

Оглавление

Оглавление	2
1 Создание приложений WPF в Visual Studio	3
1.1 Задание	3
1.2 Код программы	3
1.3 Результат работы приложения.....	7
2 Страничная навигация в приложениях WPF	8
2.1 Задание	8
2.2 Код программы	8
2.3 Результат работы приложения.....	12
3 Ответы на контрольные вопросы.....	13
4 Выводы	15

1 Создание приложений WPF в Visual Studio

1.1 Задание

Разработать приложение WPF, которое выполняет параметрические запросы к массиву объектов из лабораторной работы №1. Параметры запроса вводятся через различные элементы управления (TextBox, ComboBox, DatePicker и др.) и подставляются в предложения where запросов LINQ. Результаты запросов выводятся через окно сообщений (MessageBox).

1.2 Код программы

Напишем код документа XAML для главного окна приложения (MainWindow.xaml):

```
<Window x:Class="ЛП_3.MainWindow"
        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:ЛП_3"
        mc:Ignorable="d"
        Title="Билеты на междугородные автобусы (Конторин П.Л., ДЦИС-37)" Height="492"
Width="910"
        WindowStartupLocation="CenterScreen">
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="200" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="312" />
            <ColumnDefinition />
        </Grid.ColumnDefinitions>
        <!--Панель со списком билетов-->
        <StackPanel Grid.Row="0" Grid.Column="0" Orientation="Vertical"
Background="LightGray">
            <Label Content="Список билетов:" FontSize="20" Margin="10,0"/>
            <RadioButton Name = "t0" Checked = "RadioButton_Checked"
GroupName = "Ticket" Content="ИБ116 - Иваново" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
            <RadioButton Name = "t1" Checked = "RadioButton_Checked"
GroupName = "Ticket" Content="КС319 - Кострома" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
            <RadioButton Name = "t2" Checked = "RadioButton_Checked"
GroupName = "Ticket" Content="ИБ414 - Иваново" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
            <RadioButton Name = "t3" Checked = "RadioButton_Checked"
GroupName = "Ticket" Content="РС514 - Ростов" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
        </StackPanel>
    </Grid>
</Window>
```

```

        <RadioButton Name = "t4" Checked = "RadioButton_Checked"
        GroupName = "Ticket" Content="ВЛ907 - Владимир" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
        <RadioButton Name = "t5" Checked = "RadioButton_Checked"
        GroupName = "Ticket" Content="КС403 - Кострома" FontSize="20"
VerticalContentAlignment="Center" Margin="15,0"/>
    </StackPanel>
    <!--Панель с информацией о билете-->
    <Border Grid.Row="1" Grid.Column="0" BorderBrush="LightGray" Background="White"
BorderThickness="10" DockPanel.Dock="Top">
        <StackPanel Grid.Row="1" Grid.Column="0" Orientation="Vertical">
            <Label Content="Информация о билете:" FontSize="20"/>
            <TextBlock Name="TicketInfo" Text="" FontSize="20"/>
        </StackPanel>
    </Border>
    <!--Панель с элементами для выполнения параметрических запросов-->
    <StackPanel Grid.Row="0" Grid.Column="1" Grid.RowSpan="2" Orientation="Vertical"
Grid.ColumnSpan="2" Background="LightGray">
        <TextBlock Text="1. Рейсы с временем отправления T" Height="56" FontSize="20"
Padding="10,15"/>
        <StackPanel Height="55" Orientation="Horizontal" Background="White"
Margin="0,0,10,0">
            <Label Content="Введите T:" Width="115" FontSize="20" Padding="10,13"/>
            <TextBox Name="textbox1" Margin="100,10,0,10" Width="100" FontSize="20"/>
            <Button x:Name="button1" Click="Button1_Click" Content="Найти" Height="35"
Width="120" FontSize="20" Margin="105,10,100,10"/>
        </StackPanel>
        <TextBlock Text="2. Количество билетов с номерами мест от N1 до N2:"
Height="56" FontSize="20" Padding="10,15"/>
        <StackPanel Height="55" Orientation="Horizontal" Background="White"
Margin="0,0,10,0">
            <Label Width="50" Content="N1:" FontSize="20" Padding="10,13"/>
            <TextBox Name="textboxN1" Margin="0,10,0,10" Width="50" FontSize="20"/>
            <Label Width="50" Content="N2:" FontSize="20" Padding="10,13"
Margin="110,0,0,0"/>
            <TextBox Name="textboxN2" Margin="0,10,0,10" Width="50" FontSize="20"/>
            <Button Name="button2" Click="Button2_Click" Content="Найти" Height="35"
Width="120" FontSize="20" Margin="110,0"/>
        </StackPanel>
        <TextBlock Text="3. Пункты назначения и рейсы с длительностью более D мин:"
Height="56" FontSize="20" Padding="5,15"/>
        <StackPanel Height="55" Orientation="Horizontal" Background="White"
Margin="0,0,10,0">
            <Label Width="273" Content="Введите длительность, мин:" FontSize="20"
Padding="10,13" Margin="0,0,0,0.2"/>
            <TextBox Name="textbox3" Margin="0,10,0,10" Width="50" FontSize="20"/>
            <Button Name="button3" Click="Button3_Click" Content="Найти" Height="35"
Width="120" FontSize="20" Margin="100,0"/>
        </StackPanel>
        <TextBlock Text="4. Средняя длительность рейсов в пункт назначения P:"
Height="56" FontSize="20" Padding="5,15"/>
        <StackPanel Height="55" Orientation="Horizontal" Background="White"
Margin="0,0,10,0">
            <Label Width="286" Content="Выберите пункт назначения:" FontSize="20"
Padding="10,13"/>
            <ComboBox Name="combobox" Width="120" Margin="0,10,0,10">
                <TextBlock FontSize="20">Иваново</TextBlock>
                <TextBlock FontSize="20">Кострома</TextBlock>
                <TextBlock FontSize="20">Ростов</TextBlock>
                <TextBlock FontSize="20">Владимир</TextBlock>
            </ComboBox>
            <Button Name="button4" Click="Button4_Click" Content="Найти" Height="35"
Width="120" FontSize="20" Margin="16,0"/>
        </StackPanel>
    </StackPanel>

