**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ**

**О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**

**ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ.01 Разработка программных модулей для программного обеспечения для компьютерных систем**

по специальности

09.02.03 Программирование в компьютерных системах

Студент: Берникова В.С.

Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2020-2021

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 1**

**Тема проекта: Работа с WPF приложением**

Студент: Берникова В.С.

Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

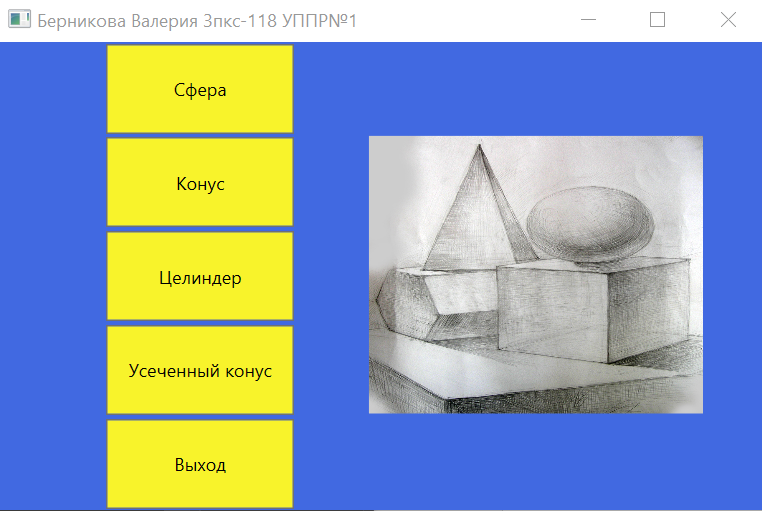
Оценка за проект: 5

Дата: 30.10.2020

**Цель работы**: Требуется разработать проект, в котором производятся вычисления основных характеристик ряда стереометрических фигур - сферы, конуса, усеченного конуса и цилиндра.

**Ниже на рисунке представлен фрагмент работающего проекта**

**Меню**



**Код WPF:**

<Window x:Class="up\_pr1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Берникова Валерия 3пкс-118 УППР№1" Height="350" Width="525" Background="#4169E1">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="6\*"/>

</Grid.ColumnDefinitions>

<Button Name="Sfera" Content="Сфера" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Left" Height="60" Margin="0,1,0,0" VerticalAlignment="Top" Width="125" Click="Sfera\_Click" Background="#F8F32B"/>

<Button Name="Konus" Content="Конус" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Left" Height="60" Margin="0,1,0,0" VerticalAlignment="Top" Width="125" Click="Konus\_Click" Background="#F8F32B"/>

<Button Name="Celinder" Content="Целиндер" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Left" Height="60" Margin="0,1,0,0" VerticalAlignment="Top" Width="125" Click="Celinder\_Click" Background="#F8F32B"/>

<Button Name="YsKonuc" Content="Усеченный конус" Grid.Column="1" Grid.Row="3" HorizontalAlignment="Left" Height="60" Margin="0,1,0,0" VerticalAlignment="Top" Width="125" Click="YsKonus\_Click" Background="#F8F32B"/>

<Button Name="Exet" Content="Выход" Grid.Column="1" Grid.Row="4" HorizontalAlignment="Left" Height="60" Margin="0,1,0,0" VerticalAlignment="Top" Width="125" Click="Exet\_Click" Background="#F8F32B"/>

<Image Grid.Column="1" Margin="175,0,40,65" Grid.Row="1" Grid.RowSpan="4" Source="01.jpg" Stretch="Fill"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

namespace up\_pr1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Sfera\_Click(object sender, RoutedEventArgs e)

{

Window\_Sfera window2 = new Window\_Sfera();

window2.ShowDialog();

}

private void Konus\_Click(object sender, RoutedEventArgs e)

{

Window\_Konus window3 = new Window\_Konus();

window3.ShowDialog();

}

private void Celinder\_Click(object sender, RoutedEventArgs e)

{

Window\_Celinder window4 = new Window\_Celinder();

window4.ShowDialog();

}

private void YsKonus\_Click(object sender, RoutedEventArgs e)

{

Window\_YsKonus window5 = new Window\_YsKonus();

window5.ShowDialog();

}

private void Exet\_Click(object sender, RoutedEventArgs e)

{

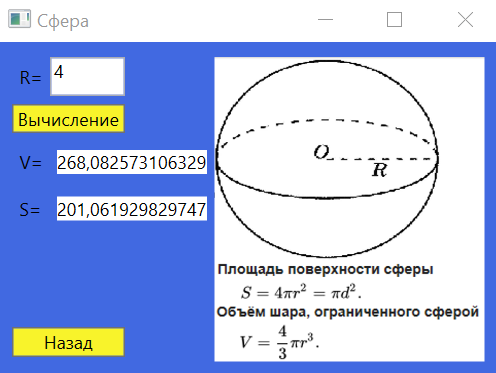
this.Close();

}

}

}

**Сфера**

****

**Код WPF:**

<Window x:Class="up\_pr1.Window\_Sfera"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Сфера" Height="260" Width="350" Background="#4169E1">

<Grid>

<Label Content="R=" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<TextBox Name="TextBoxR" HorizontalAlignment="Left" Height="26" Margin="35,10,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="50"/>

<Button Name="Chet" Content="Вычисление" HorizontalAlignment="Left" Margin="10,41,0,0" VerticalAlignment="Top" Width="75" Click="Chet\_Click" Background="#F8F32B"/>

<Label Content="V=" HorizontalAlignment="Left" Margin="10,67,0,0" VerticalAlignment="Top"/>

<TextBlock Name="TextBlockV" HorizontalAlignment="Left" Margin="40,72,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFFFFF" />

<Label Content="S=" HorizontalAlignment="Left" Margin="10,98,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.514,1.709"/>

<TextBlock Name="TextBlockS" HorizontalAlignment="Left" Margin="40,103,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFFFFF"/>

<Image Margin="145,10,10.333,9.667" Source="Image1.png" Stretch="Fill"/>

<Button Name="nazad" Content="Назад" HorizontalAlignment="Left" Margin="10,190,0,0" VerticalAlignment="Top" Width="75" Click="nazad\_Click" Background="#F8F32B"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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 up\_pr1

{

/// <summary>

/// Логика взаимодействия для Window\_Sfera.xaml

/// </summary>

public partial class Window\_Sfera : Window

{

public Window\_Sfera()

{

InitializeComponent();

}

private void Chet\_Click(object sender, RoutedEventArgs e)

{

double R = Convert.ToDouble(TextBoxR.Text);

var otvet = ClassRechnie.Sfera(R);

TextBlockV.Text = Convert.ToString(otvet.Item1);

TextBlockS.Text = Convert.ToString(otvet.Item2);

}

private void nazad\_Click(object sender, RoutedEventArgs e)

{

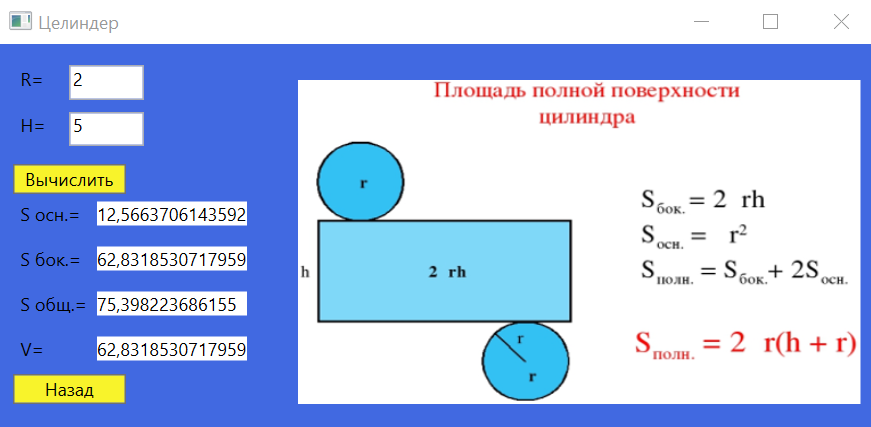
this.Close();

}

}

}

**Цилиндре**

****

**Код WPF:**

<Window x:Class="up\_pr1.Window\_Celinder"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Целиндер" Height="300" Width="600" Background="#4169E1">

<Grid>

<Label Content="R=" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<Label Content="H=" HorizontalAlignment="Left" Margin="10,41,0,0" VerticalAlignment="Top"/>

<TextBox Name="TextBoxR1" HorizontalAlignment="Left" Height="23" Margin="47,14,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="50"/>

<TextBox Name="TextBoxH" HorizontalAlignment="Left" Height="23" Margin="47,45,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="50"/>

<Button Name="Chet" Content="Вычислить" HorizontalAlignment="Left" Margin="10,80,0,0" VerticalAlignment="Top" Width="75" Click="Chet\_Click" Background="#F8F32B"/>

<Label Content="S осн.=" HorizontalAlignment="Left" Margin="10,100,0,0" VerticalAlignment="Top"/>

<Label Content="S бок.=" HorizontalAlignment="Left" Margin="10,130,0,0" VerticalAlignment="Top"/>

<Label Content="S общ.=" HorizontalAlignment="Left" Margin="10,160,0,0" VerticalAlignment="Top"/>

<Label Content="V=" HorizontalAlignment="Left" Margin="10,190,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.514,2.002"/>

<TextBlock Name="TextBlockSocn" HorizontalAlignment="Left" Margin="66,105,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Background="#FFFF" Width="100"/>

<TextBlock Name="TextBlockSboc" HorizontalAlignment="Left" Margin="66,135,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Background="#FFFF" Width="100"/>

<TextBlock Name="TextBlockSob" HorizontalAlignment="Left" Margin="66,165,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" RenderTransformOrigin="0.401,2.27" Background="#FFFF" Width="100"/>

<TextBlock Name="TextBlockV" HorizontalAlignment="Left" Margin="66,195,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Background="#FFFF" Width="100"/>

<Image Margin="200,24,10.333,22.667" Source="Image3.png" Stretch="Fill"/>

<Button Name="nazad" Content="Назад" HorizontalAlignment="Left" Margin="10,220,0,0" VerticalAlignment="Top" Width="75" Click="nazad\_Click" Background="#F8F32B"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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 up\_pr1

{

/// <summary>

/// Логика взаимодействия для Window\_Celinder.xaml

/// </summary>

public partial class Window\_Celinder : Window

{

public Window\_Celinder()

{

InitializeComponent();

}

private void Chet\_Click(object sender, RoutedEventArgs e)

{

double R = Convert.ToDouble(TextBoxR1.Text);

double H = Convert.ToDouble(TextBoxH.Text);

var otvet = ClassRechnie.Celinder(R, H);

TextBlockSocn.Text = Convert.ToString(otvet.Item1);

TextBlockSboc.Text = Convert.ToString(otvet.Item2);

TextBlockSob.Text = Convert.ToString(otvet.Item3);

TextBlockV.Text = Convert.ToString(otvet.Item4);

}

private void nazad\_Click(object sender, RoutedEventArgs e)

{

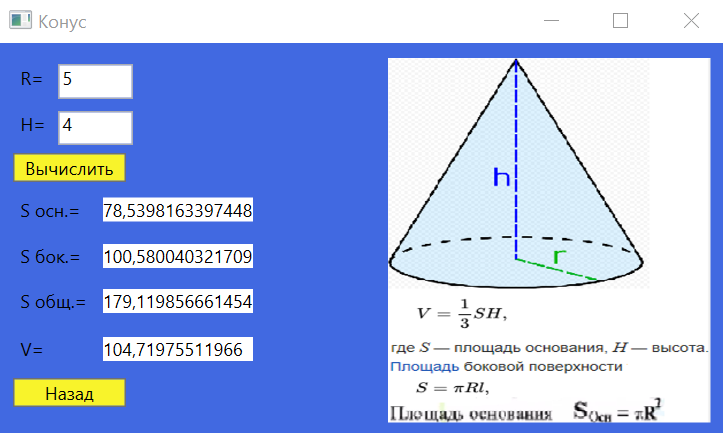
this.Close();

}

}

}

**Конус**

****

**Код WPF:**

<Window x:Class="up\_pr1.Window\_Konus"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Конус" Height="300" Width="500" Background="#4169E1">

<Grid>

<Label Content="R=" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<Label Content="H=" HorizontalAlignment="Left" Margin="10,41,0,0" VerticalAlignment="Top"/>

<TextBox Name="TextBoxR" HorizontalAlignment="Left" Height="23" Margin="40,14,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="50"/>

<TextBox Name="TextBoxH" HorizontalAlignment="Left" Height="23" Margin="40,45,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="50"/>

<Button Name="Chet" Content="Вычислить" HorizontalAlignment="Left" Margin="10,73,0,0" VerticalAlignment="Top" Width="75" Click="Chet\_Click" Background="#F8F32B"/>

<Label Content="S осн.=" HorizontalAlignment="Left" Margin="10,98,0,0" VerticalAlignment="Top"/>

<Label Content="S бок.=" HorizontalAlignment="Left" Margin="10,129,0,0" VerticalAlignment="Top"/>

<Label Content="S общ.=" HorizontalAlignment="Left" Margin="10,159,0,0" VerticalAlignment="Top"/>

<Label Content="V=" HorizontalAlignment="Left" Margin="10,191,0,0" VerticalAlignment="Top"/>

<TextBlock Name="TextBlockSo" HorizontalAlignment="Left" Margin="70,103,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockSb" HorizontalAlignment="Left" Margin="70,134,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockSob" HorizontalAlignment="Left" Margin="70,164,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockV" HorizontalAlignment="Left" Margin="70,196,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<Button Name="nazad" Content="Назад" HorizontalAlignment="Left" Margin="10,223,0,0" VerticalAlignment="Top" Width="75" Click="nazad\_Click" Background="#F8F32B"/>

<Image Margin="260,10,10.333,9.667" Source="Image4.png" Stretch="Fill"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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 up\_pr1

{

/// <summary>

/// Логика взаимодействия для Window\_Konus.xaml

/// </summary>

public partial class Window\_Konus : Window

{

public Window\_Konus()

{

InitializeComponent();

}

private void Chet\_Click(object sender, RoutedEventArgs e)

{

double R = Convert.ToDouble(TextBoxR.Text);

double H = Convert.ToDouble(TextBoxH.Text);

var otvet = ClassRechnie.Konus(R, H);

TextBlockSo.Text = Convert.ToString(otvet.Item1);

TextBlockSb.Text = Convert.ToString(otvet.Item2);

TextBlockSob.Text = Convert.ToString(otvet.Item3);

TextBlockV.Text = Convert.ToString(otvet.Item4);

}

private void nazad\_Click(object sender, RoutedEventArgs e)

