовог...  
Последовательность...  
Упражнение 3. Множ...  
Упражнение 4. Поль...  
Упражнение 5. Стра...

Самостоятельная работа 9

Самостоятельная работа 5

Самостоятельная работа 10

Самостоятельная работа 11

Самостоятельная работа 6

Самостоятельная работа 12

Самостоятельная работа 7

Самостоятельная работа 13

Лекция 1

Тест 1

Лекция 2

Лекция 3

Тест 2

Лекция 4

Тест 3

Лекция 5

Тест 4

Лекция 6

Тест 5

Экзамен

Вы можете поддержать этот курс.

## Практикум прикладного программирования на C# в среде VS.NET 2008

[+]

Записаться | Вам нравится? Нравится 17 студентам | Поделиться | Поддержать курс | Скачать электронную книгу

VK Поделиться

f Поделиться 0

Twitter G+ Поделиться 0

Поделиться 0

## Самостоятельная работа 8: Создание приложений WPF

версия для печати

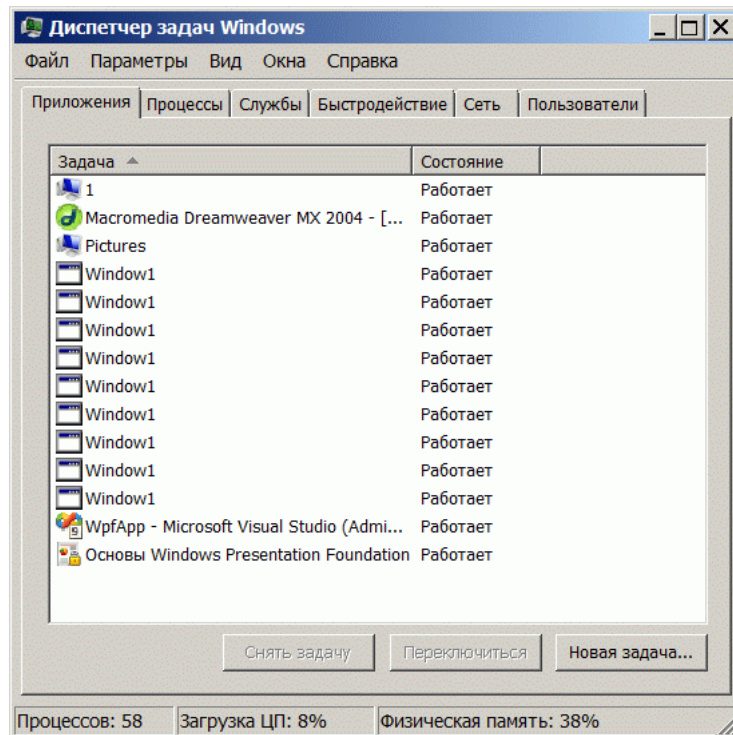
&lt; Самостоятельная работа 7 || Самостоятельная работа 8: 1 2 3 4 5 6 7 8 || Самостоятельная работа 9 &gt;

## Упражнение 3. Множественные окна в WPF

Одно и то же приложение может создавать несколько окон, которые одновременно могут быть открыты. Если этим окнам не назначен владелец, то при закрытии порождающего окна они так и останутся открытыми. Это можно проверить и на последнем примере, если выполнить следующее

- Заменить в обработчике `LifeEvents()` файла `Window0.axml.cs` код модального открытия окна `wnd1.ShowDialog()` ; на `wnd1.Show()` ;
- Запустить приложение и из окна `Window0` открыть несколько экземпляров окна `Window1`, затем закрыть окно `Window0`

После таких действий все дочерние окна остаются открытыми и их нужно закрывать вручную (если только мы их запускали не из под оболочки). К тому же пиктограммы окон не отображаются в панели задач и стоит нам перекрыть экран другим окном, то эти окна придется закрывать либо, предварительно свернув все другие окна, либо через диспетчер задач.



Такое поведение не совсем хорошее, поэтому разработчики платформы WPF предусмотрели в объекте `Application` соответствующие средства. Мы их рассмотрим в данном упражнении.

