

Лабораторная работа №5

«Страничный интерфейс»

Цели и задачи: Научиться создавать и использовать страничный интерфейс вместо стандартных окон в приложениях WPF.

Теоретическая часть:

Одним из вариантов создания пользовательского интерфейса в приложениях WPF является использование страниц (Page). Данный вид пользовательского интерфейса позволяет разработчикам создавать настольные (desktop) приложения в стиле Web приложений.

Существуют два типа страничных WPF приложений:

1. автономные приложения Windows, в которых страницы используются для части или всего пользовательского интерфейса. Данный подход следует применять в простых приложениях, чтобы навигационные и журнальные функции позволили упростить кодирование;

2. браузерные приложения, которые выполняются в браузерах и имеют ограниченные полномочия. Данный подход следует применять, когда необходимо получить облегчённую, основанную на web-технологиях модель развёртывания.

Модель для создания страниц в WPF во многом схожа с моделью для создания окон Windows. Для каждой страницы WPF создаётся файл XAML и файл отделённого (partial) процедурного кода. Создавать объекты страниц можно и с помощью одного лишь кода. На этапе компиляции для каждого модуля страницы компилятор создаёт производный класс, который объединяет написанный разработчиком код с кодом связующих элементов, который генерируется автоматически. К таким элементам, прежде всего, относятся поля, которые ссылаются на каждый именованный элемент на странице.

На этапе проектирования приложения страницы являются самым высокоуровневым компонентом пользовательского интерфейса. Но во время выполнения приложения контейнером наивысшего уровня они уже не будут, они будут обслуживаться в другом контейнере, который владеет страницами и называется хостом.

В качестве такого контейнера WPF позволяет использовать:

- класс `NavigationWindow` (или элемент «`NavigationWindow`»), который представляет собой видоизменённую версию класса `Window`;
- элемент «`Frame`», находящийся внутри другого окна;
- элемент «`Frame`», находящийся внутри другой страницы;
- элемент «`Frame`», обслуживаемый непосредственно в Internet Explorer или Firefox.

При создании автономного приложения WPF можно:

- установить свойство «StartupUri» элемента «Application» так, чтобы оно указывало на страницу;
- создать объект «NavigationWindow» вручную и затем загрузить первую страницу внутрь него;
- вставить страницы в специальное окно с помощью элемента управления «Frame».

Эти подходы предоставляют возможность задавать размер окна приложения, что является важным для придания приложению презентабельного вида при его первом запуске.

В случае создания браузерного приложения возможности для управления размером содержащего окна web-браузера не предоставляется и потому обязательно потребуется установить свойство «StartupUri» так, чтобы оно указывало на страницу.

Классы «NavigationWindow» и «Frame» определены в WPF как навигаторы. Навигатор - это класс, поддерживающий переход и журнал переходов.

Пример создания приложения WPF

Создадим примитивное приложение WPF. И заменим тег «Window» на тег «NavigationWindow».

```
<NavigationWindow x:Class="WpfPages.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Страничное приложение" Height="300" Width="500">

</NavigationWindow>
```

Также следует изменить в файле «MainWindow.xaml» базовый класс «Window» на «NavigationWindow».

Результат компилирования и запуска приложения отображен на рисунке 1.

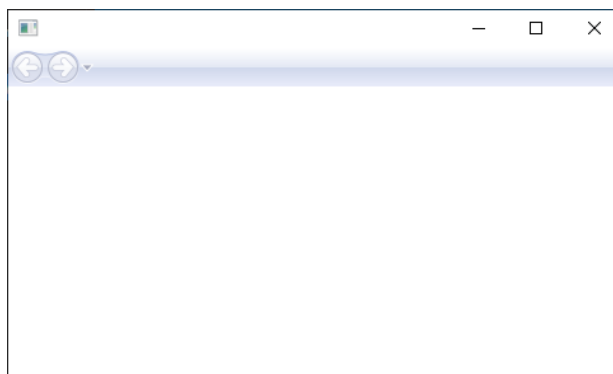


Рисунок 1 – Пустое приложение со страничным интерфейсом

Для добавления новой странице необходимо сделать клик правой кнопкой мыши в обозревателе решения по названию проекта (не путать с решением) и выбрать пункт «добавить», затем «страница» (рисунок 2). Откроется окно добавления нового элемента, в котором необходимо задать наименование новой страницы «Home». Также необходимо установить ширину и высоту, соответствующие соответствующим значениям главного окна. (в примере – ширина 500, высота 300).