{

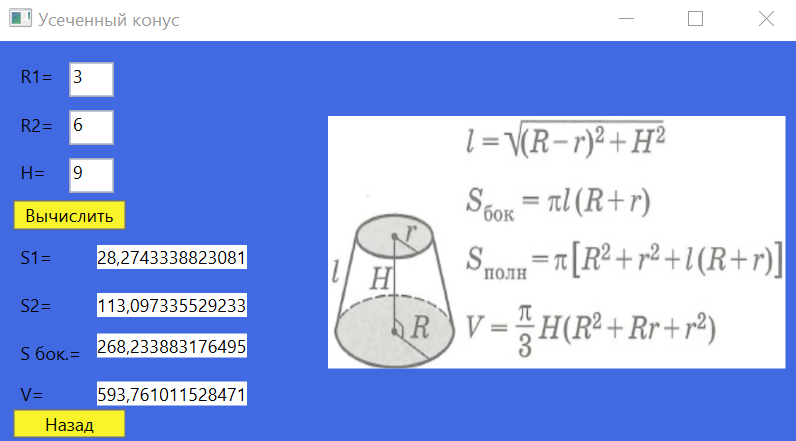
this.Close();

}

}

}

**Усеченный конус**

****

**Код WPF:**

<Window x:Class="up\_pr1.Window\_YsKonus"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Усеченный конус" Height="305" Width="550" Background="#4169E1">

<Grid>

<Label Content="R1=" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<Label Content="R2=" HorizontalAlignment="Left" Margin="10,43,0,0" VerticalAlignment="Top"/>

<Label Content="H=" HorizontalAlignment="Left" Margin="10,74,0,0" VerticalAlignment="Top"/>

<TextBox Name="TextBoxR1" HorizontalAlignment="Left" Height="23" Margin="47,14,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="30"/>

<TextBox Name="TextBoxR2" HorizontalAlignment="Left" Height="23" Margin="47,46,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="30"/>

<TextBox Name="TextBoxH" HorizontalAlignment="Left" Height="23" Margin="47,78,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="30"/>

<Button Name="Chet" Content="Вычислить" HorizontalAlignment="Left" Margin="10,106,0,0" VerticalAlignment="Top" Width="75" Click="Chet\_Click" Background="#F8F32B"/>

<Label Content="S1=" HorizontalAlignment="Left" Margin="10,131,0,0" VerticalAlignment="Top"/>

<Label Content="S2=" HorizontalAlignment="Left" Margin="10,163,0,0" VerticalAlignment="Top"/>

<Label Content="S бок.=" HorizontalAlignment="Left" Margin="10,195,0,0" VerticalAlignment="Top"/>

<Label Content="V=" HorizontalAlignment="Left" Margin="10,222,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.514,2.002"/>

<TextBlock Name="TextBlockS1" HorizontalAlignment="Left" Margin="66,136,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockS2" HorizontalAlignment="Left" Margin="66,168,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockSboc" HorizontalAlignment="Left" Margin="66,195,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<TextBlock Name="TextBlockV" HorizontalAlignment="Left" Margin="66,227,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100" Background="#FFF"/>

<Image Margin="220,50,10.333,49.667" Source="Image5.png" Stretch="Fill"/>

<Button Name="nazad" Content="Назад" HorizontalAlignment="Left" Margin="10,245,0,0" VerticalAlignment="Top" Width="75" Click="nazad\_Click" Background="#F8F32B"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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 up\_pr1

{

/// <summary>

/// Логика взаимодействия для Window\_YsKonus.xaml

/// </summary>

public partial class Window\_YsKonus : Window

{

public Window\_YsKonus()

{

InitializeComponent();

}

private void Chet\_Click(object sender, RoutedEventArgs e)

{

double R1 = Convert.ToDouble(TextBoxR1.Text);

double R2 = Convert.ToDouble(TextBoxR2.Text);

double H = Convert.ToDouble(TextBoxH.Text);

var otvet = ClassRechnie.YsKonus(R1, R2, H);

TextBlockS1.Text = Convert.ToString(otvet.Item1);

TextBlockS2.Text = Convert.ToString(otvet.Item2);

TextBlockSboc.Text = Convert.ToString(otvet.Item3);

TextBlockV.Text = Convert.ToString(otvet.Item4);

}

private void nazad\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

Код решения Class:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace up\_pr1

{

class ClassRechnie

{

public static Tuple<double, double> Sfera(double R)

{

double V = (4 \* Math.PI \* Math.Pow(R, 3)) / 3;

double S = 4 \* Math.PI \* Math.Pow(R, 2);

return new Tuple<double, double>(V, S);

}

public static Tuple<double, double, double, double> Konus(double R, double H)

{

double So = Math.PI \* Math.Pow(R, 2);

double Sb = Math.PI \* R \* Math.Sqrt(Math.Pow(R, 2) + Math.Pow(H, 2));

double Sob = So + Sb;

double V = (So \* H) / 3;

return new Tuple<double, double, double, double>(So, Sb, Sob, V);

}

public static Tuple<double, double, double, double> YsKonus(double R1, double R2, double H)

{

double S1 = Math.PI \* Math.Pow(R1, 2);

double S2 = Math.PI \* Math.Pow(R2, 2);

double Sboc = Math.PI \* Math.Sqrt(Math.Pow((R2 - R1), 2) + Math.Pow(H, 2)) \* (R2 + R1);

double V = (Math.PI / 3) \* H \* (Math.Pow(R2, 2) + R2 \* R1 + Math.Pow(R1, 2));

return new Tuple<double, double, double, double>(S1, S2, Sboc, V);

}

public static Tuple<double, double, double, double> Celinder(double R, double H)

{

double Socn = Math.PI \* Math.Pow(R, 2);

double Sboc = 2 \* Math.PI \* R \* H;

double Sob = Socn + Sboc;

double V = Math.PI \* Math.Pow(R, 2) \* H;

return new Tuple<double, double, double, double>(Socn, Sboc, Sob, V);

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 2**

**Тема проекта: «Числовой массив» - выполнение проекта с использованием классов.**

Студент: Берникова В.С.

Группа: 3ПКС-118

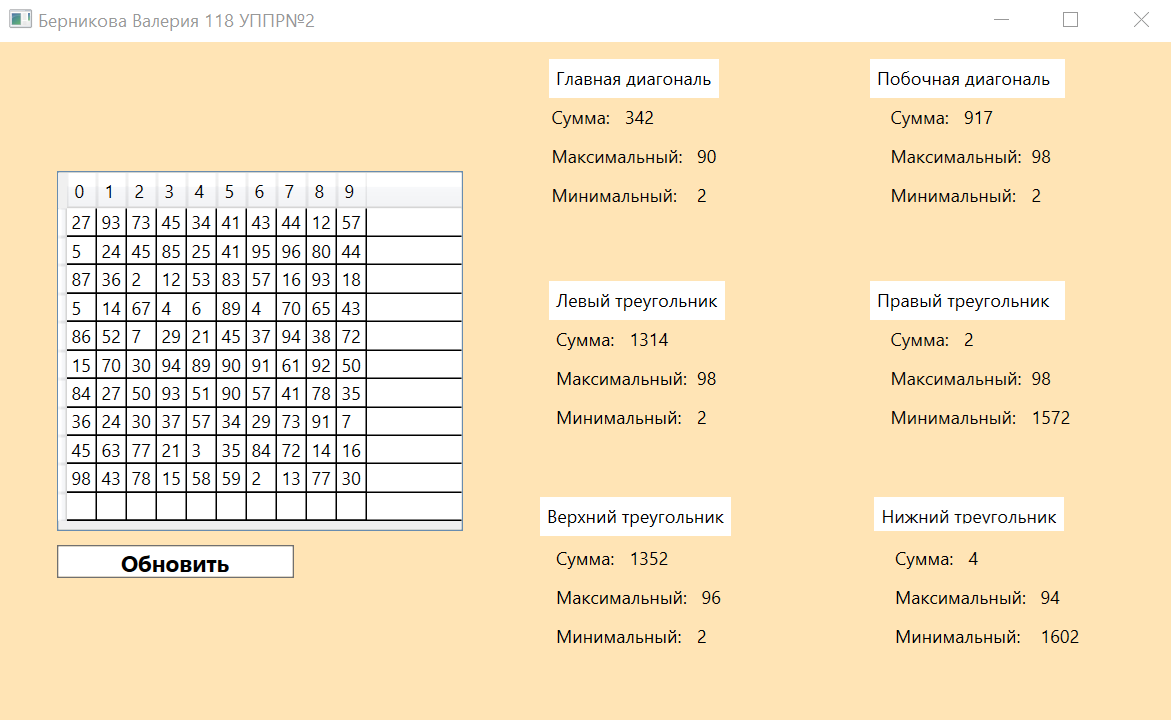
Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 06.11.2020

**Цель работы**: Изучить теоретический материал и выполнить консольный проект, демонстрирующий работу с делегатами.

**Ниже на рисунке представлен фрагмент работающего проекта**

****

**Код WPF:**

<Window x:Class="up\_pr2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Берникова Валерия 118 УППР№2" Height="500" Width="800" Background="Moccasin">

<Grid>

<DataGrid Name="grid" ItemsSource="{Binding}" HorizontalAlignment="Left" Height="240" Margin="39,86,0,0" VerticalAlignment="Top" Width="271"/>

<Button Name="newdata" Content="Обновить" HorizontalAlignment="Left" Margin="39,335,0,0" VerticalAlignment="Top" Width="158" Click="Hh" Height="22" FontSize="15" FontWeight="Bold" Background="#FFFFFF"/>

<Label Content="Главная диагональ" HorizontalAlignment="Left" Margin="367,11,0,0" VerticalAlignment="Top" Height="26" Background="White"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="364,37,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="364,63,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="364,89,0,0" VerticalAlignment="Top"/>

<Label Name="rt1" Content="" HorizontalAlignment="Left" Margin="413,37,0,0" VerticalAlignment="Top"/>

<Label Name="rt2" Content="" HorizontalAlignment="Left" Margin="461,63,0,0" VerticalAlignment="Top"/>

<Label Name="rt3" Content="" HorizontalAlignment="Left" Margin="461,89,0,0" VerticalAlignment="Top"/>

<Label Content="Побочная диагональ" HorizontalAlignment="Left" Margin="581,11,0,0" VerticalAlignment="Top" Height="26" Background="White" Width="130"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="590,37,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="590,63,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="590,89,0,0" VerticalAlignment="Top"/>

<Label Name="rt4" Content="" HorizontalAlignment="Left" Margin="639,37,0,0" VerticalAlignment="Top"/>

<Label Name="rt5" Content="" HorizontalAlignment="Left" Margin="684,63,0,0" VerticalAlignment="Top"/>

<Label Name="rt6" Content="" HorizontalAlignment="Left" Margin="684,89,0,0" VerticalAlignment="Top"/>

<Label Content="Левый треугольник" HorizontalAlignment="Left" Margin="367,159,0,0" VerticalAlignment="Top" Height="26" Background="White"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="367,185,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="367,211,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="367,237,0,0" VerticalAlignment="Top"/>

<Label Name="rt7" Content="" HorizontalAlignment="Left" Margin="416,185,0,0" VerticalAlignment="Top"/>

<Label Name="rt8" Content="" HorizontalAlignment="Left" Margin="461,211,0,0" VerticalAlignment="Top"/>

<Label Name="rt9" Content="" HorizontalAlignment="Left" Margin="461,237,0,0" VerticalAlignment="Top"/>

<Label Content="Верхний треугольник" HorizontalAlignment="Left" Margin="361,303,0,0" VerticalAlignment="Top" Height="26" Background="White"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="367,331,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="367,357,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="367,383,0,0" VerticalAlignment="Top"/>

<Label Name="rt10" Content="" HorizontalAlignment="Left" Margin="416,331,0,0" VerticalAlignment="Top"/>

<Label Name="rt11" Content="" HorizontalAlignment="Left" Margin="464,357,0,0" VerticalAlignment="Top"/>

<Label Name="rt12" Content="" HorizontalAlignment="Left" Margin="461,383,0,0" VerticalAlignment="Top"/>

<Label Content="Нижний треугольник" HorizontalAlignment="Left" Margin="584,303,0,0" VerticalAlignment="Top" Height="23" Background="White"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="593,331,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="593,357,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="593,383,0,0" VerticalAlignment="Top"/>

<Label Name="rt13" Content="" HorizontalAlignment="Left" Margin="690,383,0,0" VerticalAlignment="Top"/>

<Label Name="rt14" Content="" HorizontalAlignment="Left" Margin="690,357,0,0" VerticalAlignment="Top"/>

<Label Name="rt15" Content="" HorizontalAlignment="Left" Margin="642,331,0,0" VerticalAlignment="Top"/>

<Label Content="Правый треугольник" HorizontalAlignment="Left" Margin="581,159,0,0" VerticalAlignment="Top" Height="26" Background="White" Width="130"/>

<Label Content="Сумма:" HorizontalAlignment="Left" Margin="590,185,0,0" VerticalAlignment="Top"/>

<Label Content="Максимальный:" HorizontalAlignment="Left" Margin="590,211,0,0" VerticalAlignment="Top"/>

<Label Content="Минимальный:" HorizontalAlignment="Left" Margin="590,237,0,0" VerticalAlignment="Top"/>

<Label Name="rt16" Content="" HorizontalAlignment="Left" Margin="684,237,0,0" VerticalAlignment="Top"/>

<Label Name="rt17" Content="" HorizontalAlignment="Left" Margin="684,211,0,0" VerticalAlignment="Top"/>

<Label Name="rt18" Content="" HorizontalAlignment="Left" Margin="639,185,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

using System.Data;

namespace up\_pr2

{

public partial class MainWindow : Window

{

public DataTable Tt(int[,] arr)

{

DataTable output = new DataTable();

for (int i = 0; i < 10; i++)

{

output.Columns.Add(i.ToString());

}

for (int i = 0; i < 10; i++)

{

DataRow row = output.NewRow();

for (int j = 0; j < 10; j++)

row[j.ToString()] = arr[i, j];

output.Rows.Add(row);

}

grid.DataContext = output.DefaultView;

return output;

}

public int[,] DateNew()

{

int[,] arr = new int[10, 10];

Random rnd = new Random();

for (int i = 0; i < 10; i++)

for (int j = 0; j < 10; j++)

arr[i, j] = rnd.Next(0, 100);

return arr;

}

public MainWindow()

{

InitializeComponent();

}

private void Hh(object sender, RoutedEventArgs e)

{

int[,] arr = DateNew();

Tt(arr);

int sum = 0, min = 100, max = 0;

for (int i = 0; i < 10; i++)

{

sum += arr[i, i];

if (arr[i, i] > max)

max = arr[i, i];

if (arr[i, i] < min)

min = arr[i, i];

}

rt1.Content = Convert.ToString(sum);

rt2.Content = max;

rt3.Content = min;

for (int i = 0; i < 10; i++)

{

sum += arr[i, 9 - i];

if (arr[i, 9 - i] > max)

max = arr[i, 9 - i];

if (arr[i, 9 - i] < min)

min = arr[i, 9 - i];

}

rt4.Content = Convert.ToString(sum);

rt5.Content = max;

rt6.Content = min;

sum = 0;

min = 100;

max = 0;

for (int i = 0; i < 5; i++)

for (int j = (0 + i); j < (10 - i); j++)

{

sum += arr[j, i];

if (arr[j, i] > max)

max = arr[j, i];

if (arr[j, i] < min)

min = arr[j, i];

}

rt7.Content = Convert.ToString(sum);

rt8.Content = max;

rt9.Content = min;

sum = 0;

min = 100;

max = 0;

for (int i = 0; i < 5; i++)

for (int j = (0 + i); j < (10 - i); j++)

{

sum += arr[i, j];

if (arr[i, j] > max)

max = arr[i, j];

if (arr[i, j] < min)

min = arr[i, j];

}

rt10.Content = Convert.ToString(sum);

rt11.Content = max;

rt12.Content = min;

sum = 0;

min = 100;

max = 0;

for (int i = 5; i < 10; i++)

for (int j = 9 - i; j < i + 1; j++)

{

sum += arr[j, i];

if (arr[j, i] > max)

max = arr[j, i];

if (arr[j, i] < min)

min = arr[j, i];

}

rt13.Content = Convert.ToString(sum);

rt14.Content = max;

rt15.Content = min;

sum = 0;

min = 100;

max = 0;

for (int i = 5; i < 10; i++)

for (int j = 9 - i; j < i + 1; j++)

{

sum += arr[i, j];

if (arr[i, j] > max)

max = arr[i, j];

if (arr[i, j] < min)

min = arr[i, j];

}

rt16.Content = Convert.ToString(sum);

rt17.Content = max;

rt18.Content = min;

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 3**

**Тема проекта: Работа с классом System.String**

Студент: Берникова В.С.

Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 13.11.2020

**Цель работы**: Разработать программу C#, которая демонстрирует работу со основными свойствами и методами класса String.

**Ниже на рисунке представлен фрагмент работающего проекта**

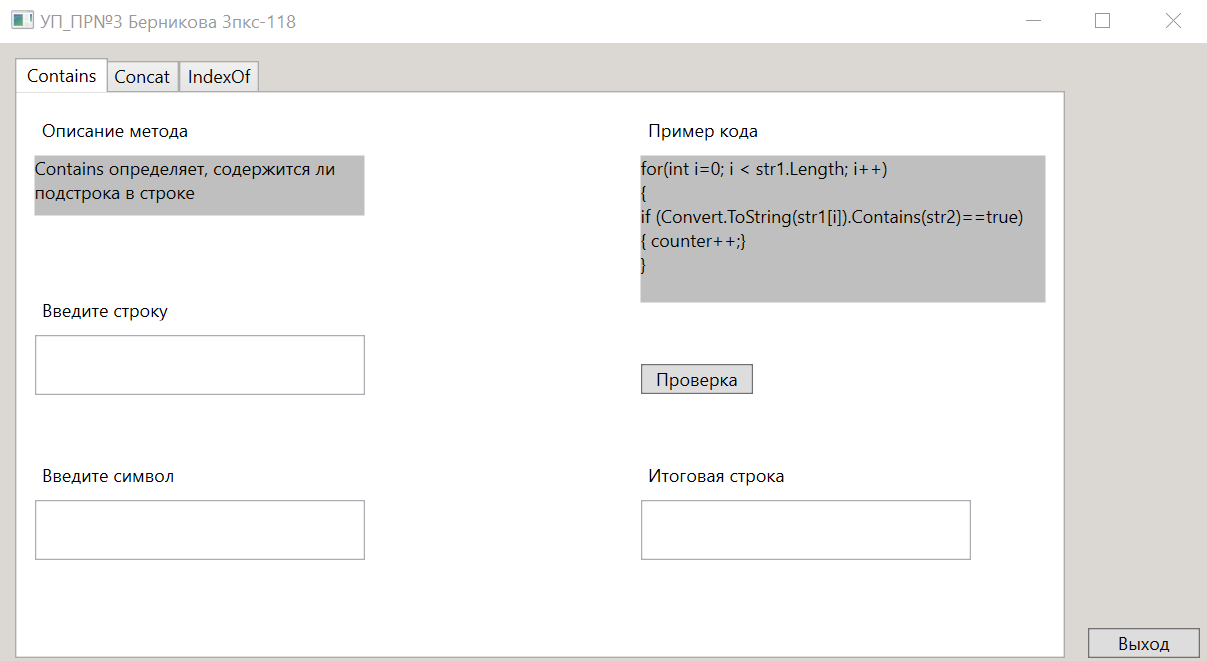


Рис. 1. Метод Contains

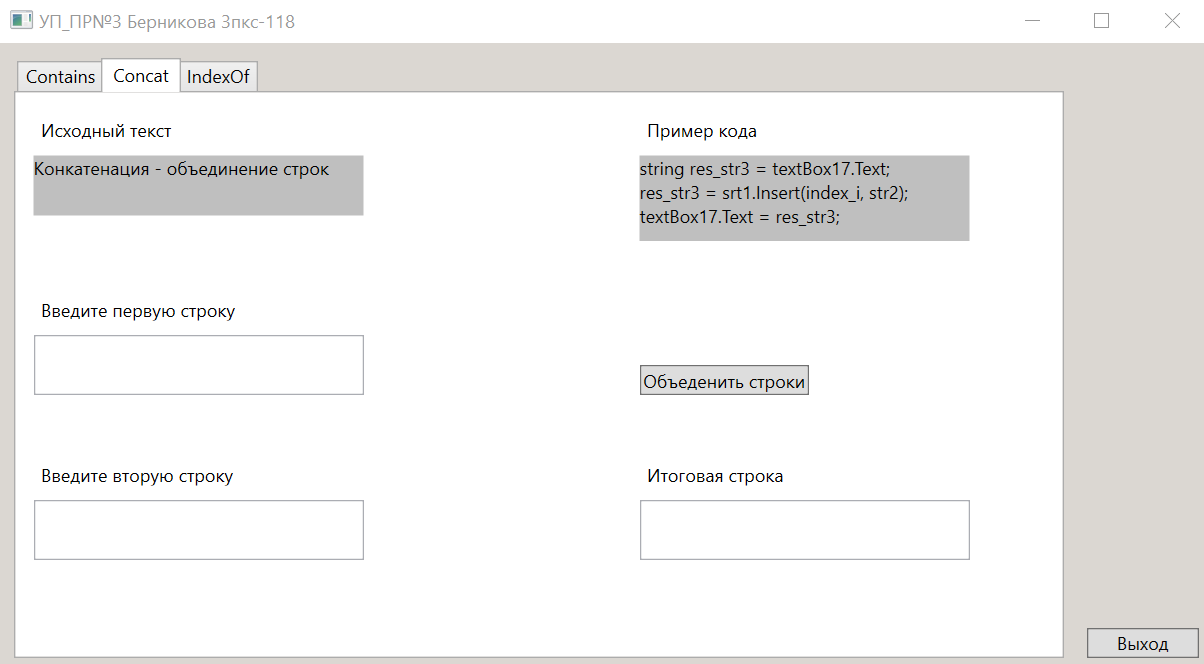


Рис. 2. Метод Concat

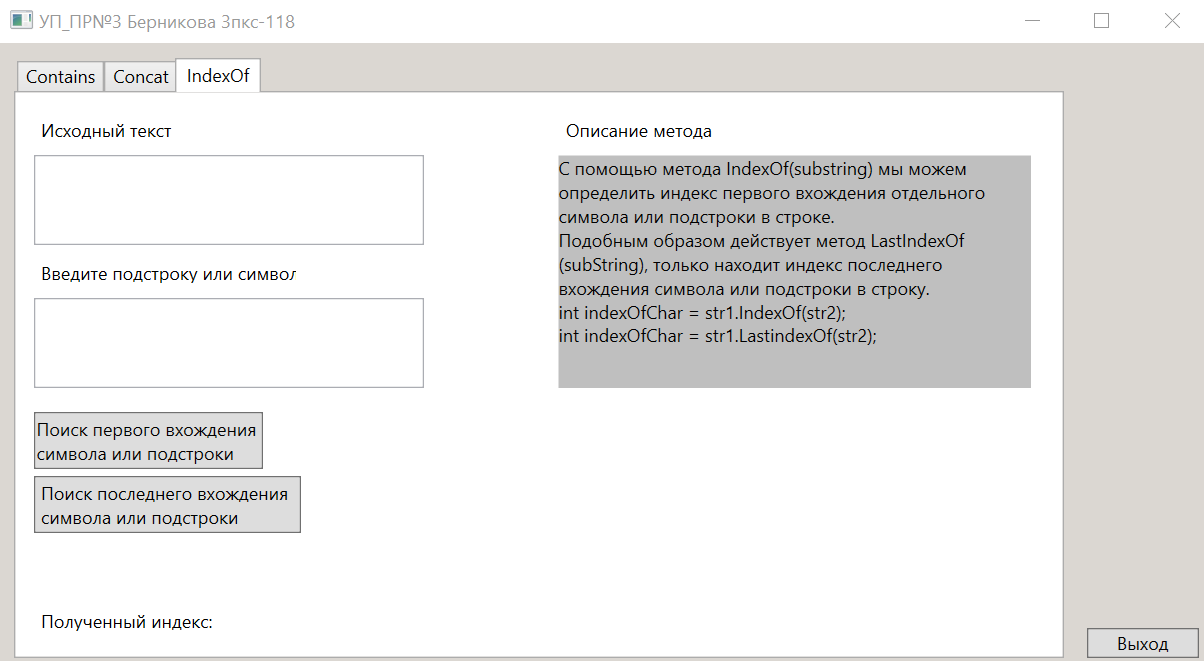


Рис. 3. Метод IndexOf

**Код программы на WPF:**

<Window x:Class="up\_pr3.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="УП\_ПР№3 Берникова 3пкс-118" Height="450" Width="820" Background="#DBD7D2">

<Grid>

<TabControl HorizontalAlignment="Left" Height="400" Margin="10,10,0,0" VerticalAlignment="Top" Width="700">

<TabItem Header="Contains">

<Grid Background="#FFFFFF">

<Label Content="Описание метода" HorizontalAlignment="Left" Height="25" Margin="10,10,0,0" VerticalAlignment="Top" Width="108"/>

<TextBlock HorizontalAlignment="Left" Margin="10,40,0,0" TextWrapping="Wrap" Text="Contains определяет, содержится ли подстрока в строке"

Background="#80808080" VerticalAlignment="Top" Height="40" Width="220"/>

<Label Content="Введите строку" HorizontalAlignment="Left" Height="25" Margin="10,130,0,0" VerticalAlignment="Top" Width="95"/>

<TextBox Name="TextBoxstr1" HorizontalAlignment="Left" Height="40" Margin="10,160,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Label Content="Введите символ" HorizontalAlignment="Left" Height="25" Margin="10,240,0,0" VerticalAlignment="Top" Width="98"/>

<TextBox Name="TextBoxstr2" HorizontalAlignment="Left" Height="40" Margin="10,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Label Content="Пример кода" HorizontalAlignment="Left" Height="25" Margin="414,10,0,0" VerticalAlignment="Top" Width="85"/>

<TextBlock HorizontalAlignment="Left" Margin="414,40,0,0" TextWrapping="Wrap"

Background="#80808080" VerticalAlignment="Top" Height="98" Width="270"><Run Text="for(int i=0; i &lt; str1.Length; i++) "/><LineBreak/><Run Text="{}{ "/><LineBreak/><Run Text="if (Convert.ToString(str1[i]).Contains(str2)==true) { counter++;} "/><LineBreak/><Run Text="}"/></TextBlock>

<Label Content="Итоговая строка" HorizontalAlignment="Left" Height="25" Margin="414,240,0,0" VerticalAlignment="Top" Width="104"/>

<TextBox Name="TextBoxotvet" HorizontalAlignment="Left" Height="40" Margin="414,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Button Name="Buttonotvet1" Content="Проверка" HorizontalAlignment="Left" Margin="414,179,0,0" VerticalAlignment="Top" Width="75" Click="Button\_Click\_1"/>

</Grid>

</TabItem>

<TabItem Header="Concat">

<Grid Background="#FFFFFF" >

<Label Content="Исходный текст" HorizontalAlignment="Left" Height="25" Margin="10,10,0,0" VerticalAlignment="Top" Width="108"/>

<TextBlock HorizontalAlignment="Left" Margin="10,40,0,0" TextWrapping="Wrap" Text="Конкатенация - объединение строк"

Background="#80808080" VerticalAlignment="Top" Height="40" Width="220"/>

<Label Content="Введите первую строку" HorizontalAlignment="Left" Height="25" Margin="10,130,0,0" VerticalAlignment="Top" Width="140"/>

<TextBox Name="TextBoxstr3" HorizontalAlignment="Left" Height="40" Margin="10,160,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Label Content="Введите вторую строку" HorizontalAlignment="Left" Height="25" Margin="10,240,0,0" VerticalAlignment="Top" Width="140"/>

<TextBox Name="TextBoxstr4" HorizontalAlignment="Left" Height="40" Margin="10,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Label Content="Пример кода" HorizontalAlignment="Left" Height="25" Margin="414,10,0,0" VerticalAlignment="Top" Width="85"/>

<TextBlock HorizontalAlignment="Left" Margin="414,40,0,0" TextWrapping="Wrap"

Background="#80808080" VerticalAlignment="Top" Height="57" Width="220"><Run Text="string res\_str3 = textBox17.Text; "/><LineBreak/><Run Text="res\_str3 = srt1.Insert(index\_i, str2); textBox17.Text = res\_str3;"/></TextBlock>

<Label Content="Итоговая строка" HorizontalAlignment="Left" Height="25" Margin="414,240,0,0" VerticalAlignment="Top" Width="104"/>

<TextBox Name="TextBoxstr5" HorizontalAlignment="Left" Height="40" Margin="414,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="220"/>

<Button Content="Объеденить строки" HorizontalAlignment="Left" Margin="414,180,0,0" VerticalAlignment="Top" Width="113" Click="Button\_Click\_2"/>

</Grid>

</TabItem>

<TabItem Header="IndexOf">

<Grid Background="#FFFFFF">

<Label Content="Исходный текст" HorizontalAlignment="Left" Height="25" Margin="10,10,0,0" VerticalAlignment="Top" Width="140"/>

<TextBox Name="TextBoxtext" HorizontalAlignment="Left" Height="60" Margin="10,40,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="260"/>

<Label Content="Введите подстроку или символ" HorizontalAlignment="Left" Height="25" Margin="10,105,0,0" VerticalAlignment="Top" Width="180"/>

<TextBox Name="TextBoxelem" HorizontalAlignment="Left" Height="60" Margin="10,135,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="260"/>

<Label Content="Описание метода" HorizontalAlignment="Left" Height="25" Margin="360,10,0,0" VerticalAlignment="Top" Width="113"/>

<TextBlock HorizontalAlignment="Left" Margin="360,40,0,0" TextWrapping="Wrap"

Text="С помощью метода IndexOf(substring) мы можем определить индекс первого вхождения отдельного символа или подстроки в строке.

