

### **ПЗ 13. Создание проекта с использованием различных управляющих компонентов.**

**Задание: разработайте проект WPF простого текстового редактора:**

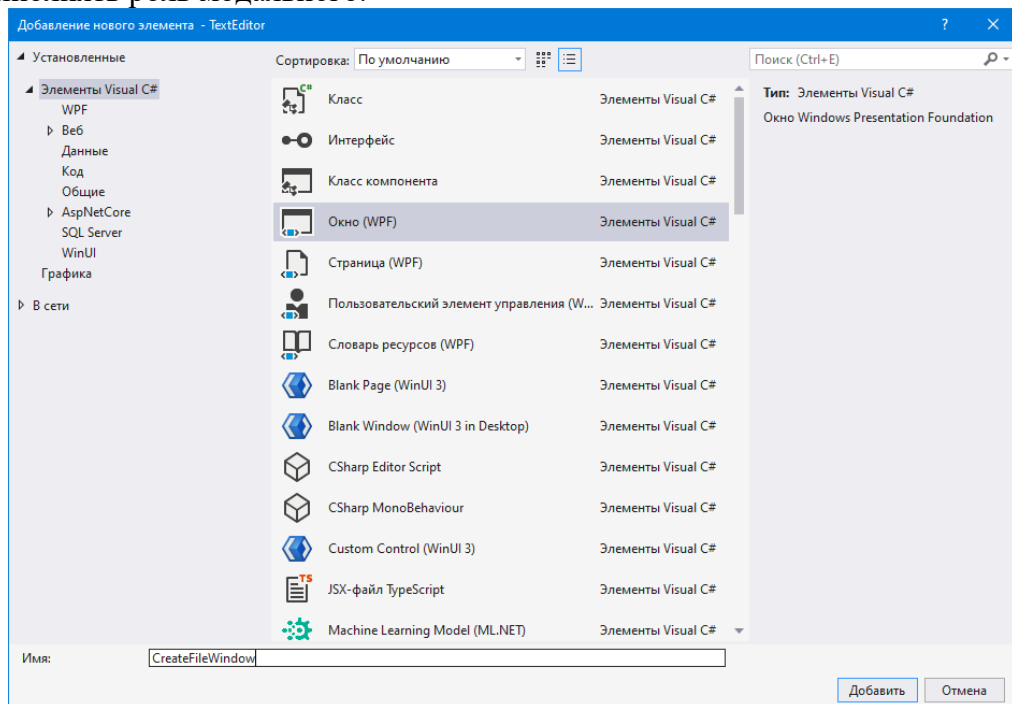
- Внутри проекта создайте папку data (именно в ней будут создаваться и храниться все текстовые документы)
- Функционал создания файла, сохранения и удаления реализоваться строго в отдельных методах: CreateFile(), SaveFile() и DeleteFile(). Внутри этих методов производить подключение к файловым потокам, сохранять/открывать/ удалять файлы и завершать потоки)
- При выборе пункта меню file -> new file открывается диалоговое окно для ввода имени файла. (см. теоретическую часть)
- После ввода имени файла:
  - Файл создается строго в папке data проекта.
  - Имя файла отображается в ListBox (каждый новый файл добавляется в конец списка)
  - Производится работа с текстом в текстовом поле и при выборе пункта меню file - > save все введенные данные сохраняются в текущий файл.
- При выборе файла из ListBox осуществляется загрузка содержимого файла в поле редактирования текста, так же вносятся изменения и по file- > save все изменения сохраняются.

## Теоретическая часть:

### Диалоговые окна

WPF поддерживает возможность создания модальных диалоговых окон. При вызове модальное окно блокирует доступ к родительскому окну, пока пользователь не закроет модальное окно.

Для работы добавим в проект новое окно, которое назовем CreateFileWindow. Это окно будет выполнять роль модального.



Изменим интерфейс CreateFileWindow:

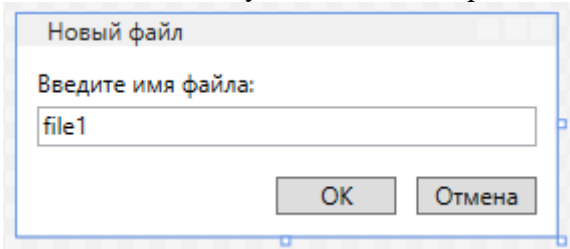
```
<Window x:Class="TextEditor.CreateFileWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:TextEditor"
        mc:Ignorable="d"
        Title="Новый файл" SizeToContent="WidthAndHeight"
        WindowStartupLocation="CenterScreen">
    <Grid Margin="10">
        <Grid.RowDefinitions>
            <RowDefinition Height="20" />
            <RowDefinition Height="20" />
            <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>

        <TextBlock>Введите имя файла:</TextBlock>
        <TextBox Name="fileNameBox" Grid.Row="1" MinWidth="250">file1</TextBox>

        <WrapPanel Grid.Row="2" HorizontalAlignment="Right" Margin="0,15,0,0">
            <Button IsDefault="True" Click="Accept_Click" MinWidth="60"
                Margin="0,0,10,0">OK</Button>
            <Button IsCancel="True" MinWidth="60">Отмена</Button>
        </WrapPanel>
    </Grid>
```