Аналогично создадим еще одну страницу и назовем ее «Report».

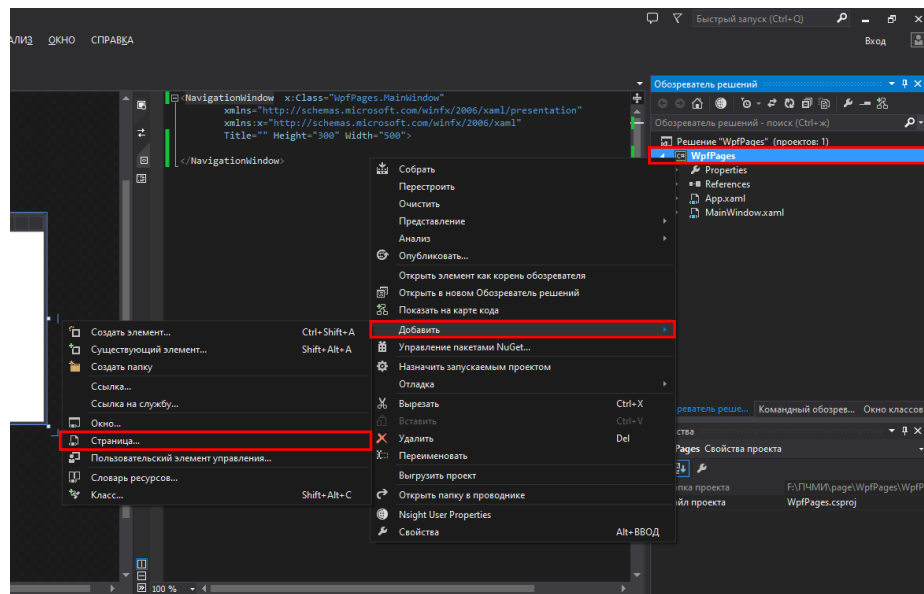


Рисунок 2 – Добавление новой страницы в проект

На странице «Home» установим необходимые параметры для элемента «Grid»:

```
<Grid Margin="10,0,10,10">
  <Grid.ColumnDefinitions>
    <ColumnDefinition />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition />
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>
</Grid>
```

В выше показанном устанавливаются отступы для сетки (верх – 0, остальные - 10), также добавляются три строки и одна колонка.

Добавим на данную страницу несколько элементов управления, а именно, «Label», «ListBox» и «Button». В список добавим несколько имен для примера. На данном этапе листинг выглядит следующим образом:

```
<Page x:Class="WpfPages.Home"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

```

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="500"
Title="Home">

<Grid Margin="10,0,10,10">
    <Grid.ColumnDefinitions>
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition />
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <Border Grid.Column="0"
        Grid.Row="0"
        Height="35"
        Padding="5"
        Background="Blue">
        <Label VerticalAlignment="Center"
            Foreground="White">Имена</Label>
    </Border>
    <ListBox Name="lbPeople"
        Grid.Column="0"
        Grid.Row="1">
        <ListBoxItem>Андрей</ListBoxItem>
        <ListBoxItem>Вася</ListBoxItem>
        <ListBoxItem>Илья</ListBoxItem>
        <ListBoxItem>Мария</ListBoxItem>
    </ListBox>
    <Button Grid.Column="0"
        Grid.Row="2"
        Margin="0,10,0,0"
        Width="125"
        Height="25"
        HorizontalAlignment="Right">Показать</Button>
</Grid>
</Page>

```

Для отображения нашей созданной страницы в свойстве главного окна «Source» укажем нашу домашнюю страницу (в данном примере это «Home.xaml»). Результат выполнения приложения на данном этапе отображен на рисунке 3.

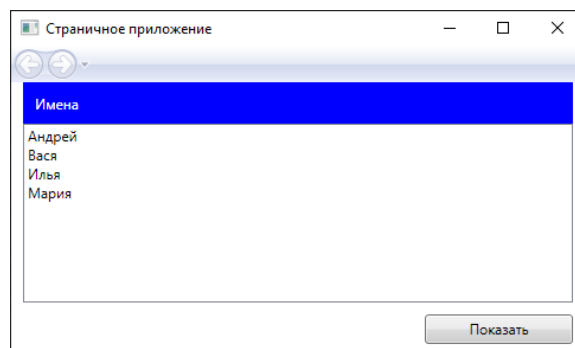


Рисунок 3 – Страничное приложение WPF с загруженной страницей

Создадим обработчик события для кнопки на странице «Home». Для этого в конструкторе макета приложения достаточно кликнуть дважды по созданной кнопке. Созданный обработчик приведем к следующему виду:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    Report reportPage = new Report(this.lbPeople.SelectedItem);
    this.NavigationService.Navigate(reportPage);
}
```