Подобным образом действует метод LastIndexOf (subString), только находит индекс последнего вхождения символа или подстроки в строку.

int indexOfChar = str1.IndexOf(str2);

int indexOfChar = str1.LastindexOf(str2);"

Background="#80808080" VerticalAlignment="Top" Height="155" Width="315"/>

<Button Content="Поиск первого вхождения &#xD;&#xA;символа или подстроки" HorizontalAlignment="Left" Margin="10,211,0,0" VerticalAlignment="Top" Width="153" Height="38" Click="Button\_Click\_3"/>

<Button Content="Поиск последнего вхождения &#xD;&#xA;символа или подстроки" HorizontalAlignment="Left" Margin="10,254,0,0" VerticalAlignment="Top" Width="178" Height="38" Click="Button\_Click\_4"/>

<Label Content="Полученный индекс:" HorizontalAlignment="Left" Height="25" Margin="10,337,0,0" VerticalAlignment="Top" Width="125"/>

<TextBlock Name="TextBlockotvet3" HorizontalAlignment="Left" Margin="140,343,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"/>

</Grid>

</TabItem>

</TabControl>

<Button Content="Выход" HorizontalAlignment="Left" Margin="725,390,0,0" VerticalAlignment="Top" Width="75" Click="Button\_Click"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

namespace up\_pr3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e) // Contain

{

string str1 = Convert.ToString(TextBoxstr1.Text);

string str2 = Convert.ToString(TextBoxstr2.Text);

int itogovayz\_stroka=0;

for (int i = 0; i < str1.Length; i++)

{

if (Convert.ToString(str1[i]).Contains(str2) == true)

{

itogovayz\_stroka++;

}

}

TextBoxotvet.Text = Convert.ToString(itogovayz\_stroka);

}

private void Button\_Click\_2(object sender, RoutedEventArgs e) // Concat

{

string str3 = Convert.ToString(TextBoxstr3.Text);

string str4 = Convert.ToString(TextBoxstr4.Text);

string itogovayz\_stroka;

itogovayz\_stroka = String.Concat(str3, str4);

TextBoxstr5.Text = Convert.ToString(itogovayz\_stroka);

}

private void Button\_Click\_3(object sender, RoutedEventArgs e) // perv\_simvl

{

string text = Convert.ToString(TextBoxtext.Text);

string elem = Convert.ToString(TextBoxelem.Text);

int perv\_index;

perv\_index = text.IndexOf(elem);

TextBlockotvet3.Text = Convert.ToString(perv\_index);

}

private void Button\_Click\_4(object sender, RoutedEventArgs e) // posl\_simvl

{

string text = Convert.ToString(TextBoxtext.Text);

string elem = Convert.ToString(TextBoxelem.Text);

int posl\_index;

posl\_index = text.LastIndexOf(elem);

TextBlockotvet3.Text = Convert.ToString(posl\_index);

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 4**

**Тема проекта: Работа с делегатами (часть 1)**

Студент: Берникова В.С.

Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 20.11.2020

**Цель работы**: Изучить теоретический материал и выполнить консольный проект, демонстрирующий работу с делегатами.

**Ниже на рисунке представлен фрагмент работающего проекта**

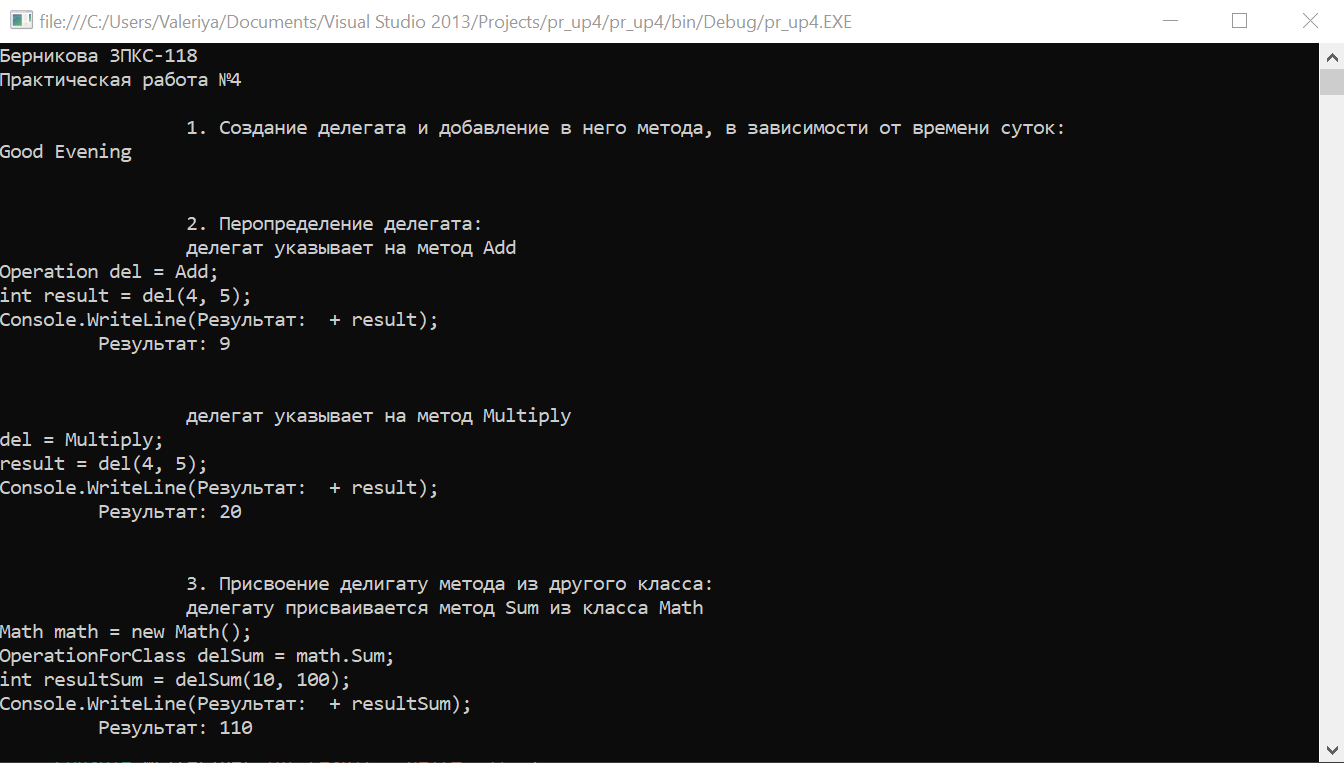


Рис. 1. Делегаты

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace pr\_up4

{

class Math

{

public int Sum(int x, int y) { return x + y; }

}

class Program

{

delegate int Operation(int x, int y);

delegate int OperationForClass(int x, int y);

delegate void Message();

static void Main(string[] args)

{

Console.WriteLine("Берникова 3ПКС-118");

Console.WriteLine("Практическая работа №4\n");

// делегата Message()

Console.WriteLine("\t\t 1. Создание делегата и добавление в него метода, в зависимости от времени суток:");

Message mes;

if (DateTime.Now.Hour < 12) {

mes = GoodMorning; }

else {

mes = GoodEvening; }

mes();

Console.WriteLine("\n");

// делегата Перопределение делегата: делегат указывает на метод Add

Console.WriteLine("\t\t 2. Перопределение делегата:");

Console.WriteLine("\t\t делегат указывает на метод Add");

Console.WriteLine("Operation del = Add;");

Console.WriteLine("int result = del(4, 5);");

Console.WriteLine("Console.WriteLine(Результат: + result);");

Operation del = Add;

int result = del(4, 5);

Console.WriteLine("\t Результат: " + result);

Console.WriteLine("\n");

// делегата Переопределение делегата: делегат указывает на метод Multiply

Console.WriteLine("\t\t делегат указывает на метод Multiply");

Console.WriteLine("del = Multiply;");

Console.WriteLine("result = del(4, 5);");

Console.WriteLine("Console.WriteLine(Результат: + result);");

del = Multiply;

result = del(4, 5);

Console.WriteLine("\t Результат: " + result);

Console.WriteLine("\n");

// Присвоение делегату метода из другого класса: делегату присваивается метод Sum из класса Math

Console.WriteLine("\t\t 3. Присвоение делигату метода из другого класса:");

Console.WriteLine("\t\t делегату присваивается метод Sum из класса Math");

Console.WriteLine("Math math = new Math();");

Console.WriteLine("OperationForClass delSum = math.Sum;");

Console.WriteLine("int resultSum = delSum(10, 100);");

Console.WriteLine("Console.WriteLine(Результат: + resultSum);");

Math math = new Math();

OperationForClass delSum = math.Sum;

int resultSum = delSum(10, 100);

Console.WriteLine("\t Результат: " + resultSum);

Console.Read();

}

private static void GoodMorning()

{

Console.WriteLine("Good Morning");

}

private static void GoodEvening()

{

Console.WriteLine("Good Evening");

}

private static int Add(int x, int y)

{

return x + y;

}

private static int Multiply(int x, int y)

{

return x \* y;

}

private static int delSum(int x, int y)

{

return x + y;

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 5**

**Тема проекта: Работа с делегатами (часть 2)**

Студент: Берникова В.С.

Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 27.11.2020

**Цель работы**: Изучить теоретический материал и выполнить консольный проект, демонстрирующий работу с делегатами.

**Ниже на рисунке представлен фрагмент работающего проекта**

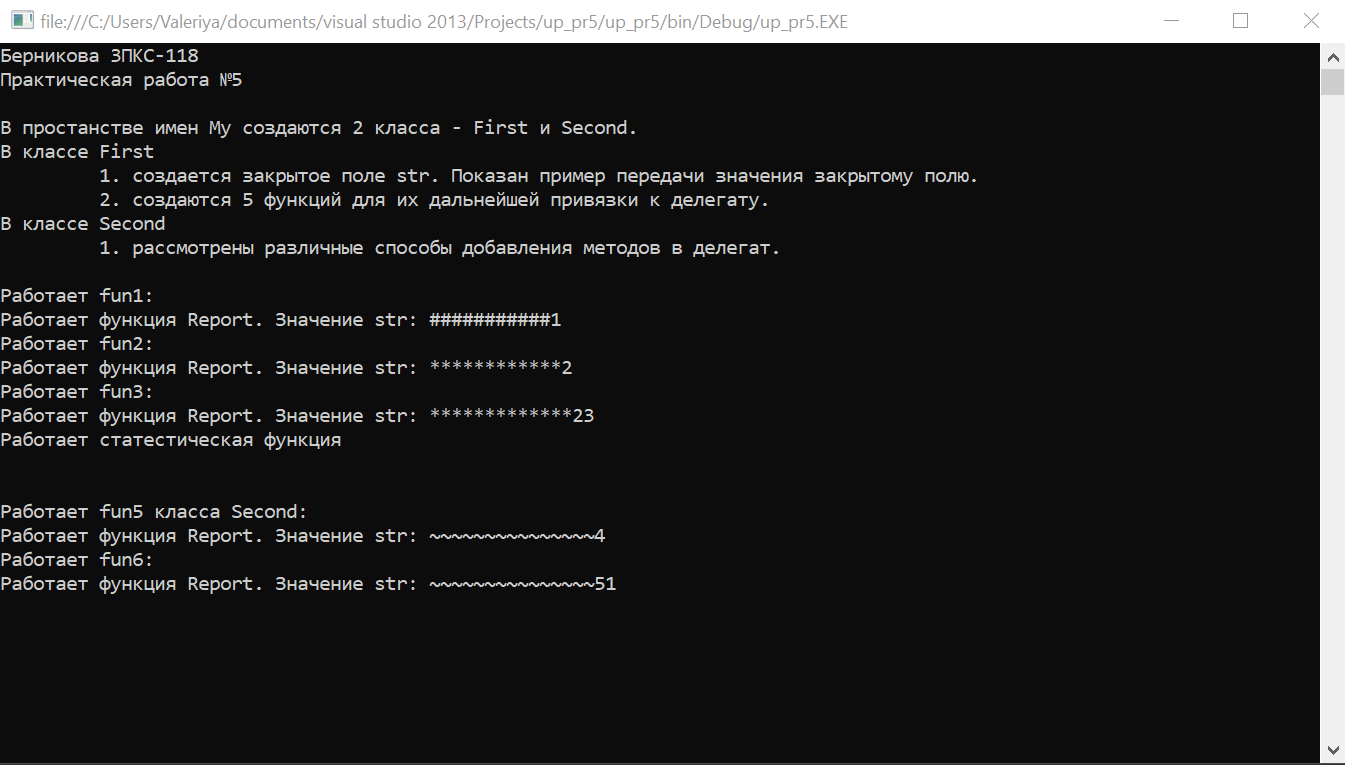


Рис. 1. Делегаты

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace up\_pr5

{

class First

{

public static string str { get; set; }

public static void fun1()

{

Console.WriteLine("Берникова 3ПКС-118");

Console.WriteLine("Практическая работа №5\n");

Console.WriteLine("В простанстве имен Му создаются 2 класса - First и Second.");

Console.WriteLine("В классе First");

Console.WriteLine("\t 1. создается закрытое поле str. Показан пример передачи значения закрытому полю.");

Console.WriteLine("\t 2. создаются 5 функций для их дальнейшей привязки к делегату.");

Console.WriteLine("В классе Second");

Console.WriteLine("\t 1. рассмотрены различные способы добавления методов в делегат.\n");

str = "###########1";

Console.WriteLine("Работает fun1:");

}

public static void fun2()

{

str = "\*\*\*\*\*\*\*\*\*\*\*\*2";

Console.WriteLine("Работает fun2:");

}

public static void fun3()

{

str = "\*\*\*\*\*\*\*\*\*\*\*\*\*23";

Console.WriteLine("Работает fun3:");

}

}

class Second

{

public static void fun4()

{

Console.WriteLine("Работает статестическая функция");

Console.WriteLine("\n");

First.str = "~~~~~~~~~~~~~~~4";

Console.WriteLine("Работает fun5 класса Second:");

}

public static void fun5()

{

First.str = "~~~~~~~~~~~~~~~51";

Console.WriteLine("Работает fun6:");

}

}

class Rechenie

{

delegate void Report();

static void Main(string[] args)

{

Report del;

del = First.fun1;

del += ReportD;

del();

del = First.fun2;

del += ReportD;

del();

del = First.fun3;

del += ReportD;

del();

del = Second.fun4;

del += ReportD;

del();

del = Second.fun5;

del += ReportD;

del();

Console.ReadKey();

}

public static void ReportD()

{

Console.WriteLine("Работает функция Report. Значение str: " + First.str);

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 6**

**Тема проекта: «События С#».**

Студент: Берникова В.С.