```

```
</Grid>
</Window>
```

Напишем связанный с данным документом код на C# (модуль MainWindow.xaml.cs), содержащий обработчики событий:

```
using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;

namespace ЛР_3
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        class Ticket
        {
            // Автоматические свойства класса Ticket
            public string TicketID { get; set; } // Код билета
            public int Trip { get; set; } // Рейс
            public string Point { get; set; } // Пункт назначения
            public string Departure { get; set; } // Время отправления
            public int Duration { get; set; } // Длительность
            public int Number { get; set; } // Номер места
            public string BusID { get; set; } // Код автобуса
            // Переопределение метода ToString для вывода информации об объектах
            public override string ToString()
            {
                return string.Format( " - Код билета: {0}\n" +
                                     " - Рейс: {1}\n" +
                                     " - Пункт назначения: {2}\n" +
                                     " - Время отправления: {3}\n" +
                                     " - Длительность: {4} мин\n" +
                                     " - Номер места: {5}\n" +
                                     " - Код автобуса: {6}\n",
                                     TicketID,
                                     Trip,
                                     Point,
                                     Departure,
                                     Duration,
                                     Number,
                                     BusID
                                    );
            }
        }
        // Объявление массива объектов класса Ticket с инициализацией элементов
        private Ticket[] tickets = new[]
        {
            new Ticket { TicketID = "ИБ116",
                         Trip = 307,
                         Point = "Иваново",
                         Departure = "08:30",
                         Duration = 120,
                         Number = 14,
                         BusID = "A071"
                       },
            new Ticket { TicketID = "KC319",
                         Trip = 418,
                         Point = "Кострома",
                         Departure = "08:30",

```

```

        Duration = 100,
        Number = 05,
        BusID = "A019"
    },
    new Ticket {
        TicketID = "ИБ414",
        Trip = 312,
        Point = "Иваново",
        Departure = "13:40",
        Duration = 130,
        Number = 20,
        BusID = "B027"
    },
    new Ticket {
        TicketID = "РС514",
        Trip = 207,
        Point = "Ростов",
        Departure = "16:00",
        Duration = 80,
        Number = 31,
        BusID = "A071"
    },
    new Ticket {
        TicketID = "ВЛ907",
        Trip = 425,
        Point = "Владимир",
        Departure = "18:40",
        Duration = 200,
        Number = 19,
        BusID = "A019"
    },
    new Ticket {
        TicketID = "КС403",
        Trip = 315,
        Point = "Кострома",
        Departure = "20:20",
        Duration = 80,
        Number = 27,
        BusID = "A071"
    },
    },
};
public MainWindow()
{
    InitializeComponent();
}
// Вывести информацию о билете по выбранному положению переключателя RadioButton
private void RadioButton_Checked(object sender, RoutedEventArgs e)
{
    RadioButton rb = sender as RadioButton;
    TicketInfo.Text = tickets[Convert.ToInt16(rb.Name.Remove(0, 1))].ToString();
}
// Получить рейсы с временем отправления T при нажатии button1
private void Button1_Click(object sender, RoutedEventArgs e)
{
    string tb = textbox1.Text;
    var trip = from t in tickets
               where t.Departure == tb
               select t.Trip;
    string mb = "";
    foreach (var t in trip) mb += string.Format("- {0}\n", t.ToString());
    MessageBox.Show(mb, "Рейсы с временем отправления T");
}
// Получить количество билетов с номерами мест от N1 до N2 при нажатии button2
private void Button2_Click(object sender, RoutedEventArgs e)
{
    int n1 = int.Parse(textboxN1.Text);
    int n2 = int.Parse(textboxN2.Text);
    var num = from t in tickets
               where t.Number >= n1 && t.Number <= n2

```

```

        select t.Number;
        string mb = string.Format("{0} шт.", num.Count());
        MessageBox.Show(mb, "Количество билетов с номерами мест от N1 до N2");
    }
    // Получить пункты назначения и рейсы с длительностью более D мин при нажатии
button3
private void Button3_Click(object sender, RoutedEventArgs e)
{
    int d = int.Parse(textbox3.Text);
    var pointTrip = from t in tickets
                    where t.Duration > 100
                    select t;

    string mb = "";
    foreach (var p in pointTrip) mb += string.Format("- {0}, {1}\n",
p.Point.ToString(), p.Trip.ToString());
    MessageBox.Show(mb, "Пункты назначения и рейсы с длительностью более D мин");
}
// Получить среднюю длительность рейсов в пункт назначения P при нажатии button4
private void Button4_Click(object sender, RoutedEventArgs e)
{
    var averDur = from t in tickets
                  where t.Point == combobox.Text
                  select t.Duration;
    string mb = string.Format("{0} мин", averDur.Average());
    MessageBox.Show(mb, "Средняя длительность рейсов в пункт назначения P");
}
}
}

```

1.3 Результат работы приложения

Окно запущенного приложения WPF показано на рисунке 1.1.

Рисунок 1.1 – Окно работающего приложения WPF

Результат одного из параметрических запросов, выводимых в окне сообщения, представлен на рисунке 1.2.

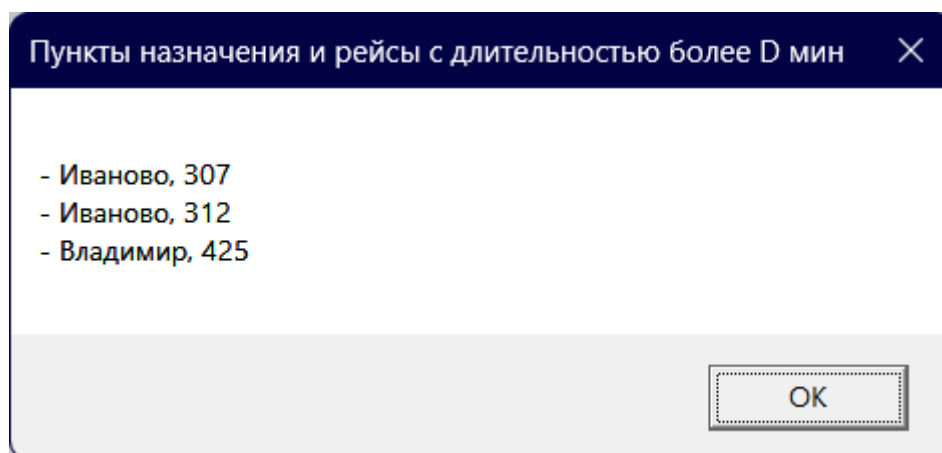


Рисунок 1.2 – Окно сообщения при выполнении запроса №3

2 Страничная навигация в приложениях WPF

2.1 Задание

Требуется разработать приложение WPF, использующее страничную навигацию. В создаваемое приложение необходимо добавить четыре страницы, каждая из которых должна содержать заголовок (элемент Label), главное меню (элемент Menu) и строку состояния (элемент StatusBar). Одна из страниц (главная) должна содержать ссылки (элементы Hyperlink) для переходов на остальные страницы. Три другие страницы должны соответствовать сущностям, определенным в XML-документе из работы №2. В каждой из этих страниц должна присутствовать таблица (элемент DataGrid) с определенным набором столбцов (на основе данных из XML-документа).