</Window>

Здесь определено текстовое поле для ввода пароля и две кнопки. Вторая кнопка с атрибутом `IsCancel="True"` будет выполнять роль отмены. А первая кнопка будет подтверждать ввод.



Для подтверждения ввода и успешного выхода из модального окна определим в файле кода `CreateFileWindow` обработчик первой кнопки `Accept_Click`:

```
namespace TextEditor
{
    /// <summary>
    /// Логика взаимодействия для CreateFileWindow.xaml
    /// </summary>
    public partial class CreateFileWindow : Window
    {
        public CreateFileWindow()
        {
            InitializeComponent();
        }

        private void Accept_Click(object sender, RoutedEventArgs e)
        {
            this.DialogResult = true;
        }

        public string FileName
        {
            get { return fileNameBox.Text; }
        }
    }
}
```

Для успешного выхода из модального диалогового окна нам надо для свойства `DialogResult` установить значение `true`. Для второй кнопки необязательно определять обработчик, так как у нее установлен атрибут `IsCancel="True"`, следовательно, ее нажатие будет эквивалентно результату `this.DialogResult = false;`. Этот же результат будет при закрытии диалогового окна на крестик.

Кроме того, здесь определяется свойство `FileName`, через которое мы можем извне получить введенное имя.

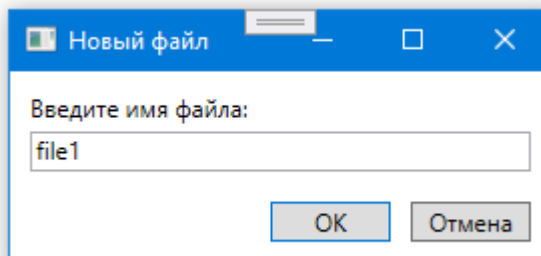
Изменим главную форму `MainWindow`, чтобы из нее запускать диалоговое окно. В-первых, вызов данного диалогового окна будет происходить при выборе пункта меню `File` - `> create new`;

```
...
<MenuItem Header="File">
    <MenuItem x:Name="newFileMenuItem" Header="new file"
Click="newFileMenuItem_Click"></MenuItem>
    <MenuItem x:Name="openFileMenuItem" Header="open file"></MenuItem>
```

И определим обработчик для этого пункта меню:

```
private void newFileMenuItem_Click(object sender, RoutedEventArgs e)
{
    CreateFileWindow createFileWindow = new CreateFileWindow();
    if (createFileWindow.ShowDialog() == true)
        filename = createFileWindow.FileName;
}
```

В итоге при выборе пункта меню будет отображаться следующее диалоговое окно:



## ListBox

Представляет собой обычный список. Содержит коллекцию элементов `ListBoxItem`, которые являются типичными элементами управления содержимым. Также `ListBox` может содержать любые другие элементы, например:

```
<ListBox Name="phonesList">
    <TextBlock FontWeight="Bold" TextDecorations="Underline" Text="Новинки" />
    <ListBoxItem Background="LightGray">LG Nexus 5X</ListBoxItem>
    <ListBoxItem>Huawei Nexus 6P</ListBoxItem>
    <ListBoxItem Background="LightGray">iPhone 6S</ListBoxItem>
    <ListBoxItem>iPhone 6S Plus</ListBoxItem>
    <ListBoxItem Background="LightGray">Asus Zenphone 2</ListBoxItem>
    <ListBoxItem>Microsoft Lumia 950</ListBoxItem>
</ListBox>
```



Все эти элементы будут находиться в коллекции `phonesList.Items` и, таким образом, по счетчику можно к ним обращаться, например,

```
phonesList.Items[0]
```

- первый элемент `ListBox`, который в данном случае представляет `TextBlock`. Также мы можем установить элемент:

```
phonesList.Items[2]="LG G 4";
```

## Выделение элементов

ListBox поддерживает множественный выбор. Для этого нужно установить свойство `SelectionMode="Multiple"` или `SelectionMode="Extended"`. В последнем случае, чтобы выделить несколько элементов, необходимо держать нажатой клавишу `Ctrl` или `Shift`. По умолчанию `SelectionMode="Single"`, то есть допускается только единственное выделение.