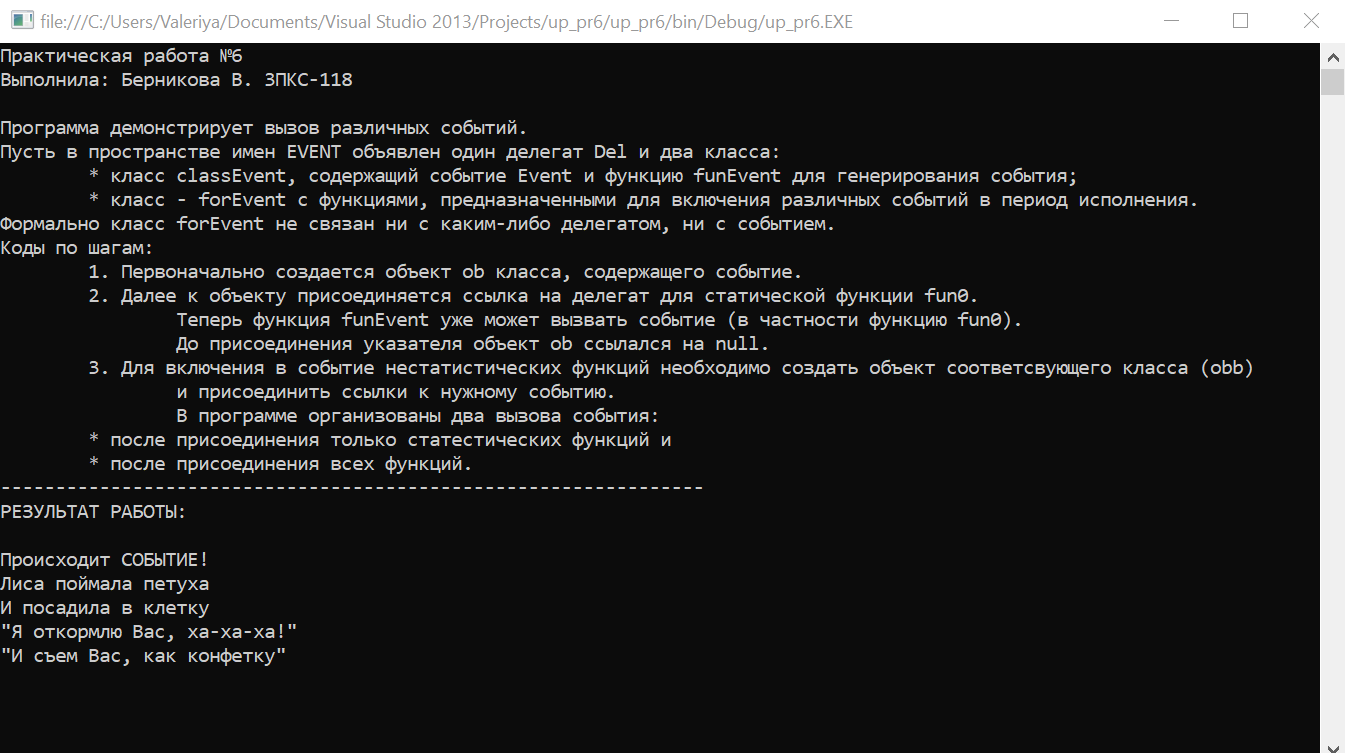
Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 04.12.2020

**Цель проекта:** разработать проект C#, в котором генерируется события и осуществляются различные формы привязки событий к делегатам.



**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace up\_pr6

{

delegate void Del();

class classEvent

{

public event Del Event;

public void Del2()

{Event();}

}

class forEvent

{

public string ob { set { funEvent = value; } get { return funEvent; } }

string funEvent;

public void fun0(string ob)

{this.ob = ob;}

public void Obb()

{

Console.WriteLine("Лиса поймала петуха " + ob);

Console.WriteLine("И посадила в клетку" + ob);

Console.WriteLine(@"""Я откормлю Вас, ха-ха-ха!""" + ob);

Console.WriteLine(@"""И съем Вас, как конфетку""" + ob);

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Практическая работа №6");

Console.WriteLine("Выполнила: Берникова В. 3ПКС-118\n");

Console.WriteLine("Программа демонстрирует вызов различных событий.");

Console.WriteLine("Пусть в пространстве имен EVENT объявлен один делегат Del и два класса:");

Console.WriteLine("\t\* класс classEvent, содержащий событие Event и функцию funEvent для генерирования события;");

Console.WriteLine("\t\* класс - forEvent с функциями, предназначенными для включения различных событий в период исполнения.");

Console.WriteLine("Формально класс forEvent не связан ни с каким-либо делегатом, ни с событием.");

Console.WriteLine("Коды по шагам:");

Console.WriteLine("\t1. Первоначально создается объект ob класса, содержащего событие.");

Console.WriteLine("\t2. Далее к объекту присоединяется ссылка на делегат для статической функции fun0.");

Console.WriteLine("\t\tТеперь функция funEvent уже может вызвать событие (в частности функцию fun0).");

Console.WriteLine("\t\tДо присоединения указателя объект ob ссылался на null.");

Console.WriteLine("\t3. Для включения в событие нестатистических функций необходимо создать объект соответсвующего класса (obb)");

Console.WriteLine("\t\tи присоединить ссылки к нужному событию.");

Console.WriteLine("\t\tВ программе организованы два вызова события:");

Console.WriteLine("\t\* после присоединения только статестических функций и");

Console.WriteLine("\t\* после присоединения всех функций.");

Console.WriteLine("----------------------------------------------------------------");

Console.WriteLine("РЕЗУЛЬТАТ РАБОТЫ:\n");

Console.WriteLine("Происходит СОБЫТИЕ!");

classEvent del = new classEvent();

forEvent del2 = new forEvent();

del.Event += del2.Obb;

del.Del2();

Console.ReadKey();

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 7**

**Тема проекта: «События С#».**

Студент: Берникова В.С.

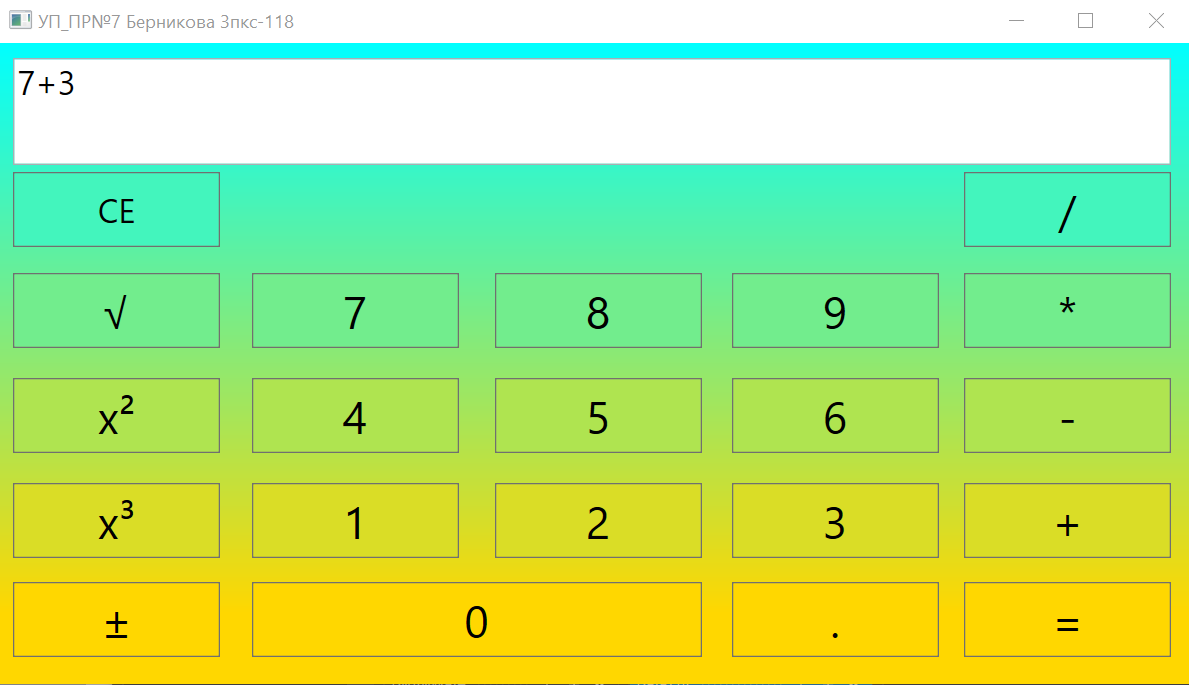
Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

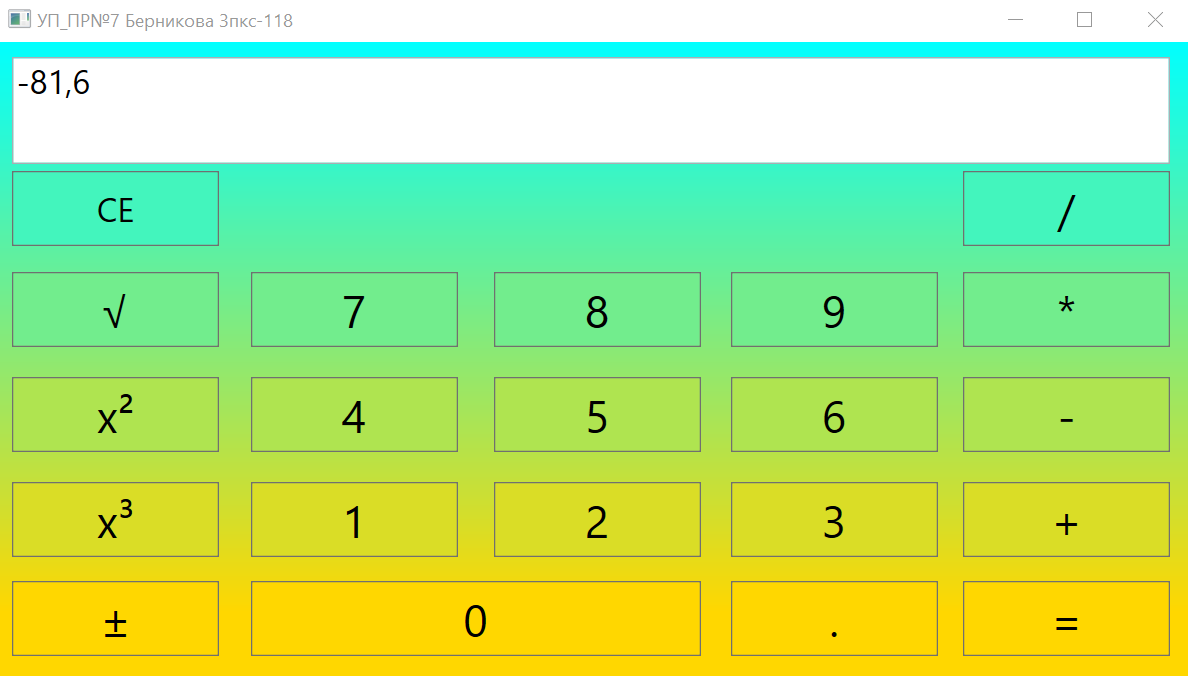
Дата: 11.12.2020

**Цель проекта:** разработать калькулятор на языке С#. В проекте использовать прием создания одного обработчика для нескольких событий.













**Код программы на WPF:**

<Window x:Class="up\_pr7.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="УП\_ПР№7 Берникова 3пкс-118" Height="464.958" Width="809.972">

<Grid x:Name="Calculyator">

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Aqua" Offset="0"/>

<GradientStop Color="Gold" Offset="0.889"/>

</LinearGradientBrush>

</Grid.Background>

<TextBox Name="TextBox\_Vichislenie" HorizontalAlignment="Left" Height="71" Margin="10,10,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="772" FontSize="22"/>

<Button Name="Button\_CE" Content="CE" HorizontalAlignment="Left" Height="50" Margin="10,86,0,0" VerticalAlignment="Top" Width="138" FontSize="22" Background="#FF43F5BD" />

<Button Name="Button\_Koren" Content="√" HorizontalAlignment="Left" Height="50" Margin="10,153,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF72ED8D"/>

<Button Name="Button\_7" Content="7" HorizontalAlignment="Left" Height="50" Margin="169,153,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF72ED8D"/>

<Button Name="Button\_8" Content="8" HorizontalAlignment="Left" Height="50" Margin="331,153,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF72ED8D"/>

<Button Name="Button\_9" Content="9" HorizontalAlignment="Left" Height="50" Margin="489,153,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF72ED8D"/>

<Button Name="Button\_Umnojenie" Content="\*" HorizontalAlignment="Left" Height="50" Margin="644,153,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF72ED8D"/>

<Button Name="Button\_Delenie" Content="/" HorizontalAlignment="Left" Height="50" Margin="644,86,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FF43F5BD"/>

<Button Name="Button\_Vtoraya\_Stepen" Content="x²" HorizontalAlignment="Left" Height="50" Margin="10,223,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFAFE450"/>

<Button Name="Button\_Tretiya\_Stepen" Content="x³" HorizontalAlignment="Left" Height="50" Margin="10,293,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFDADD26"/>

<Button Name="Button\_4" Content="4" HorizontalAlignment="Left" Height="50" Margin="169,223,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFAFE450"/>

<Button Name="Button\_5" Content="5" HorizontalAlignment="Left" Height="50" Margin="331,223,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFAFE450"/>

<Button Name="Button\_6" Content="6" HorizontalAlignment="Left" Height="50" Margin="489,223,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFAFE450"/>

<Button Name="Button\_Vichitanie" Content="-" HorizontalAlignment="Left" Height="50" Margin="644,223,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFAFE450"/>

<Button Name="Button\_1" Content="1" HorizontalAlignment="Left" Height="50" Margin="169,293,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFDADD26"/>

<Button Name="Button\_2" Content="2" HorizontalAlignment="Left" Height="50" Margin="331,293,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFDADD26"/>

<Button Name="Button\_3" Content="3" HorizontalAlignment="Left" Height="50" Margin="489,293,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFDADD26"/>

<Button Name="Button\_Slojenie" Content="+" HorizontalAlignment="Left" Height="50" Margin="644,293,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="#FFDADD26"/>