2.2 Код программы

Напишем код XAML-документа MainWindow.xaml (главное окно):

```
<Window x:Class="ЛР_3_2.MainWindow"
        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"
```

```

        mc:Ignorable="d"
        Title="Банковские услуги" Height="350" Width="500"
        WindowStartupLocation="CenterScreen">
<Grid>
    <Frame Source="MainPage.xaml"/>
</Grid>
</Window>

```

Напишем код XAML-документа MainPage.xaml (главная страница):

```

<Page x:Class="ЛР_3_2.MainPage"
    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="350" d:DesignWidth="500"
    Title="Главная страница">
<Grid>
    <!--Описание строк и столбцов панели Grid-->
    <Grid.RowDefinitions>
        <RowDefinition Height="20"/>
        <RowDefinition Height="50"/>
        <RowDefinition Height="*/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="20"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="200"/>
        <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>
    <!--Панель с главным меню приложения-->
    <StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2">
        <Menu/>
    </StackPanel>
    <!--Панель с заголовком страницы-->
    <StackPanel Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2">
        <Label>Банковские услуги</Label>
    </StackPanel>
    <!--Панель со ссылками на другие страницы-->
    <StackPanel Grid.Row="2" Grid.Column="0">
        <TextBlock Margin="25, 15, 0, 0">
            <Hyperlink NavigateUri="BanksPage.xaml">Банки</Hyperlink>
        </TextBlock>
        <TextBlock Margin="25, 15, 0, 0">
            <Hyperlink NavigateUri="DepositsPage.xaml">Вклады</Hyperlink>
        </TextBlock>
        <TextBlock Margin="25, 15, 0, 0">
            <Hyperlink NavigateUri="DepositorsPage.xaml">Вкладчики</Hyperlink>
        </TextBlock>
    </StackPanel>
    <!--Панель с изображением-->
    <StackPanel Grid.Row="2" Grid.Column="1">
        <Image Name="logo" Source="Images\142646.png" Height="158" Margin="0,0,84,0"/>
    </StackPanel>
    <!--Панель с данными исполнителя-->
    <StackPanel Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2"
HorizontalAlignment="Center">
        <TextBlock>Работу выполнил: Конторин П.Л., ДЦИС-37</TextBlock>
    </StackPanel>
    <!--Панель со строкой состояния-->
    <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
        <StatusBar/>
    </StackPanel>

```



```
</Grid>
</Page>
```

Напишем код XAML-документа BanksPage.xaml (страница Банки):

```
<Page x:Class="ЛР_3_2.BanksPage"
      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="350" d:DesignWidth="500"
      Title="Страница банки">
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="20"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>
        <!--Панель с главным меню приложения-->
        <StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2">
            <Menu/>
        </StackPanel>
        <!--Панель с заголовком страницы-->
        <StackPanel Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2">
            <Label>Банки</Label>
        </StackPanel>
        <!--Панель с таблицей-->
        <StackPanel Grid.Row="2" Grid.Column="0">
            <DataGrid Name="BanksDataGrid">
                <DataGrid.Columns>
                    <DataGridTextColumn Header="БИК" Width="250"/>
                    <DataGridTextColumn Header="Название" Width="*/>
                </DataGrid.Columns>
            </DataGrid>
        </StackPanel>
        <!--Панель со строкой состояния-->
        <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
            <StatusBar/>
        </StackPanel>
    </Grid>
</Page>
```

Напишем код XAML-документа DepositsPage.xaml (страница Вклады):

```
<Page x:Class="ЛР_3_2.DepositsPage"
      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="350" d:DesignWidth="500"
      Title="Страница вклады">
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
    </Grid>
```

```

        <RowDefinition Height="50"/>
        <RowDefinition Height="*/>
        <RowDefinition Height="20"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>
    <!--Панель с главным меню приложения-->
    <StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2">
        <Menu/>
    </StackPanel>
    <!--Панель с заголовком страницы-->
    <StackPanel Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2">
        <Label>Вклады</Label>
    </StackPanel>
    <!--Панель с таблицей-->
    <StackPanel Grid.Row="2" Grid.Column="0">
        <DataGrid Name="BanksDataGrid">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Код" Width="166"/>
                <DataGridTextColumn Header="Вид" Width="166"/>
                <DataGridTextColumn Header="Процент" Width="*/>
            </DataGrid.Columns>
        </DataGrid>
    </StackPanel>
    <!--Панель со строкой состояния-->
    <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
        <StatusBar/>
    </StackPanel>
</Grid>
</Page>

```

Напишем код XAML-документа DepositorsPage.xaml (страница Вкладчики):

```

<Page x:Class="ЛР_3_2.DepositorsPage"
    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="350" d:DesignWidth="500"
    Title="Страница вкладчики">
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="20"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>
        <!--Панель с главным меню приложения-->
        <StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2">
            <Menu/>
        </StackPanel>
        <!--Панель с заголовком страницы-->
        <StackPanel Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2">
            <Label>Вкладчики</Label>
        </StackPanel>
        <!--Панель с таблицей-->
    </Grid>

```

```

<StackPanel Grid.Row="2" Grid.Column="0">
    <DataGrid Name="BanksDataGrid">
        <DataGrid.Columns>
            <DataGridTextColumn Header="ИНН" Width="166"/>
            <DataGridTextColumn Header="ФИО" Width="166"/>
            <DataGridTextColumn Header="Сумма" Width="*/>
        </DataGrid.Columns>
    </DataGrid>
</StackPanel>
<!--Панель со строкой состояния-->
<StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
    <StatusBar/>
</StackPanel>
</Grid>
</Page>

```

2.3 Результат работы приложения

Окно работающего приложения WPF с загруженными в него страницами показано на рисунках 2.1 и 2.2.

