**25. Технология Windows Presentation Foundation**

Задание 1. Изучить теоретический материал. Выполнить на практике все примеры создания WPF приложения.

Решение:

<Window x:Class="Task1.MainWindow"

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

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

xmlns:task1="clr-namespace:Task1"

Title="MainWindow" Height="350" Width="525">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<TextBlock x:Name="textBlockHello" Grid.Row="0" Grid.ColumnSpan="2" Margin="10"/>

<TextBox x:Name="textBoxEnterName" Grid.Row="1" Grid.ColumnSpan="2" Margin="10"/>

<Button x:Name="buttonRun" Content="Сказать Имя" Grid.Row="2" Width="200" Height="60" Margin="10" Click="Button\_Click"/>

<Button x:Name="buttonAbout" Content="Справка" Grid.Row="2" Grid.Column="1" Width="200" Height="60" Margin="10" Click="ButtonAbout\_Click"/>

</Grid>

</Window>

using System.Windows;

namespace Task1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

buttonRun.Click += Button\_Click;

buttonAbout.Click += ButtonAbout\_Click;

}

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

{

string info = $"О программе: {this.Title}";

AboutApp about = new AboutApp(info);

about.ShowDialog();

}

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

{

string name = textBoxEnterName.Text;

textBlockHello.Text = $"Привет, {name}!";

}

}

}

Анализ результатов:

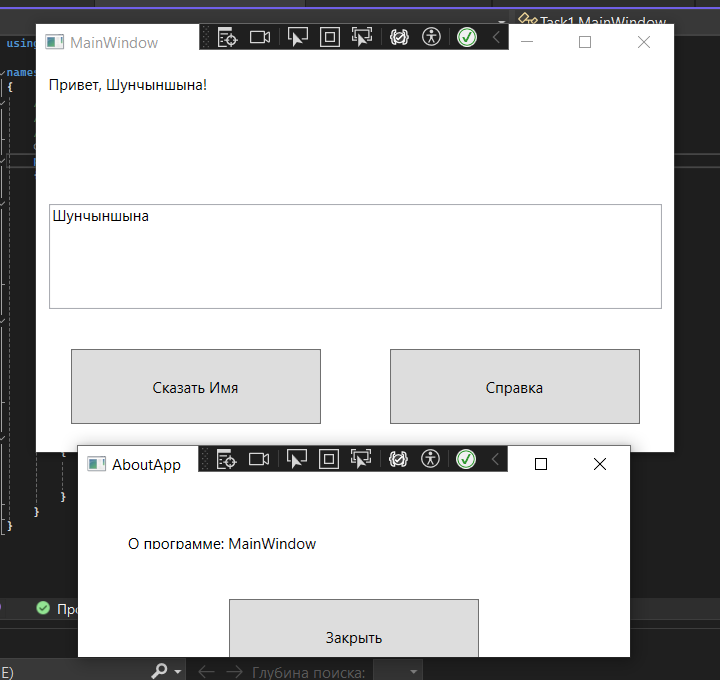


Рисунок 25.1 – Результат задания

Задание 2: Переделать задание 1 таким образом чтобы главная форма приняла следующий вид и все элементы выполняли необходимые действия. Измените название главной формы на “Приложение Hello Name”

Решение:

<Window x:Class="Task2.MainWindow"

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

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

Title="MainWindow" Height="350" Width="525">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<TextBlock x:Name="textBlockHello" Grid.Row="0" Grid.Column="1" Margin="10" Text="Привет Мир!" TextAlignment="Center"/>

<TextBox x:Name="textBoxEnterName" Grid.Row="1" Grid.ColumnSpan="3" Margin="10" TextAlignment="Center"/>

<Button x:Name="buttonRun" Content="Сказать Имя" Grid.Row="2" Width="Auto" Height="60" Margin="10" Click="Button\_Click"/>

<Button x:Name="buttonAbout" Content="Справка" Grid.Row="2" Grid.Column="2" Width="Auto" Height="60" Margin="10" Click="ButtonAbout\_Click"/>