<Button Name="Button\_Plus\_Minec" Content="±" HorizontalAlignment="Left" Height="50" Margin="10,359,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="Gold"/>

<Button Name="Button\_Tochka" Content="." HorizontalAlignment="Left" Height="50" Margin="489,359,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="Gold"/>

<Button Name="Button\_Ravno" Content="=" HorizontalAlignment="Left" Height="50" Margin="644,359,0,0" VerticalAlignment="Top" Width="138" FontSize="30" Background="Gold"/>

<Button Name="Button\_0" Content="0" HorizontalAlignment="Left" Height="50" Margin="169,359,0,0" VerticalAlignment="Top" Width="300" FontSize="30" Background="Gold"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

using System.Data;

namespace up\_pr7

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

foreach (UIElement el in Calculyator.Children)

{

if (el is Button)

{

((Button)el).Click += Button\_Click;

}

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string vvod = (string)((Button)e.OriginalSource).Content;

switch (vvod)

{

case "CE":

{

TextBox\_Vichislenie.Text = "";

break;

}

case "=":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

TextBox\_Vichislenie.Text = chislo;

break;

}

case "x²":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

var kvadrat = Convert.ToDouble(chislo);

kvadrat = Math.Pow(kvadrat, 2);

chislo = Convert.ToString(kvadrat);

TextBox\_Vichislenie.Text = chislo;

break;

}

case "x³":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

var kub = Convert.ToDouble(chislo);

kub = Math.Pow(kub, 3);

chislo = Convert.ToString(kub);

TextBox\_Vichislenie.Text = chislo;

break;

}

case "±":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

var plus\_minus = Convert.ToDouble(chislo);

plus\_minus = -plus\_minus;

chislo = Convert.ToString(plus\_minus);

TextBox\_Vichislenie.Text = chislo;

break;

}

case ".":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

chislo = TextBox\_Vichislenie.Text + '.';

TextBox\_Vichislenie.Text = chislo;

break;

}

case "√":

{

string chislo = new DataTable().Compute(TextBox\_Vichislenie.Text, null).ToString();

var koren = Convert.ToDouble(chislo);

koren = Math.Sqrt(koren);

chislo = Convert.ToString(koren);

TextBox\_Vichislenie.Text = chislo;

break;

}

default:

{

TextBox\_Vichislenie.Text += vvod;

break;

}

}

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 8**

**Тема проекта: Работа с MS Office в WPF.**

Студент: Берникова В.С.

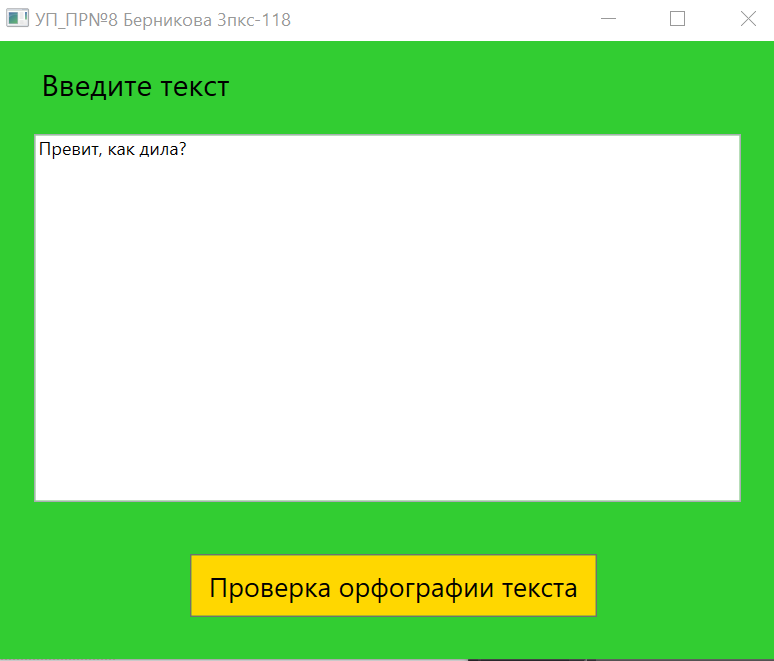
Группа: 3ПКС-118

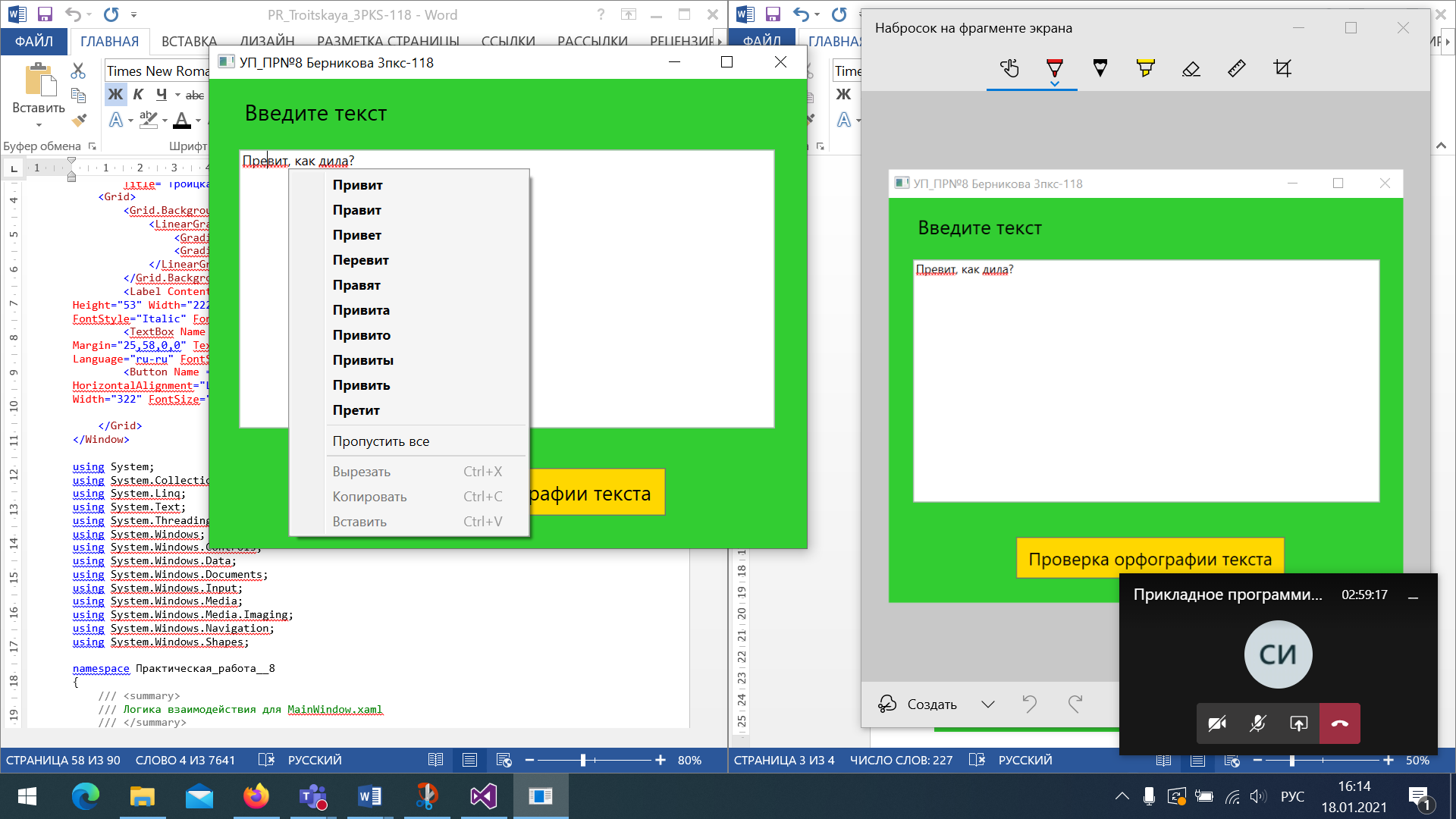
Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 18.12.2020

**Цель проекта:** разработать программу, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и после нажатия соответствующей кнопки проверить орфографию введенного текста.





**Код программы на WPF:**

<Window x:Class="up\_pr8.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="УП\_ПР№8 Берникова 3пкс-118" Height="450" Width="540">

<Grid Background="LimeGreen">

<Label Content="Введите текст" HorizontalAlignment="Left" Height="36" Margin="26,10,0,0" VerticalAlignment="Top" Width="217" FontSize="20"/>

<TextBox Name="TextBox\_vvod" SpellCheck.IsEnabled="False" Language="ru-ru" AcceptsReturn="True"

HorizontalAlignment="Left" Height="245" Margin="26,62,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="471"/>

<Button Content="Проверка орфографии текста" HorizontalAlignment="Left" Height="42" Margin=

"130,342,0,0" VerticalAlignment="Top" Width="271" FontSize="18" Click="Button\_Click" Background="Gold"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

namespace up\_pr8

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

TextBox\_vvod.SpellCheck.IsEnabled = true;

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 9**

**Тема проекта: Работа с MS Office в WPF.**

Студент: Берникова В.С.

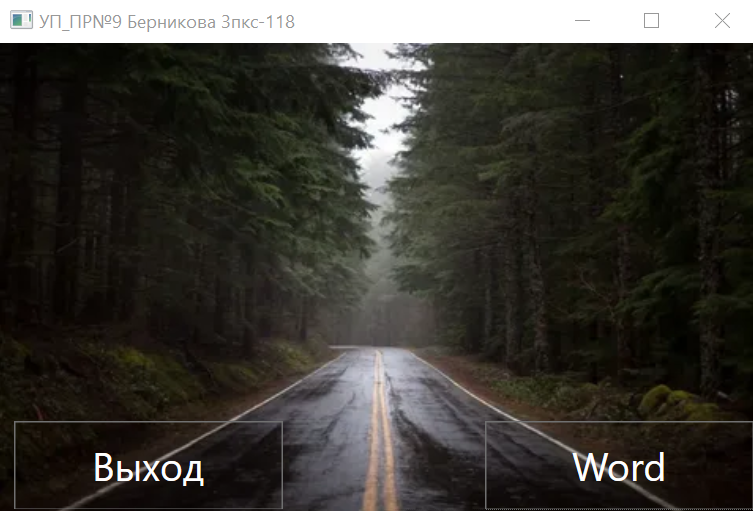
Группа: 3ПКС-118

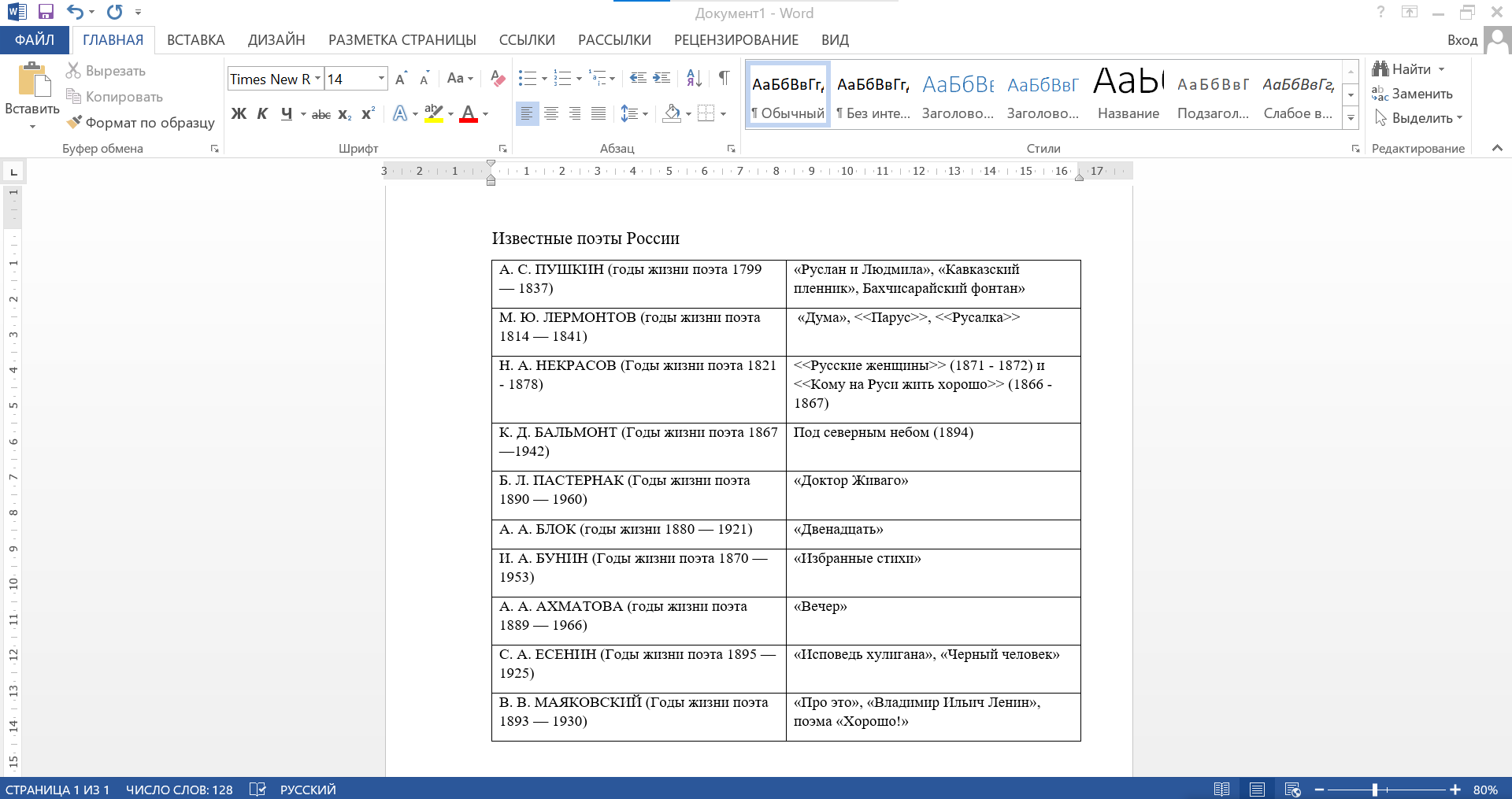
Преподаватель: Титов Н.Г.