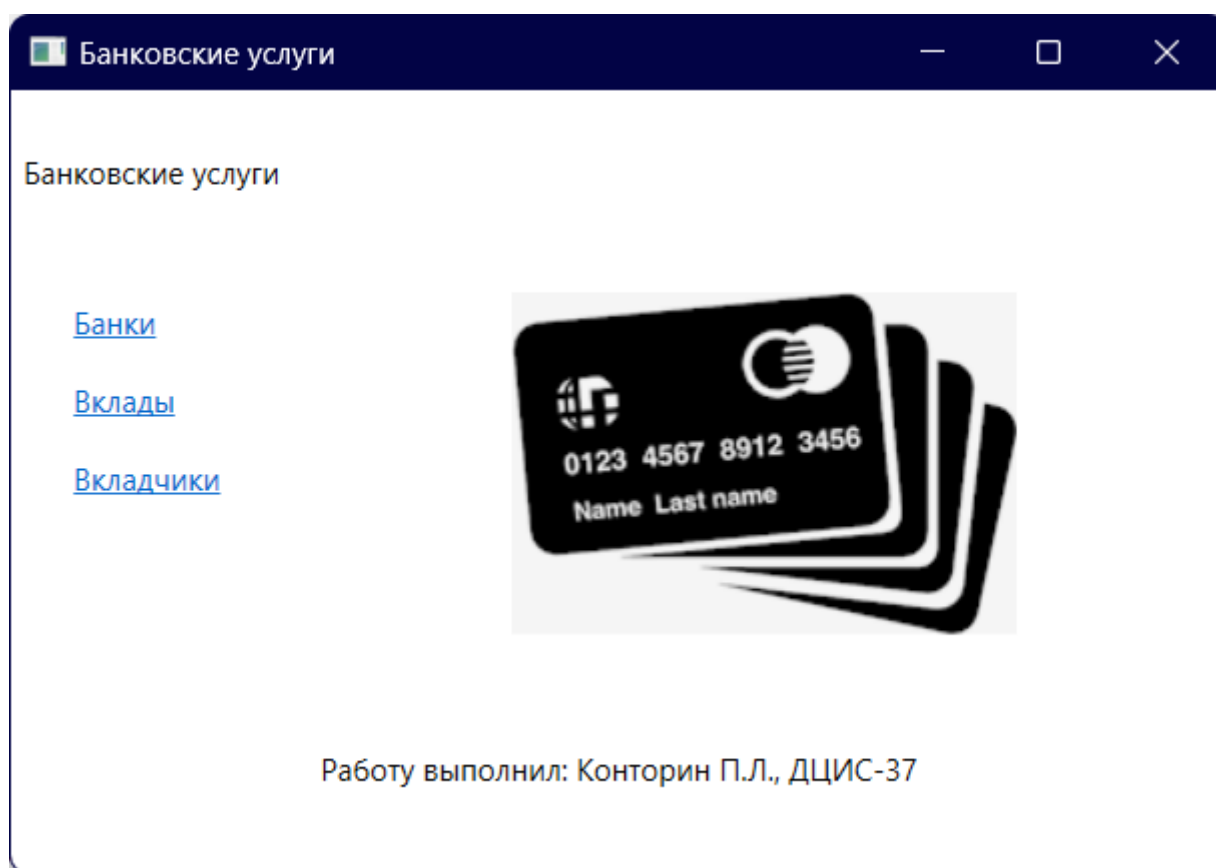


Рисунок 2.1 – Окно приложения с загруженной главной страницей

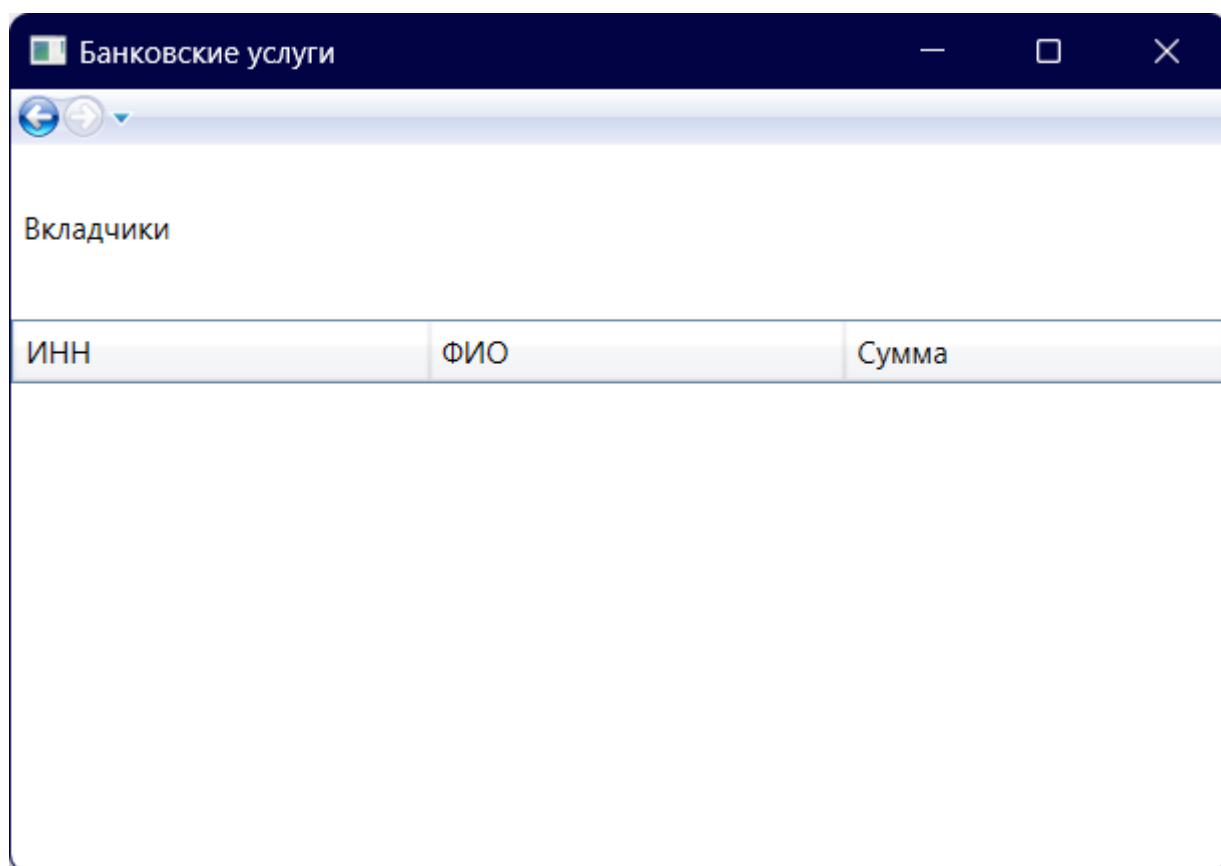


Рисунок 2.2 – Окно приложения с загруженной страницей «Вкладчики»

Оглавление

Оглавление	2
1 Привязка данных WPF	3
1.1 Задание	3
1.2 Код программы	3
1.3 Результат работы приложения.....	7
2 Шаблоны данных.....	8
2.1 Задание	8
2.2 Код программы	8
2.3 Результат работы приложения.....	14
3 Ответы на контрольные вопросы.....	16
4 Выводы	17

1 Привязка данных WPF

1.1 Задание

Требуется разработать приложение, которое позволяет модифицировать XML-документ, полученный в лабораторной работе №2 из массива объектов. Данное приложение должно выводить данные по всем элементам, обеспечивать возможность редактирования данных, а также добавления и удаления элементов.