- Добавьте к решению новый WPF-проект с именем `WpfApp3` и назначьте его стартовым

Искать в кур

## Вопросы и



Але

При рабс  
имен  
нижн  
usin  
usin  
usin  
usin  
usin  
usin  
Micro

```
nam  
{  
    pt  
    Form  
    {
```

граф  
вид

Load  
Even

Form

Form  
m\_F

Main  
send

(осв

прик  
устр

send

пред  
граф

pres  
Pres

пред  
свой

pres  
false  
реж

pres  
1; //

для ,

пере

с ма

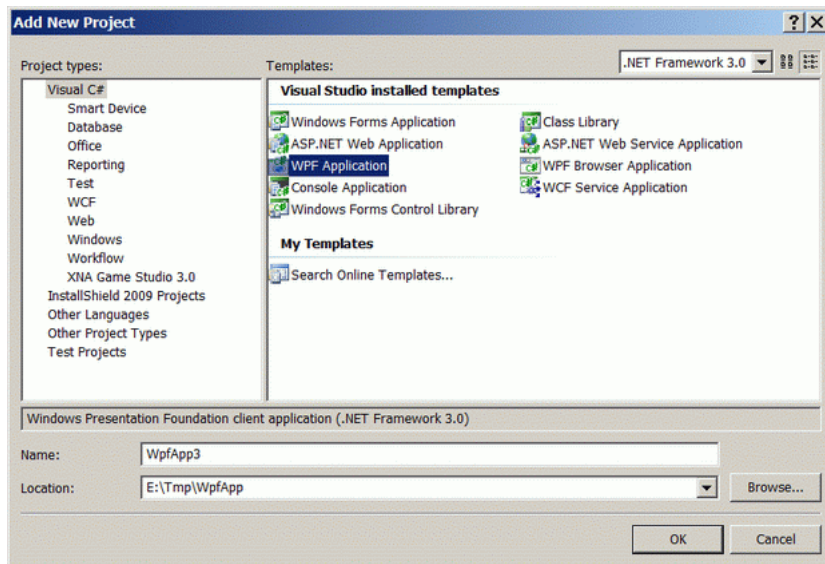
= Sv

задн  
обл

pres  
this.

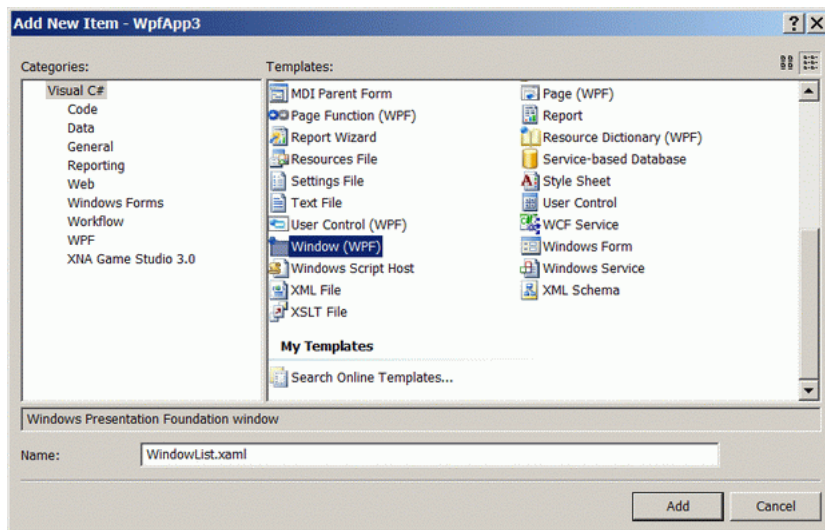
pres  
this.

устр



[увеличить изображение](#)

- В панели Solution Explorer удалите из проекта **WpfApp3** файл с автоматически созданным именем **Window1.xaml** и добавьте новый файл с именем **WindowList.xaml** (чтобы не переименовывать во многих местах)



[увеличить изображение](#)

- Откройте файл **App.xaml** проекта **WpfApp3** и поменяйте значение параметра **StartupUri** объекта **Application** на имя файла нового стартового окна

```
<Application x:Class="WpfApp3.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    StartupUri="WindowList.xaml"
>
</Application>
```

- Откройте файл **WindowList.xaml** в режиме View XAML и настройте стартовое окно следующим образом

```
<Window x:Class="WpfApp3.WindowList"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="WindowList"
    SizeToContent="WidthAndHeight"
    MinWidth="200"
>
    <StackPanel>
        <Button Click="NewWindowClicked">Создать новое окно</Button>
        <Button Click="ListOpenWindows">Список открытых окон</Button>
    </StackPanel>
</Window>
```

- Проследите, чтобы одновременно были созданы зарегистрированные в кнопках обработки
- Модифицируйте файл **WindowList.xaml.cs** следующим образом

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

наст

Грап  
efau  
Devi

pres

OnPi

devi  
work  
e);

}

}

Выб

Невс

сбор

Vers

Publ

d" и

комг

указ


Дел

мож

[OTE](#)