Оценка за проект:

Дата: 25.12.2020

**Цель проекта:** необходимо разработать программу, которая при нажатии на кнопку создает таблицу в MS Word.





**Код WPF:**

<Window x:Class="up\_pr9.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="УП\_ПР№9 Берникова 3пкс-118" Height="350" Width="520">

<Grid>

<Image Grid.ColumnSpan="2" Margin="0,0,0.333,-0.333" Source="фон пр 9.jpg" Stretch="Fill"/>

<Button Name ="Button\_Vihod" Content="Выход" HorizontalAlignment="Left" Height="59" Margin="10,252,0,0"

VerticalAlignment="Top" Width="179" FontSize="26" Background="{x:Null}" Foreground="White" Click="Button\_Vihod\_Click"/>

<Button Name ="Button\_Word" Content="Word" HorizontalAlignment="Left" Height="59" Margin="324,252,0,0"

VerticalAlignment="Top" Width="179" FontSize="26" Background="{x:Null}" Foreground="White" Click="Button\_Word\_Click"/>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

using Word = Microsoft.Office.Interop.Word;

namespace up\_pr9

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Word\_Click(object sender, RoutedEventArgs e)

{

Word.Application word\_app = new Word.Application();

word\_app.Visible = true;

object missing = Type.Missing;

Word.Document word\_doc = word\_app.Documents.Add(Visible: true);

object obekt1 = 0;

object obekt2 = 0;

Word.Range range = word\_doc.Range(ref obekt1, ref obekt2);

range.InsertBefore("Известные поэты России");

range.Font.Name = "Times New Roman";

range.Font.Size = 14;

range.InsertParagraphAfter();

range.InsertParagraphAfter();

range.SetRange(range.End, range.End);

range.Font.Name = "Times New Roman";

range.Font.Size = 14;

range.Tables.Add(word\_doc.Paragraphs[2].Range, 10, 2, ref missing, ref missing);

Word.Table table = word\_doc.Tables[1];

table.Range.Font.Size = 12;

table.Range.Font.Name = "Times New Roman";

table.Columns.DistributeWidth();

table.Cell(1, 1).Range.Text = "А. С. ПУШКИН (годы жизни поэта 1799 — 1837)";

table.Cell(1, 2).Range.Text = "«Руслан и Людмила», «Кавказский пленник», Бахчисарайский фонтан»";

table.Cell(2, 1).Range.Text = "М. Ю. ЛЕРМОНТОВ (годы жизни поэта 1814 — 1841)";

table.Cell(2, 2).Range.Text = " «Дума», <<Парус>>, <<Русалка>>";

table.Cell(3, 1).Range.Text = "Н. А. НЕКРАСОВ (Годы жизни поэта 1821 - 1878)";

table.Cell(3, 2).Range.Text = "<<Русские женщины>> (1871 - 1872) и <<Кому на Руси жить хорошо>> (1866 - 1867)";

table.Cell(4, 1).Range.Text = "К. Д. БАЛЬМОНТ (Годы жизни поэта 1867 —1942)";

table.Cell(4, 2).Range.Text = "Под северным небом (1894)";

table.Cell(5, 1).Range.Text = "Б. Л. ПАСТЕРНАК (Годы жизни поэта 1890 — 1960)";

table.Cell(5, 2).Range.Text = "«Доктор Живаго»";

table.Cell(6, 1).Range.Text = "А. А. БЛОК (годы жизни 1880 — 1921)";

table.Cell(6, 2).Range.Text = "«Двенадцать»";

table.Cell(7, 1).Range.Text = "И. А. БУНИН (Годы жизни поэта 1870 — 1953)";

table.Cell(7, 2).Range.Text = "«Избранные стихи»";

table.Cell(8, 1).Range.Text = "А. А. АХМАТОВА (годы жизни поэта 1889 — 1966)";

table.Cell(8, 2).Range.Text = "«Вечер»";

table.Cell(9, 1).Range.Text = "С. А. ЕСЕНИН (Годы жизни поэта 1895 — 1925)";

table.Cell(9, 2).Range.Text = "«Исповедь хулигана», «Черный человек» ";

table.Cell(10, 1).Range.Text = "В. В. МАЯКОВСКИЙ (Годы жизни поэта 1893 — 1930)";

table.Cell(10, 2).Range.Text = "«Про это», «Владимир Ильич Ленин», поэма «Хорошо!»";

table.Borders.Enable = 1;

}

private void Button\_Vihod\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 10**

**Тема проекта: Работа с MS Office в WPF.**

Студент: Берникова В.С.

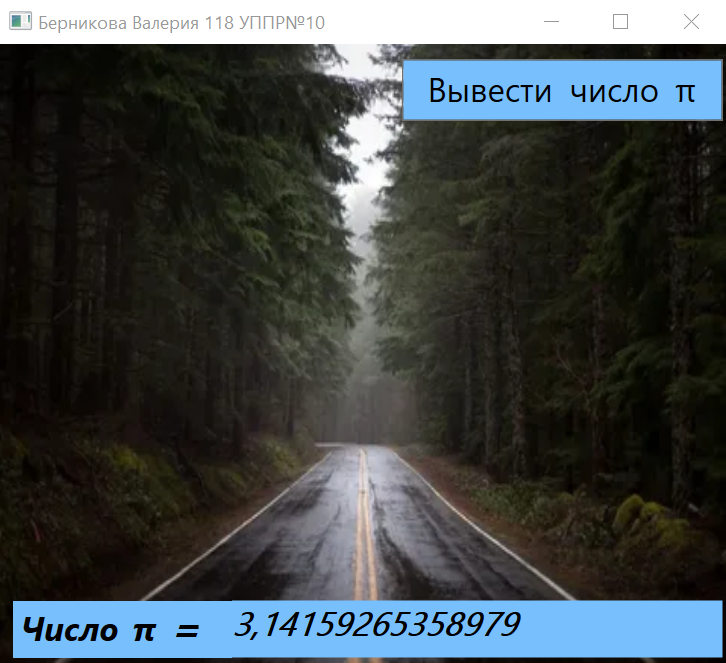
Группа: 3ПКС-118

Преподаватель: Титов Н.Г.

Оценка за проект: 5

Дата: 09.01.2021

**Цель проекта:** цель данной программы - показать легкость доступа к функциям MS Excel.



**Код программы на WPF:**

<Window x:Class="up\_pr10.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Берникова Валерия 118 УППР№10" Height="450" Width="500">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="Снимок.PNG.jpg"/>

</Grid.Background>

<Label Content="Число π =" HorizontalAlignment="Left" Height="38" Margin="10,371,0,0" VerticalAlignment="Top"

Width="146" FontSize="22" FontStyle="Italic" FontWeight="Bold" FontFamily="Snap ITC" Background="#FF77BFFD"/>

<TextBlock Name="TextBlock\_Pi" HorizontalAlignment="Left" Height="38" Margin="156,371,0,0" TextWrapping="Wrap"

Text="" VerticalAlignment="Top" Width="327" FontSize="22" FontWeight="Bold" FontFamily="Segoe UI Symbol" FontStyle="Italic" Background="#FF77BFFD"/>

<Button Name="Button\_Pi" Content="Вывести число π" HorizontalAlignment="Left" Height="41" Margin="269,10,0,0"

VerticalAlignment="Top" Width="214" FontSize="22" Click="Button\_Pi\_Click" FontFamily="SimSun-ExtB" Background="#FF77BFFD" />

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

using Exc = Microsoft.Office.Interop;

namespace up\_pr10

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Pi\_Click(object sender, RoutedEventArgs e)

{

var XL = new Microsoft.Office.Interop.Excel.Application();

var PI = XL.WorksheetFunction.Pi();

TextBlock\_Pi.Text = Math.PI.ToString();

XL.Quit();

}

}

}

**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРОЕКТА № 11**

**Тема проекта: Работа с файлами с различными кодировками**

Студент: Берникова В.С.

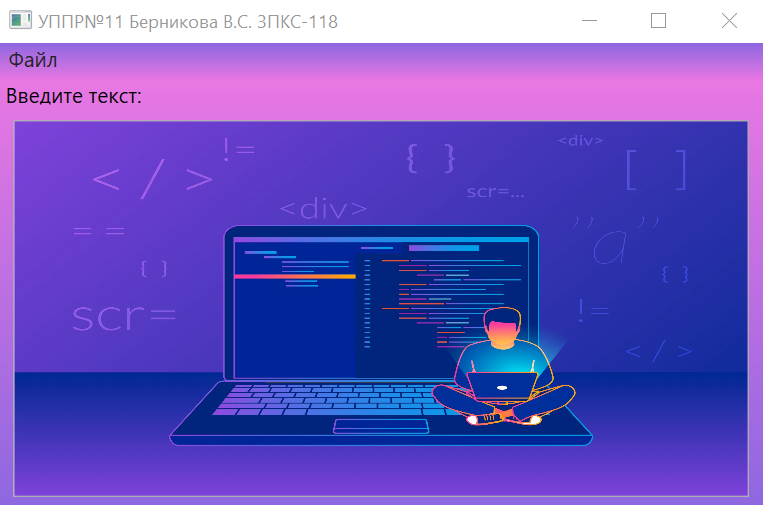
Группа: 3ПКС-118

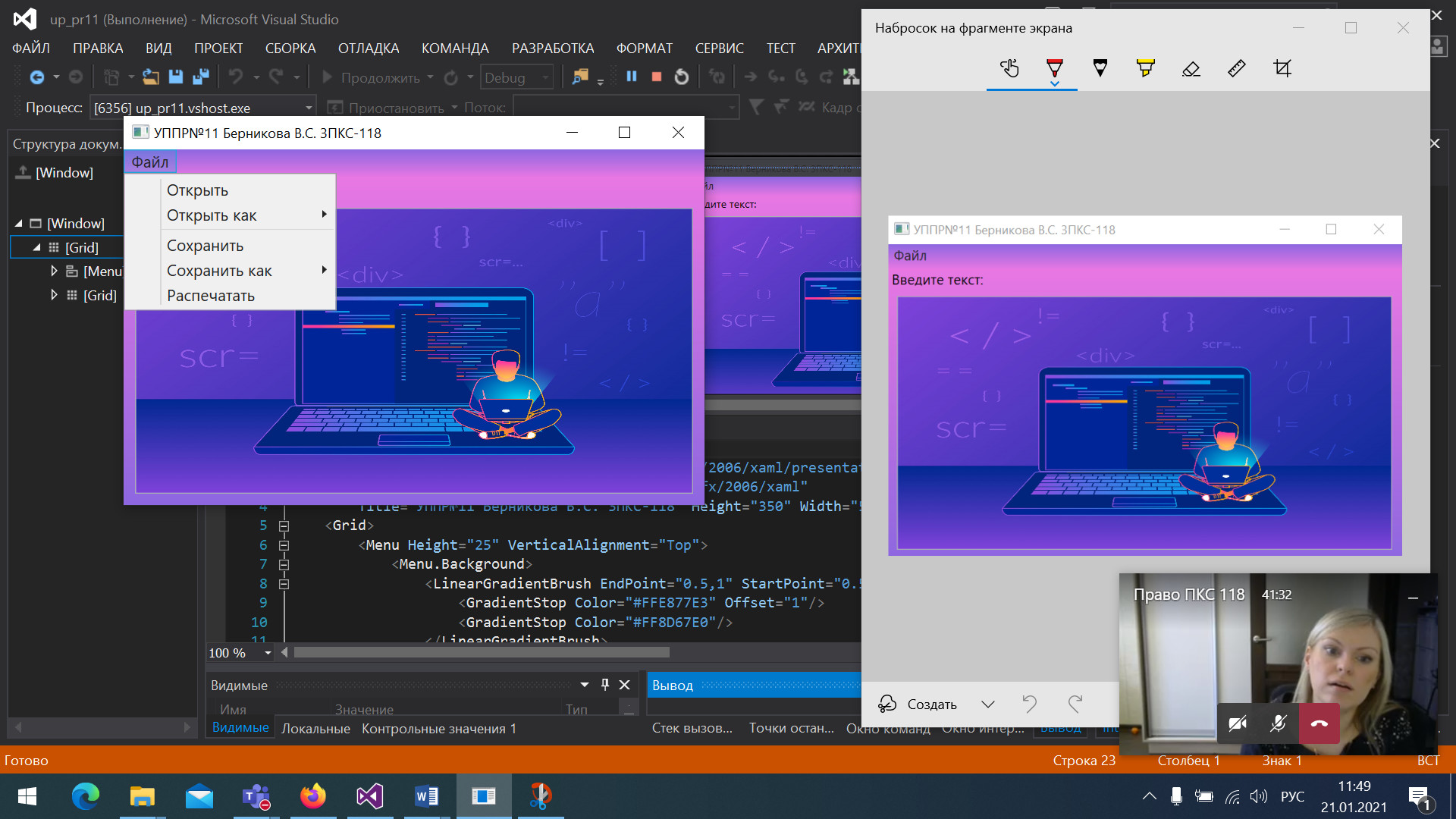
Преподаватель: Титов Н.Г.

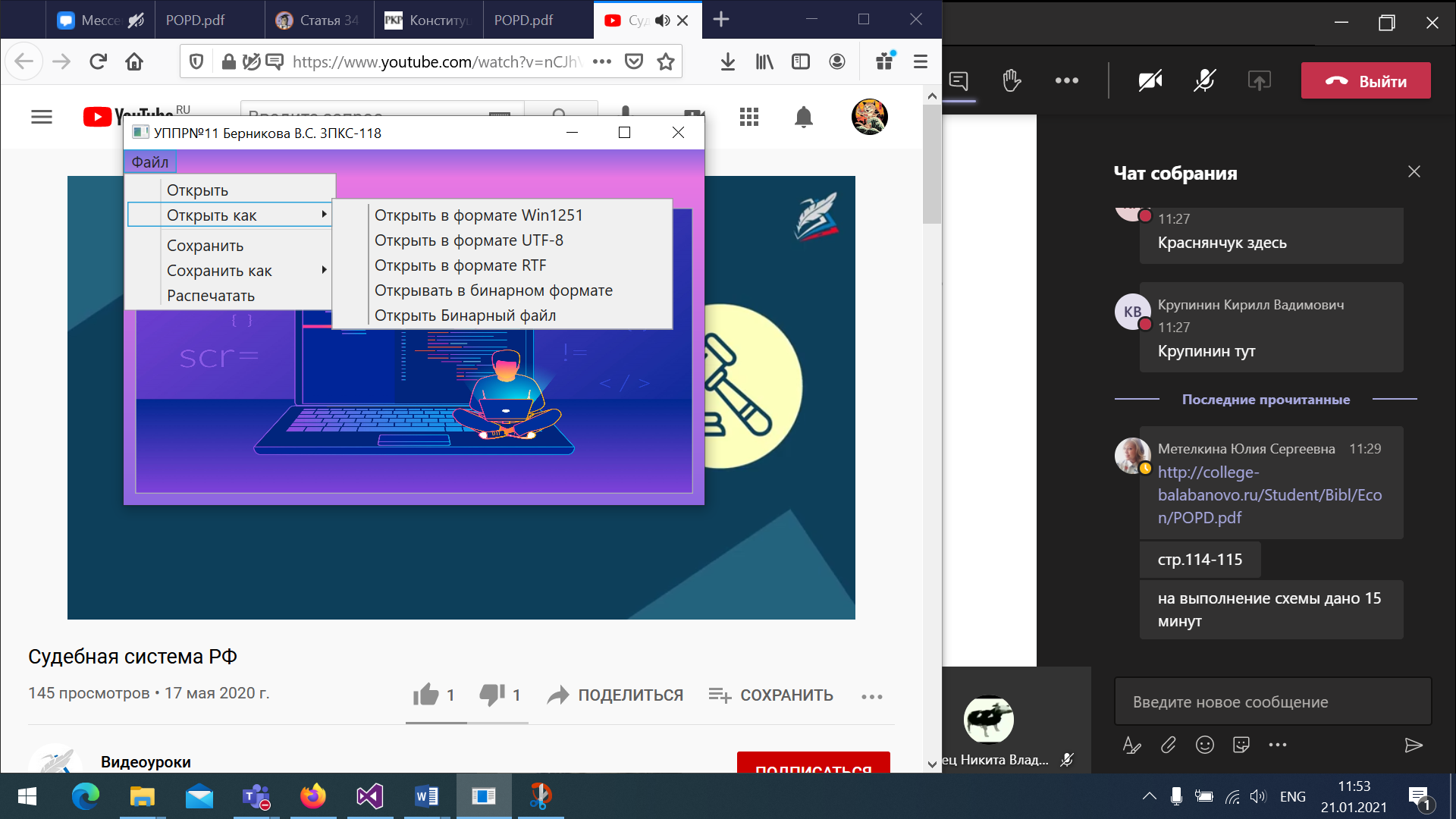
Оценка за проект:

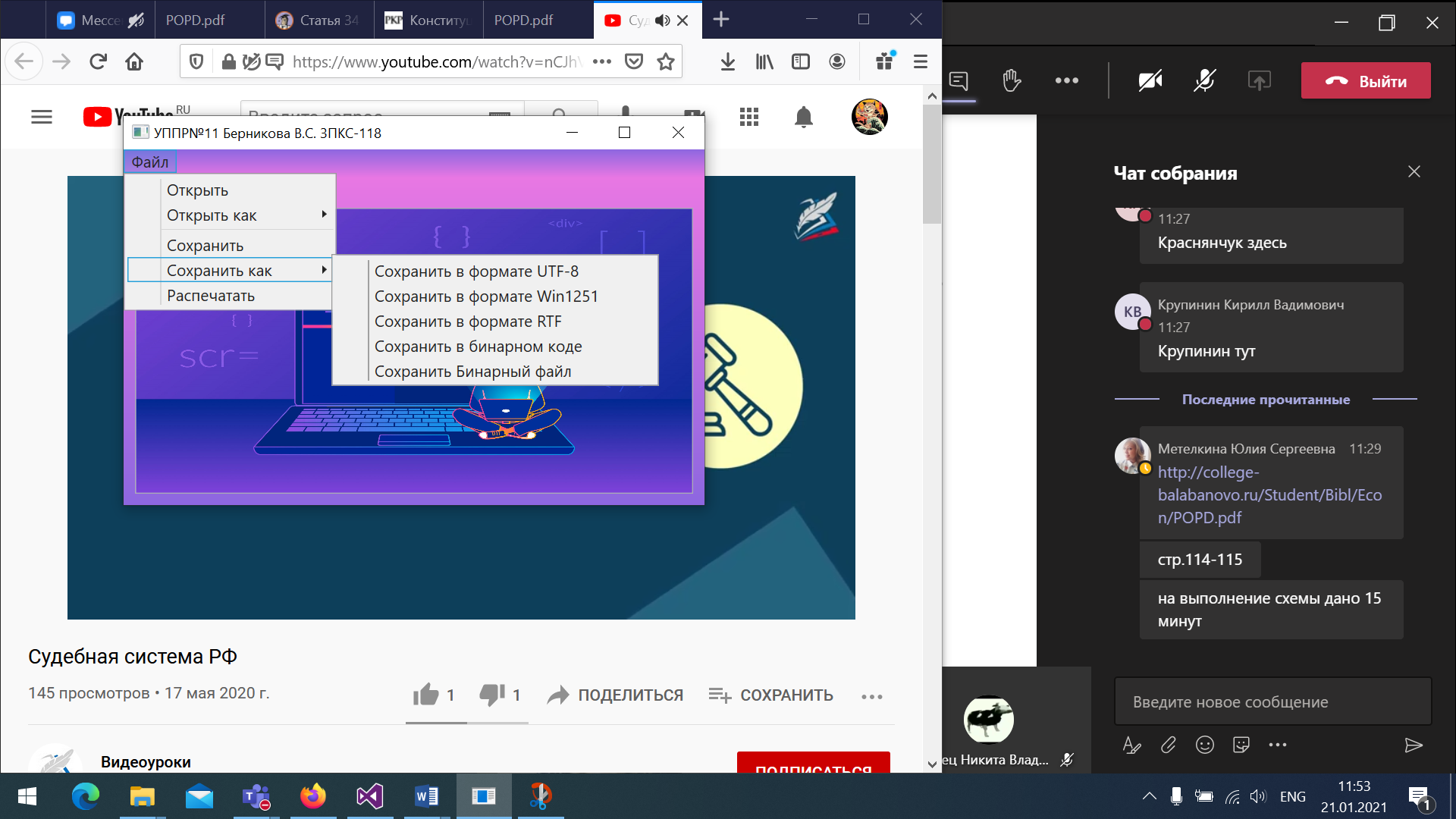
Дата: 15.01.2021

**Цель проекта:** написать программу на языке C#, которая реализует работу с файлами в различных кодировках.









**Код WPF:**

<Window x:Class="up\_pr11.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="УППР№11 Берникова В.С. 3ПКС-118" Height="350" Width="525">

<Grid>

<Menu Height="25" VerticalAlignment="Top">

<Menu.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#FFE877E3" Offset="1"/>

<GradientStop Color="#FF8D67E0"/>

</LinearGradientBrush>

</Menu.Background>

<MenuItem Header="Файл" FontSize="14">

<MenuItem Name="MenuItem\_Open" Header="Открыть" Click="MenuItem\_Open\_Click" ></MenuItem>

<MenuItem Name="MenuItem\_Open\_As" Header="Открыть как" >

<MenuItem Name="MenuItem\_Open\_Win1251" Header="Открыть в формате Win1251" Click="MenuItem\_Open\_Win1251\_Click">

</MenuItem>

<MenuItem Name="MenuItem\_Open\_UTF8" Header="Открыть в формате UTF-8" Click="MenuItem\_Open\_UTF8\_Click"></MenuItem>

<MenuItem Name="MenuItem\_Open\_RTF" Header="Открыть в формате RTF" Click="MenuItem\_Open\_RTF\_Click"></MenuItem>

<MenuItem Name="MenuItem\_Open\_Binaron\_Format" Header="Открывать в бинарном формате" Click="MenuItem\_Open\_Binaron\_Format\_Click" ></MenuItem>

<MenuItem Name="MuneItem\_Open\_Binary\_File" Header="Открыть Бинарный файл" Click="MenuItem\_Open\_Binary\_File\_Click" ></MenuItem>

</MenuItem>

<Separator />

<MenuItem Name="MenuItem\_Save" Header="Сохранить" Click="MenuItem\_Save\_Click" >

</MenuItem>

<MenuItem Header="Сохранить как " >

<MenuItem Name="MenuItem\_Save\_UTF8" Header="Сохранить в формате UTF-8" Click="MenuItem\_Save\_UTF8\_Click"></MenuItem>

<MenuItem Name="MenuItem\_Save\_Win1251" Header="Сохранить в формате Win1251" Click="MenuItem\_Save\_Win1251\_Click"></MenuItem>

<MenuItem Name="MenuItem\_Save\_RTF" Header="Сохранить в формате RTF" Click="MenuItem\_Save\_RTF\_Click" ></MenuItem>

<MenuItem Name="MenuItem\_Save\_Binaron\_Format" Header="Сохранить в бинарном коде" Click="MenuItem\_Save\_Binaron\_Format\_Click" ></MenuItem>

<MenuItem Name="MenuItem\_Save\_Binary\_File" Header="Сохранить Бинарный файл" Click="MenuItem\_Save\_Binary\_File\_Click" ></MenuItem>

</MenuItem>

<MenuItem Name="MenuItem\_Print\_Out" Header="Распечатать" Click="MenuItem\_Print\_Out\_Click" ></MenuItem>

</MenuItem>

</Menu>

<Grid Margin="0,25,0,0">

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#FF8D67E0" Offset="1"/>

<GradientStop Color="#FFE877E3"/>

</LinearGradientBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<RichTextBox x:Name="RichTextBox\_Text" Margin="10,26,10,10" FontSize="14" >

<RichTextBox.Background>

<ImageBrush ImageSource="1.jpg"/>

</RichTextBox.Background>

</RichTextBox>

<Label Content="Введите текст:" HorizontalAlignment="Left" Height="26" VerticalAlignment="Top" Width="106" FontSize="14" Margin="0,-5,0,0"/>

</Grid>

</Grid>

</Window>

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.IO;

using System.Text;

using System.Threading.Tasks;

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.Navigation;

using System.Windows.Shapes;

using Microsoft.Win32;

using Path = System.IO.Path;

using Word = Microsoft.Office.Interop.Word;

namespace up\_pr11

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void MenuItem\_Open\_Click(object sender, RoutedEventArgs e)

{

RichTextBox\_Text.Document.Blocks.Clear();

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "RichText Files (\*.rtf)|\*.rtf|All files (\*.\*)|\*.\*";

if (ofd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

using (FileStream fs = new FileStream(ofd.FileName, FileMode.Open))

{

if (Path.GetExtension(ofd.FileName).ToLower() == ".rtf")

doc.Load(fs, DataFormats.Rtf);

else if (Path.GetExtension(ofd.FileName).ToLower() == ".txt")

doc.Load(fs, DataFormats.Text);

else

doc.Load(fs, DataFormats.Xaml);

}

}

}

private void MenuItem\_Save\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Text Files (\*.txt)|\*.txt|RichText Files (\*.rtf)|\*.rtf|XAML Files (\*.xaml)|\*.xaml|All files (\*.\*)|\*.\*";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

using (FileStream fs = File.Create(sfd.FileName))

{

if (Path.GetExtension(sfd.FileName).ToLower() == ".rtf")

doc.Save(fs, DataFormats.Rtf);

else if (Path.GetExtension(sfd.FileName).ToLower() == ".txt")

doc.Save(fs, DataFormats.Text);

else

doc.Save(fs, DataFormats.Xaml);

}

}

}

private void MenuItem\_Open\_Win1251\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.FileName = "";

openFileDialog1.Filter = "Текстовые файлы (\*.txt) | \*.txt";

openFileDialog1.ShowDialog();

var text = new System.IO.StreamReader(openFileDialog1.FileName, Encoding.GetEncoding(1251));

RichTextBox\_Text.AppendText(text.ReadToEnd());

text.Close();

}

private void MenuItem\_Open\_RTF\_Click(object sender, RoutedEventArgs e)

{

RichTextBox\_Text.Document.Blocks.Clear();

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "RichText Files (\*.rtf)|\*.rtf|All files (\*.\*)|\*.\*";

if (ofd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

using (FileStream fs = new FileStream(ofd.FileName, FileMode.Open))

{

if (Path.GetExtension(ofd.FileName).ToLower() == ".rtf")

doc.Load(fs, DataFormats.Rtf);

}

}

}

private void MenuItem\_Open\_Binaron\_Format\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.FileName = "";

openFileDialog1.ShowDialog();

openFileDialog1.Filter = "Файлы USP|\*.usp|Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

var text = new System.IO.StreamReader(openFileDialog1.FileName, Encoding.GetEncoding(1251));

text.Close();

var textBin = new System.IO.StreamReader(openFileDialog1.FileName, Encoding.GetEncoding(1251));

string textBin2 = textBin.ReadToEnd();

StringBuilder sb = new StringBuilder();

foreach (char c in textBin2.ToCharArray())

{

sb.Append(Convert.ToString(c, 2).PadLeft(8, '0'));

}

RichTextBox\_Text.AppendText(sb.ToString());

}

private void MenuItem\_Open\_Binary\_File\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.Filter = "Файлы USP|\*.usp|Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

openFileDialog1.FileName = "";

openFileDialog1.ShowDialog();

var textBin = new System.IO.StreamReader(openFileDialog1.FileName, Encoding.GetEncoding(1251));

RichTextBox\_Text.AppendText(textBin.ReadToEnd());

}

private void MenuItem\_Save\_RTF\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Файлы RTF (\*.RTF) | \*.RTF";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

using (FileStream fs = File.Create(sfd.FileName))

{

doc.Save(fs, DataFormats.Rtf);

}

}

}

private void MenuItem\_Save\_Win1251\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = sfd.Filter = "Текстовые файлы (\*.txt) | \*.txt";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

string str = doc.Text;

File.WriteAllText(sfd.FileName, str, Encoding.GetEncoding(1251));

}

}

private void MenuItem\_Save\_Binaron\_Format\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "USP|\*usp|Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

var texts = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd).Text;

StringBuilder sb = new StringBuilder();

foreach (char c in texts)

{

sb.Append(Convert.ToString(c, 2).PadLeft(8, '0'));

}

saveFileDialog1.ShowDialog();

System.IO.File.WriteAllText(saveFileDialog1.FileName, sb.ToString());

}

private void MenuItem\_Save\_Binary\_File\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "USP|\*usp|Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

string str = doc.Text;

File.WriteAllText(sfd.FileName, str, Encoding.GetEncoding(1251));

}

}

private void MenuItem\_Print\_Out\_Click(object sender, RoutedEventArgs e)

{

PrintDialog pd = new PrintDialog();

if ((pd.ShowDialog() == true))

{

pd.PrintVisual(RichTextBox\_Text as Visual, "Print Visual");

pd.PrintDocument((((IDocumentPaginatorSource)RichTextBox\_Text.Document).DocumentPaginator), "Print Document");

}

}

private void MenuItem\_Open\_UTF8\_Click(object sender, RoutedEventArgs e)

{

RichTextBox\_Text.Document.Blocks.Clear();

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = ofd.Filter = "Unicode Text (\*.txt)|\*.txt |All files (\*.\*)|\*.\*";

if (ofd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

var text = new System.IO.StreamReader(ofd.FileName, Encoding.GetEncoding(1200));

doc.Text = text.ReadToEnd();

}

}

private void MenuItem\_Save\_UTF8\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = sfd.Filter = "Unicode Text (\*.txt)|\*.txt |All files (\*.\*)|\*.\*";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(RichTextBox\_Text.Document.ContentStart, RichTextBox\_Text.Document.ContentEnd);

string str = doc.Text;

File.WriteAllText(sfd.FileName, str, Encoding.UTF8);

}

}

}

}