1.2 Код программы

В качестве XML-документа воспользуемся файлом ticket.xml, полученным в лабораторной работе №2:

```
<?xml version="1.0" encoding="utf-8"?>
<ticketBus>
  <ticket id="ИБ116" trip="307">
    <point>Иваново</point>
    <departure>08:30</departure>
    <duration unit="мин">120</duration>
    <number>14</number>
  </ticket>
  <ticket id="КС319" trip="418">
    <point>Кострома</point>
    <departure>08:30</departure>
    <duration unit="мин">100</duration>
    <number>5</number>
  </ticket>
  <ticket id="ИБ414" trip="312">
    <point>Иваново</point>
    <departure>13:40</departure>
    <duration unit="мин">130</duration>
    <number>20</number>
  </ticket>
  <ticket id="РС514" trip="207">
    <point>Ростов</point>
    <departure>16:00</departure>
    <duration unit="мин">80</duration>
    <number>31</number>
  </ticket>
  <ticket id="ВЛ907" trip="425">
    <point>Владимир</point>
    <departure>18:40</departure>
    <duration unit="мин">200</duration>
    <number>19</number>
  </ticket>
  <ticket id="КС403" trip="315">
    <point>Кострома</point>
    <departure>20:20</departure>
    <duration unit="мин">80</duration>
    <number>27</number>
  </ticket>
</ticketBus>
```

```
</ticket>
</ticketBus>
```

Напишем код XAML приложения WPF:

```
<Window x:Class="ЛР_4_1.MainWindow"
        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"
        mc:Ignorable="d"
        Title="Билеты на междугородные автобусы (Конторин П.Л., ДЦИС-37)"
        Height="550" Width="700" Loaded="Window_Loaded"
        WindowStartupLocation="CenterScreen">
    <Window.Resources>
        <!--Определение шаблона данных для вывода списка товаров-->
        <DataTemplate x:Key="TicketDataTemplate" DataType="ticket">
            <Border Margin="5" BorderThickness="2" BorderBrush="Orange"
                    CornerRadius="5">
                <StackPanel>
                    <StackPanel Orientation="Horizontal">
                        <TextBlock Margin="3" Text="{Binding Path=Attribute[id].Value}"
                                   FontWeight="Bold"/>
                        <TextBlock Margin="3" Text=" - "/>
                        <TextBlock Margin="3" Text="{Binding Path=Element[point].Value}"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal">
                        <TextBlock Margin="3" Text="отправление в"/>
                        <TextBlock Margin="3" Text="{Binding
                                                Path=Element[departure].Value}"/>
                    </StackPanel>
                </StackPanel>
            </Border>
        </DataTemplate>
        <!--Определение стиля для текстовых полей-->
        <Style TargetType="{x:Type TextBox}">
            <Setter Property="Margin" Value="3"/>
            <Setter Property="Width" Value="170"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Style>
    </Window.Resources>
    <Grid Background="LightGray">
        <Grid.RowDefinitions>
            <RowDefinition Height="260"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="310"/>
            <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>
        <!--Панель со списком элементов-->
        <StackPanel Grid.Row="0" Grid.Column="0">
            <Label HorizontalContentAlignment="Center" Content="Список билетов"/>
            <ListBox Name="listBoxTickets" Height="232" Margin="17, 3, 0.4, 0"
                    BorderThickness="2"
                    ItemTemplate="{StaticResource TicketDataTemplate}"
                    ItemsSource="{Binding}"/>
        </StackPanel>
        <!--Панель с кнопками-->
        <StackPanel Grid.Row="1" Grid.Column="0" Margin="16.6, 0, 0, 0"
                    Background="LightGray">
            <Label HorizontalContentAlignment="Center" Content="Управление списком
билетов"/>
        </StackPanel>
    </Grid>
</Window>
```

```

        <StackPanel Orientation="Horizontal">
            <Button Name="addButton" Click="addButton_Click" Content="Добавить"
                Width="80" Background="White" BorderThickness="2"
                BorderBrush="Orange"/>
            <Button Name="delButton" Click="delButton_Click" Content="Удалить"
                Width="80" Background="White" BorderThickness="2"
                BorderBrush="Orange" Margin="27, 0" />
            <Button Name="saveButton" Click="saveButton_Click" Content="Сохранить"
                Width="80" Background="White" BorderThickness="2"
                BorderBrush="Orange"/>
        </StackPanel>
    </StackPanel>
    <!-- Панель с кодом XML-документа -->
    <StackPanel Grid.Row="0" Grid.Column="1" Grid.RowSpan="3">
        <Label HorizontalContentAlignment="Center" Content="Код XML-документа"/>
        <TextBox Name="textBoxXML" Width="350" Height="470"
            IsReadOnly="True" VerticalScrollBarVisibility="Auto"/>
    </StackPanel>
    <!-- Панель с текстовыми полями -->
    <StackPanel Grid.Row="2" Grid.Column="0" Margin="16.6, 0, 0, 0"
        DataContext="{Binding ElementName=listBoxTickets, Path=SelectedItem}">
        <Label HorizontalContentAlignment="Center" Content="Редактировать выбранный
            билет"/>
        <StackPanel Orientation="Horizontal">
            <Label Width="120">Код билета</Label>
            <TextBox Text="{Binding Path=Attribute[id].Value}"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal" Margin="0, 2">
            <Label Width="120">Рейс</Label>
            <TextBox Text="{Binding Path=Attribute[trip].Value}"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal">
            <Label Width="120">Пункт назначения</Label>
            <TextBox Text="{Binding Path=Element[point].Value}"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal" Margin="0, 2">
            <Label Width="120">Время отправления</Label>
            <TextBox Text="{Binding Path=Element[departure].Value}"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal">
            <Label Width="120">Длительность, мин</Label>
            <TextBox Text="{Binding Path=Element[duration].Value}"/>
        </StackPanel>
        <StackPanel Orientation="Horizontal" Margin="0, 2">
            <Label Width="120">Номер места</Label>
            <TextBox Text="{Binding Path=Element[number].Value}"/>
        </StackPanel>
    </StackPanel>
</Grid>
</Window>

```

Напишем код на C#:

```

using System.Windows;
using System.Xml.Linq;

namespace ЛР_4_1
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        // XML-документ с данными о товарах
    }
}

```



```

private XDocument xDoc;
public MainWindow()
{
    InitializeComponent();
}
// Выполнение действий при загрузке окна MainWindow
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    try
    {
        xDoc = XDocument.Load("ticket.xml");
        // Привязка XML-данных к списку listBoxTickets
        listBoxTickets.DataContext = xDoc.Descendants("ticketBus").Elements();
        textBoxXML.Text = xDoc.ToString();
    }
    catch (System.IO.FileNotFoundException ex)
    {
        // Вывод сообщения об ошибке при отсутствии XML-файла
        MessageBox.Show(ex.Message);
        return;
    }
}
// Добавление нового элемента ticket в XML-документ при нажатии addButton
private void addButton_Click(object sender, RoutedEventArgs e)
{
    // Элемент newTicket со значениями по умолчанию
    XElement newTicket = new XElement("ticket",
        new XAttribute("id", "Код билета"),
        new XAttribute("trip", "Рейс"),
        new XElement("point", "Пункт назначения"),
        new XElement("departure", "Время отправления"),
        new XElement("duration", "Длительность",
            new XAttribute("unit", "мин")),
        new XElement("number", "Номер места")
    );

    // Добавление элемента newTicket в начало списка элементов ticket
    xDoc.Element("ticketBus").Element("ticket").AddBeforeSelf(newTicket);
    // Обновление списка listBoxTickets и текстового поля textBoxXML
    listBoxTickets.DataContext = xDoc.Descendants("ticketBus").Elements();
    textBoxXML.Text = xDoc.ToString();
    // Сохранение XML-документа
    xDoc.Save("ticket.xml");
}
// Удаление выделенного элемента ticket из XML-документа при нажатии delButton
private void delButton_Click(object sender, RoutedEventArgs e)
{
    XElement delTicket = (XElement)listBoxTickets.SelectedItem;
    delTicket.Remove();
    listBoxTickets.DataContext = xDoc.Descendants("ticketBus").Elements();
    textBoxXML.Text = xDoc.ToString();
    xDoc.Save("ticket.xml");
}
// Принудительное сохранение данных с обновлением отображаемого кода XML
private void saveButton_Click(object sender, RoutedEventArgs e)
{
    textBoxXML.Text = xDoc.ToString();
    xDoc.Save("ticket.xml");
}
}
}

```

1.3 Результат работы приложения

Окно работающего приложения WPF с привязкой данных показано на рисунке 1.1.