## Студенты

 **Дми**  
Росс  
[про](#)

 **Але**  
Росс  
[про](#)

```

using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace WpfApp3
{
    /// <summary>
    /// Interaction logic for WindowList.xaml
    /// </summary>
    public partial class WindowList : Window
    {
        static int createCount;
        public WindowList()
        {
            InitializeComponent();

            // Определение заголовка окна
            this.Title = "Окно № " + (createCount++).ToString();
        }

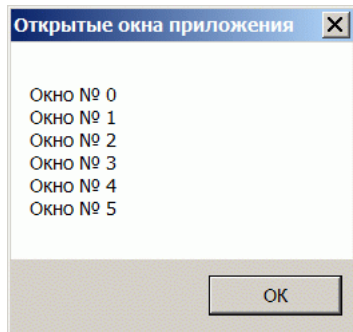
        private void NewWindowClicked(object sender, RoutedEventArgs e)
        {
            new WindowList().Show();
        }

        private void ListOpenWindows(object sender, RoutedEventArgs e)
        {
            StringBuilder sb = new StringBuilder();
            foreach (Window openWindow in Application.Current.Windows)
                sb.AppendLine(openWindow.Title);

            MessageBox.Show(sb.ToString(), "Открытые окна приложения");
        }
    }
}

```

- 🖱️ Запустите приложение - оно хранит и выдает список открытых в текущий момент окон



Однако все окна, открытые приложением, приходится по-прежнему закрывать вручную. Для решения этой проблемы есть два пути:

1. Назначить какое-нибудь из открытых окон главным, присвоив свойству `Application.Current.MainWindow` ссылку на это окно
2. Использовать метод `Application.Current.Shutdown()`

Какой из этих двух вариантов будет работать, определяет свойство `Application.Current.ShutdownMode`. Оно может принимать одно из следующих значений одноименного перечисления:

1. `ShutdownMode.OnLastWindowClose` - закрывать приложение вместе с закрытием последнего окна (установлено по умолчанию)
2. `ShutdownMode.OnMainWindowClose` - закрывать приложение вместе с закрытием главного окна
3. `ShutdownMode.OnExplicitShutdown` - закрывать приложение только тогда, когда явно вызван метод `Shutdown()`

Приложение в режиме `ShutdownMode.OnExplicitShutdown` может закрыться только методом `Shutdown()`, но и в двух предыдущих режимах метод `Shutdown()` работает всегда.

Испытаем оба способа. Для разнообразия сделаем главным окно под номером 3, а во все окна добавим кнопку вызова метода `Shutdown()`.

- 🖱️ Откройте файл `WindowList.xaml` в режиме **View XAML** и добавьте к стартовому окну еще одну кнопку

```

<Window x:Class="WpfApp3.WindowList"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="WindowList"
        SizeToContent="WidthAndHeight"
        MinWidth="200"
        >
    <StackPanel>

```

```

        <Button Click="NewWindowClicked">Создать новое окно</Button>
        <Button Click="ListOpenWindows">Список открытых окон</Button>
        <Button Click="AllCloseWindows">Закрыть все окна</Button>
    </StackPanel>
</Window>

```

- 🔗 Проследите, чтобы одновременно был создан зарегистрированный в кнопке обработчик
- 🔗 Модифицируйте файл `WindowList.xaml.cs` следующим образом

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace WpfApp3
{
    /// <summary>
    /// Interaction logic for WindowList.xaml
    /// </summary>
    public partial class WindowList : Window
    {
        static int createCount;
        public WindowList()
        {
            InitializeComponent();

            // Определение заголовка окна и назначение главного окна
            if (createCount == 3)
            {
                Application.Current.ShutdownMode = ShutdownMode.OnMainWindowClose;
                Application.Current.MainWindow = this;
                this.Title = "Главное окно № " + (createCount++).ToString();
            }
            else
            {
                this.Title = "Окно № " + (createCount++).ToString();
            }
        }

        private void NewWindowClicked(object sender, RoutedEventArgs e)
        {
            new WindowList().Show();
        }

        private void ListOpenWindows(object sender, RoutedEventArgs e)
        {
            StringBuilder sb = new StringBuilder();
            foreach (Window openWindow in Application.Current.Windows)
            {
                sb.AppendLine(openWindow.Title);
            }

            MessageBox.Show(sb.ToString(), "Открытые окна приложения");
        }

        private void AllCloseWindows(object sender, RoutedEventArgs e)
        {
            //Application.Current.ShutdownMode = ShutdownMode.OnExplicitShutdown;
            Application.Current.Shutdown(); // Закрывает при любом режиме
        }
    }
}

```

- 🔗 Запустите приложение и убедитесь, что оно в любом месте закрывается методом `Shutdown()`, а также при закрытии третьего окна

Если установить режим `ShutdownMode.OnExplicitShutdown`, то приложение можно завершить только методом `Shutdown()`.

[Дальше >>](#)

[< Самостоятельная работа 7](#) || **Самостоятельная работа 8:** [1](#) [2](#) [3](#) [4](#) **[5](#)** [6](#) [7](#) [8](#) || [Самостоятельная работа 9 >](#)