В выше приведенном коде класс «Report» это наша вторая, ранее созданная, страница. XAML разметку для страницы «Report» приведем к следующему виду:

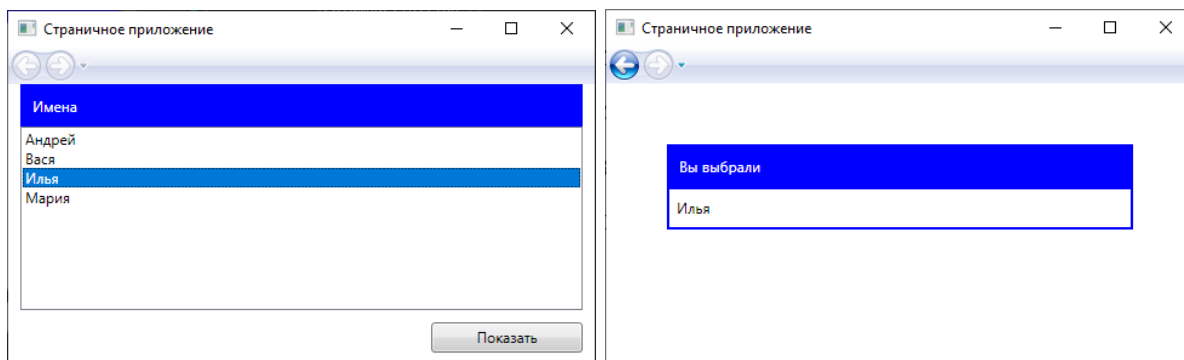
```
<Page x:Class="WpfPages.Report"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="500" Title="Report">
    <StackPanel Margin="50">
        <Border Height="35"
                Padding="5"
                Background="Blue">
            <Label Foreground="White"
                    Content="Вы выбрали"/>
        </Border>
        <Border Height="35"
                Padding="1"
                BorderThickness="2"
                BorderBrush="Blue">
            <Label Name="lName"
                    VerticalAlignment="Center"/>
        </Border>
    </StackPanel>
</Page>
```

Обратите внимание, что в приведенном выше листинге вместо тега «Grid» используется «StackPanel». Данный элемент (тег) располагает элементы внутри себя последовательно, друг за другом.

Также для корректной компиляции и выполнения необходимо добавить новый конструктор класса «Report». Для этого перейдем к файлу «Report.xaml.cs» и добавим следующий код, ниже конструктора без параметров:

```
public Report(object data) : this()
{
    lName.Content = data is ListBoxItem ?
        (data as ListBoxItem).Content.ToString() : "Ничего";
}
```

При удачной компиляции проекта, будет запущено окно с выбором имен, в котором при нажатии на кнопку «Показать» в этом же окне откроется вторая страница. Обратите внимание, что для возврата на предыдущую страницу необходимо нажать кнопку «Назад» (рисунок 4).



а)

б)

Рисунок 4 – Пример работы приложения

а) Начальная страница

б) Вторая страница приложения

Задания на лабораторную работу

1. Изучить материал, представленный в данных методических указаниях к лабораторной работе, а также материал находящийся в сети интернет в свободном доступе по данной теме;
2. Разработать приложение, согласно заданию преподавателя;
3. Составить отчёт. В отчёте отобразить:
 - цели и задачи лабораторной работы;
 - исходный код приложения (отображающий только логику программы и описание работы);
 - код XAML;
 - скриншоты работы приложения;
 - выводы по данной лабораторной работе.