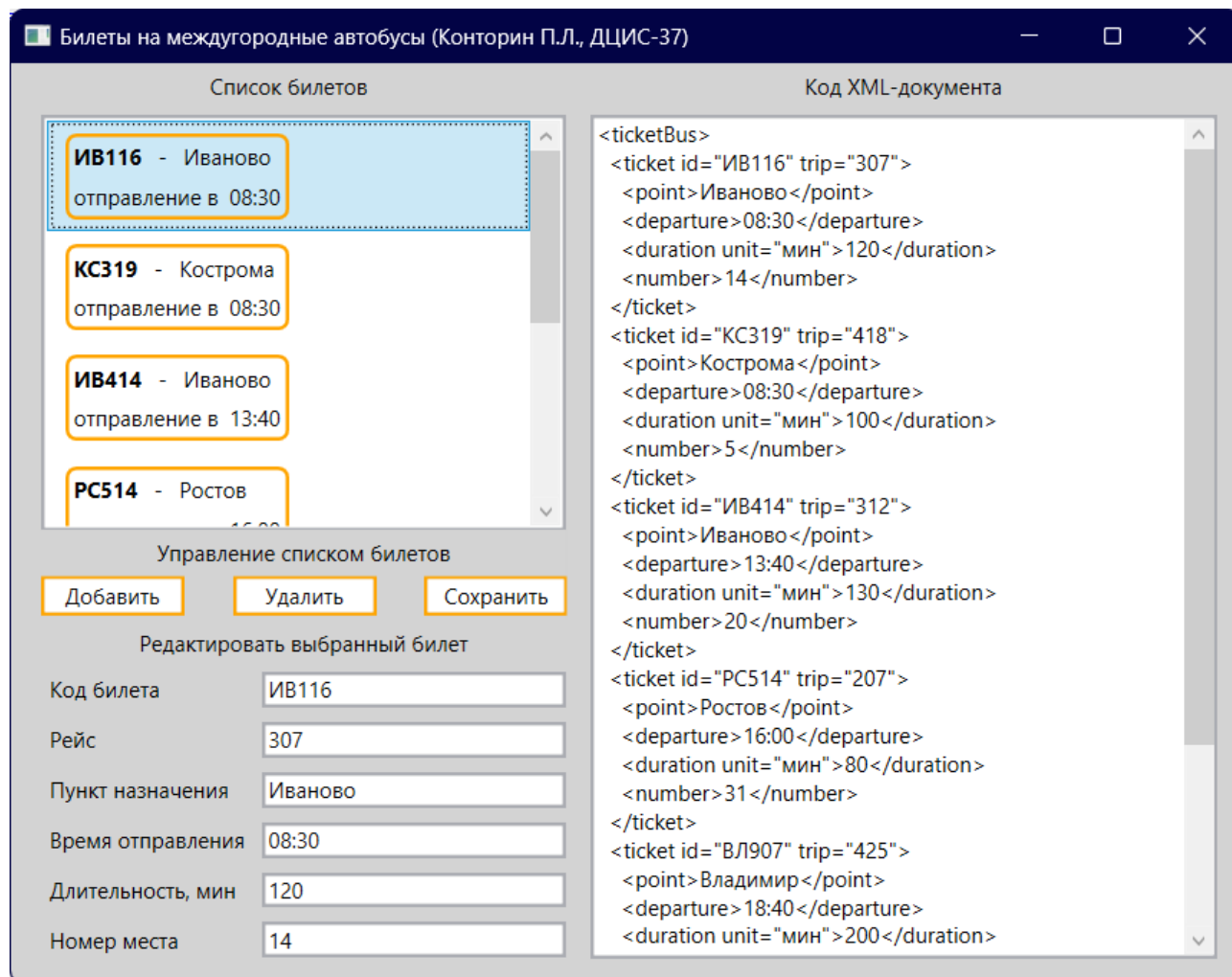


Рисунок 1.1 – Рабочее окно приложения WPF

2 Шаблоны данных

2.1 Задание

Требуется доработать приложение со страничной навигацией из лабораторной работы №3, добавив привязку таблиц (элементы DataGrid) и их отдельных столбцов к данным из XML-документа. При этом данные должны выводиться в табличной форме при загрузке соответствующих страниц приложения.

2.2 Код программы

В качестве XML-документа воспользуемся файлом bank.xml, полученным в лабораторной работе №2:

```
<?xml version="1.0" encoding="utf-8"?>
<банковские_услуги>
  <банк бик="044525225" название="Сбербанк">
    <вклад код="с6137">
      <вид>срочный</вид>
      <процент ед_изм="%">11,6</процент>
      <вкладчик инн="374981357822">
        <фio>Воробьев Сергей Борисович</фio>
        <сумма ед_изм="руб.">150000,00</сумма>
      </вкладчик>
    </вклад>
    <вкладчик инн="761594352278">
      <фio>Прокопьев Антон Сергеевич</фio>
      <сумма ед_изм="руб.">275000,00</сумма>
    </вкладчик>
    <вкладчик инн="762248659912">
      <фio>Бутенко Алексей Викторович</фio>
      <сумма ед_изм="руб.">680000,00</сумма>
    </вкладчик>
  </банк>
  <вклад код="с6265">
    <вид>накопительный</вид>
    <процент ед_изм="%">15,3</процент>
    <вкладчик инн="117856842341">
      <фio>Копылов Семен Андреевич</фio>
      <сумма ед_изм="руб.">1250000,00</сумма>
    </вкладчик>
  </вклад>
  <вклад код="с6318">
    <вид>социальный</вид>
    <процент ед_изм="%">1,95</процент>
    <вкладчик инн="374981357822">
      <фio>Воробьев Сергей Борисович</фio>
      <сумма ед_изм="руб.">150000,00</сумма>
    </вкладчик>
    <вкладчик инн="762248659912">
      <фio>Бутенко Алексей Викторович</фio>
```

```

        <сумма_ед_изм="руб.">680000,00</сумма>
    </вкладчик>
</вклад>
</банк>
<банк бик="044525974" название="Тинькофф">
    <вклад код="тн175">
        <вид>до востребования</вид>
        <процент_ед_изм="%">5,16</процент>
        <вкладчик_инн="374981357822">
            <фио>Воробьев Сергей Борисович</фио>
            <сумма_ед_изм="руб.">150000,00</сумма>
        </вкладчик>
        <вкладчик_инн="761594352278">
            <фио>Прокопьев Антон Сергеевич</фио>
            <сумма_ед_изм="руб.">275000,00</сумма>
        </вкладчик>
    </вклад>
    <вклад код="тн261">
        <вид>сберегательный</вид>
        <процент>8,19</процент>
        <вкладчик_инн="762248659912">
            <фио>Бутенко Алексей Викторович</фио>
            <сумма_ед_изм="руб.">680000,00</сумма>
        </вкладчик>
    </вклад>
</банк>
</банковские_услуги>

```

Напишем код присоединенного модуля на C# для главного окна:

```

using System.Windows;
using System.Xml.Linq;

namespace ЛР_4_2
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    // Класс для определения набора статических методов, реализующих
    // логику LINQ to XML
    class XMLDocBanks
    {
        // Метод, возвращающий XDocument, заполненный на основе
        // содержимого файла "bank.xml"
        public static XDocument GetXDocument()
        {
            try
            {
                XDocument xDoc = XDocument.Load("bank.xml");
                return xDoc;
            }
            catch (System.IO.FileNotFoundException ex)
            {
                // Вывод сообщения об ошибке при отсутствии XML-файла
                MessageBox.Show(ex.Message);
                return null;
            }
        }
    }

    public partial class MainWindow : Window
    {
        public MainWindow()
        {

```

```

        InitializeComponent();
    }
}

```

Напишем код XAML-документа BanksPage.xaml (страница Банки):

```

<Page x:Class="LP_4_2.BanksPage"
    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="350" d:DesignWidth="500"
    Title="Страница банки" Loaded="Page_Load">
    <Page.Resources>
        <Style TargetType="{x:Type DataGridColumnHeader}">
            <Setter Property="Background" Value="Orange" />
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
        </Style>
    </Page.Resources>
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="20"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
        <!--Панель с главным меню приложения-->
        <StackPanel Grid.Row="0">
            <Menu/>
        </StackPanel>
        <!--Панель с заголовком страницы-->
        <StackPanel Grid.Row="1">
            <Label>Банки</Label>
        </StackPanel>
        <!--Панель с таблицей-->
        <StackPanel Grid.Row="2">
            <DataGrid x:Name="BanksDataGrid" AutoGenerateColumns="False">
                <DataGrid.Columns>
                    <DataGridTextColumn Header="БИК"
                        Binding="{Binding Path=Attribute[бик].Value}" Width="250"/>
                    <DataGridTextColumn Header="Название"
                        Binding="{Binding Path=Attribute[название].Value}" Width="*/>
                </DataGrid.Columns>
                <!--Описание шаблона данных для деталей строки-->
                <DataGrid.RowDetailsTemplate>
                    <DataTemplate>
                        <Border BorderThickness="4" BorderBrush="Orange" Padding="10">
                            <TextBlock>
                                <Hyperlink NavigateUri="DepositsPage.xaml">Перейти к списку вкладов</Hyperlink>
                            </TextBlock>
                        </Border>
                    </DataTemplate>
                </DataGrid.RowDetailsTemplate>
            </DataGrid>
        </StackPanel>
        <!--Панель со строкой состояния-->
        <StackPanel Grid.Row="3">
            <StatusBar/>
        </StackPanel>
    </Grid>

```

</Page>

Напишем код присоединенного модуля на C# для страницы Банки:

```
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Xml.Linq;

namespace ЛР_4_2
{
    /// <summary>
    /// Логика взаимодействия для BanksPage.xaml
    /// </summary>
    public partial class BanksPage : Page
    {
        public BanksPage()
        {
            InitializeComponent();
        }
        // Выполнение различных действий при загрузке страницы
        private void Page_Load(object sender, RoutedEventArgs e)
        {
            XDocument xDoc = XMLDocBanks.GetXDocument();
            // Запрос LINQ с получением данных о банках
            var banks = from b in xDoc.Descendants("банк") select b;
            // Привязка результатов запроса LINQ к элементу BanksDataGrid
            BanksDataGrid.ItemsSource = banks.ToList();
        }
    }
}
```

Напишем код XAML-документа DepositsPage.xaml (страница Вклады):

```
<Page x:Class="ЛР_4_2.DepositsPage"
      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="350" d:DesignWidth="500"
      Title="Страница вклады" Loaded="Page_Load">
    <Page.Resources>
        <Style TargetType="{x:Type DataGridColumnHeader}">
            <Setter Property="Background" Value="Orange" />
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
        </Style>
    </Page.Resources>
    <Grid>
        <!--Описание строк и столбцов панели Grid-->
        <Grid.RowDefinitions>
            <RowDefinition Height="20"/>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
        <!--Панель с главным меню приложения-->
        <StackPanel Grid.Row="0">
            <Menu/>
        </StackPanel>
        <!--Панель с заголовком страницы-->
        <StackPanel Grid.Row="1">
```

```

        <Label>Вклады</Label>
    </StackPanel>
    <!--Панель с таблицей-->
    <StackPanel Grid.Row="2">
        <DataGrid Name="BanksDataGrid" AutoGenerateColumns="False">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Код"
                    Binding="{Binding Path=Attribute[код].Value}" Width="166"/>
                <DataGridTextColumn Header="Вид"
                    Binding="{Binding Path=Element[вид].Value}" Width="166"/>
                <DataGridTextColumn Header="Процент"
                    Binding="{Binding Path=Element[процент].Value}" Width="*/>
            </DataGrid.Columns>
            <!--Описание шаблона данных для деталей строки-->
            <DataGrid.RowDetailsTemplate>
                <DataTemplate>
                    <Border BorderThickness="4" BorderBrush="Orange" Padding="10">
                        <TextBlock>
                            <Hyperlink NavigateUri="DepositorsPage.xaml">Перейти к
                                списку вкладчиков</Hyperlink>
                        </TextBlock>
                    </Border>
                </DataTemplate>
            </DataGrid.RowDetailsTemplate>
        </DataGrid>
    </StackPanel>
    <!--Панель со строкой состояния-->
    <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
        <StatusBar/>
    </StackPanel>
</Grid>
</Page>

```

Напишем код присоединенного модуля на C# для страницы Вклады:

```

using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Xml.Linq;

namespace ЛР_4_2
{
    /// <summary>
    /// Логика взаимодействия для DepositsPage.xaml
    /// </summary>
    public partial class DepositsPage : Page
    {
        public DepositsPage()
        {
            InitializeComponent();
        }
        private void Page_Load(object sender, RoutedEventArgs e)
        {
            XDocument xDoc = XMLDocBanks.GetXDocument();
            // Запрос LINQ с получением данных о банках
            var banks = from b in xDoc.Descendants("вклад") select b;
            // Привязка результатов запроса LINQ к элементу BanksDataGrid
            BanksDataGrid.ItemsSource = banks.ToList();
        }
    }
}

```

Напишем код XAML-документа DepositorsPage.xaml (страница Вкладчики):

```
<Page x:Class="ЛР_4_2.DepositorsPage"
      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"
      xmlns:local="clr-namespace:ЛР_4_2"
      mc:Ignorable="d"
      d:DesignHeight="350" d:DesignWidth="500"
      Title="Страница вкладчики" Loaded="Page_Load">
  <Page.Resources>
    <Style TargetType="{x:Type DataGridColumnHeader}">
      <Setter Property="Background" Value="Orange" />
      <Setter Property="HorizontalContentAlignment" Value="Center"/>
    </Style>
  </Page.Resources>
  <Grid>
    <!--Описание строк и столбцов панели Grid-->
    <Grid.RowDefinitions>
      <RowDefinition Height="20"/>
      <RowDefinition Height="50"/>
      <RowDefinition Height="*/>
      <RowDefinition Height="20"/>
    </Grid.RowDefinitions>
    <!--Панель с главным меню приложения-->
    <StackPanel Grid.Row="0">
      <Menu/>
    </StackPanel>
    <!--Панель с заголовком страницы-->
    <StackPanel Grid.Row="1">
      <Label>Вкладчики</Label>
    </StackPanel>
    <!--Панель с таблицей-->
    <StackPanel Grid.Row="2">
      <DataGrid Name="BanksDataGrid" AutoGenerateColumns="False">
        <DataGrid.Columns>
          <DataGridTextColumn Header="ИНН"
            Binding="{Binding Path=Attribute[инн].Value}" Width="166"/>
          <DataGridTextColumn Header="ФИО"
            Binding="{Binding Path=Element[фio].Value}" Width="166"/>
          <DataGridTextColumn Header="Сумма"
            Binding="{Binding Path=Element[сумма].Value}" Width="*/>
        </DataGrid.Columns>
      </DataGrid>
    </StackPanel>
    <!--Панель со строкой состояния-->
    <StackPanel Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
      <StatusBar/>
    </StackPanel>
  </Grid>
</Page>
```

Напишем код присоединенного модуля на C# для страницы Вкладчики:

```
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Xml.Linq;

namespace ЛР_4_2
```



```

{
    /// <summary>
    /// Логика взаимодействия для DepositorsPage.xaml
    /// </summary>
    public partial class DepositorsPage : Page
    {
        public DepositorsPage()
        {
            InitializeComponent();
        }
        private void Page_Load(object sender, RoutedEventArgs e)
        {
            XDocument xDoc = XMLDocBanks.GetXDocument();
            // Запрос LINQ с получением данных о банках
            var banks = from b in xDoc.Descendants("вкладчик") select b;
            // Привязка результатов запроса LINQ к элементу BanksDataGrid
            BanksDataGrid.ItemsSource = banks.ToList();
        }
    }
}

```

2.3 Результат работы приложения

Окно работающего приложения WPF с загруженными в него страницами показано на рисунках 2.1, 2.2 и 2.3.

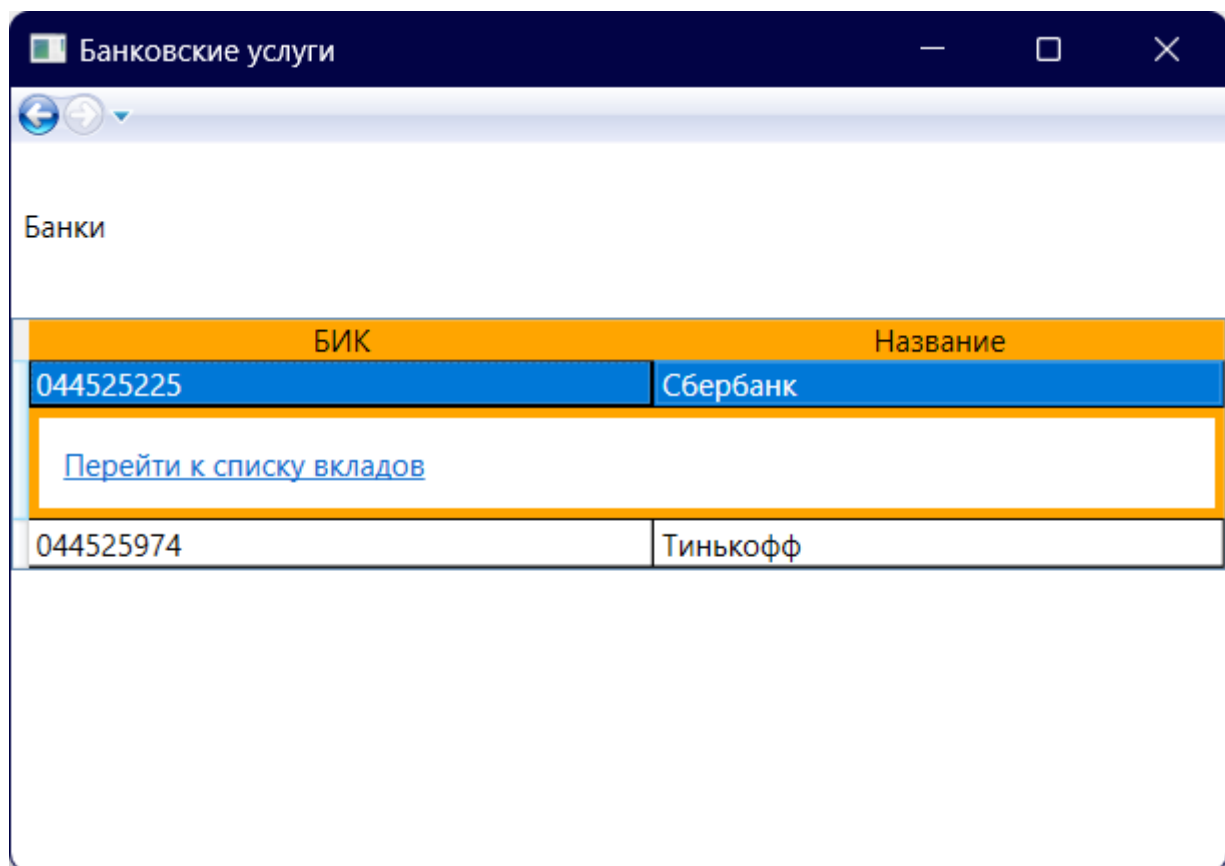


Рисунок 2.1 – Окно приложения с загруженной страницей «Банки»



Рисунок 2.2 – Окно приложения с загруженной страницей «Вклады»



Рисунок 2.3 – Окно приложения с загруженной страницей «Вкладчики»