<Button x:Name="buttonExit" Content="Закрыть" Grid.Row="0" Grid.Column="2" Width="150" Height="60" Click="buttonExit\_Click"/>

</Grid>

</Window>

using System.Windows;

namespace Task2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

buttonRun.Click += Button\_Click;

buttonAbout.Click += ButtonAbout\_Click;

}

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

{

string info = $"О программе: {this.Title}";

AboutApp about = new AboutApp(info);

about.ShowDialog();

}

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

{

string name = textBoxEnterName.Text;

textBlockHello.Text = $"Привет, {name}!";

}

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

{

Close();

}

}

}

Анализ результатов:

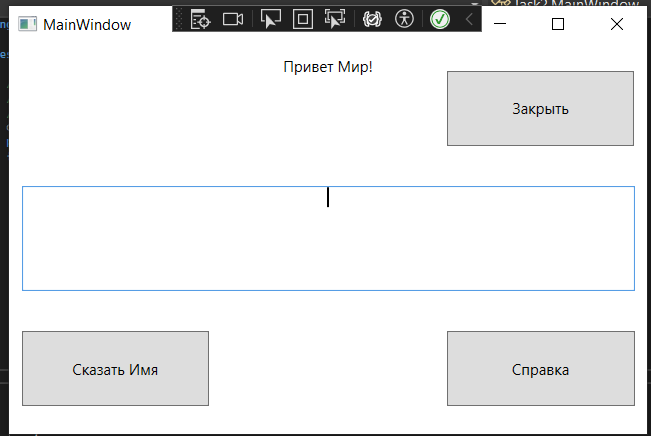


Рисунок 25.2 – Результат задания

Задание 3: Создайте проект WPF и выполните решение алгоритма y=x^2. Элементы управления Button TextBlock, TextBox. Расположение элементов продумайте самостоятельно. Построить график функции. Таблицу данных получить путём изменения параметра X с шагом h. Самостоятельно выбрать удобные параметры настройки.

Решение:

<Window x:Class="ExponentialGraph.MainWindow"

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

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

Title="Exponential Graph" Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10">

<TextBlock Text="Начальное значение X: " VerticalAlignment="Center"/>

<TextBox Name="StartXTextBox" Width="50" Margin="5"/>

<TextBlock Text="Конечное значение X: " VerticalAlignment="Center"/>

<TextBox Name="EndXTextBox" Width="50" Margin="5"/>

<TextBlock Text="Шаг h: " VerticalAlignment="Center"/>

<TextBox Name="StepTextBox" Width="50" Margin="5"/>

<Button Content="Построить график" Click="PlotButton\_Click" Margin="5"/>

</StackPanel>

<Canvas Name="GraphCanvas" Grid.Row="1" Background="White"/>

</Grid>

</Window>

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Shapes;

namespace ExponentialGraph

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

double startX, endX, step;

if (!double.TryParse(StartXTextBox.Text, out startX) ||

!double.TryParse(EndXTextBox.Text, out endX) ||

!double.TryParse(StepTextBox.Text, out step) ||

step <= 0)

{

MessageBox.Show("Пожалуйста, введите корректные значения.");

return;

}

DrawGraph(startX, endX, step);

}

private void DrawGraph(double startX, double endX, double step)

{

GraphCanvas.Children.Clear();

double canvasWidth = GraphCanvas.ActualWidth;

double canvasHeight = GraphCanvas.ActualHeight;

double xRange = endX - startX;

double maxY = Math.Exp(endX);

double minY = Math.Exp(startX);

for (double x = startX; x <= endX; x += step)

{

double y = Math.Exp(x);

double canvasX = (x - startX) / xRange \* canvasWidth;

double canvasY = canvasHeight - ((y - minY) / (maxY - minY) \* canvasHeight);

Ellipse point = new Ellipse

{

Width = 3,

Height = 3,

Fill = Brushes.Red

};

Canvas.SetLeft(point, canvasX);

Canvas.SetTop(point, canvasY);

GraphCanvas.Children.Add(point);

}

}

}

}

Анализ результатов:

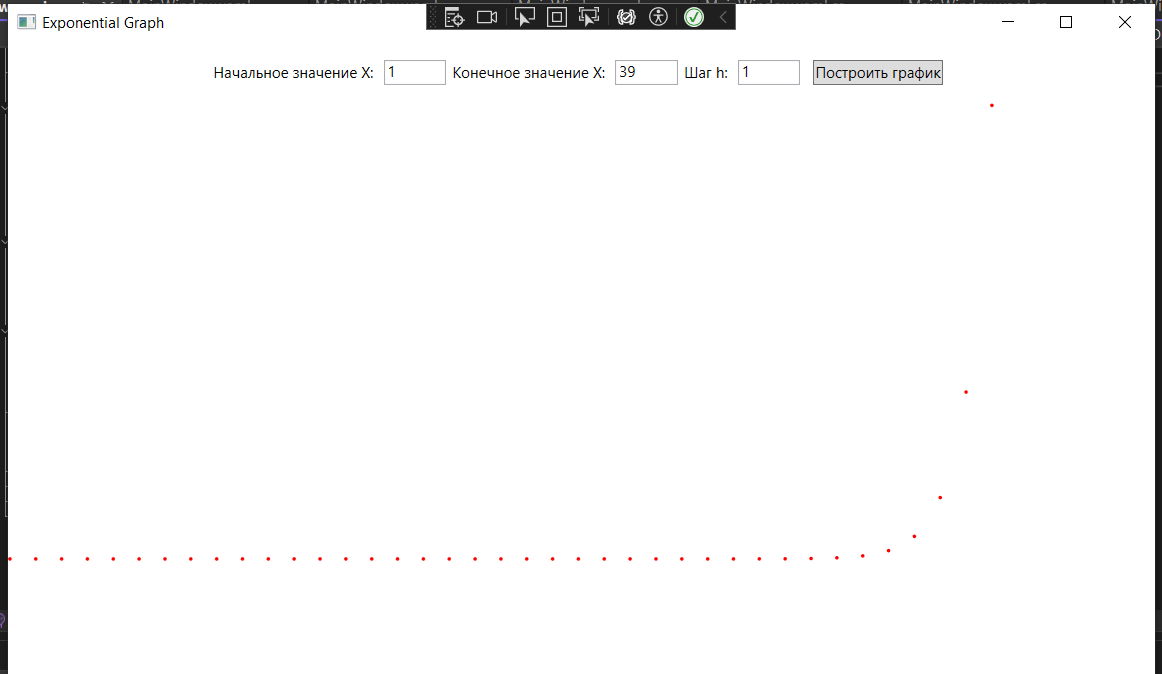


Рисунок 25.3 – Результат задания

Задание 4: Написать программу с кнопкой. При наведении курсора на элемент управления он должен создавать новую кнопку а при клике удалять себя.

Решение:

<Window x:Class="DynamicButtonsApp.MainWindow"

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

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

Title="Dynamic Buttons App" Height="450" Width="800">

<Grid Name="MainGrid" Background="LightGray">

<Button Name="InitialButton" Content="Hover over me" Width="100" Height="30"

HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10"

MouseEnter="Button\_MouseEnter" Click="Button\_Click"/>

</Grid>

</Window>

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace DynamicButtonsApp

{

public partial class MainWindow : Window

{

private Random random;

public MainWindow()

{

InitializeComponent();

random = new Random();

}

private void Button\_MouseEnter(object sender, MouseEventArgs e)

{

// Создание новой кнопки

Button newButton = new Button

{

Content = "Hover over me",

Width = 100,

Height = 30,

Margin = new Thickness(10)

};

// Установка случайного местоположения для новой кнопки

newButton.HorizontalAlignment = HorizontalAlignment.Left;

newButton.VerticalAlignment = VerticalAlignment.Top;

newButton.Margin = new Thickness(random.Next((int)(MainGrid.ActualWidth - newButton.Width)),

random.Next((int)(MainGrid.ActualHeight - newButton.Height)),

0, 0);

// Добавление обработчиков событий

newButton.MouseEnter += Button\_MouseEnter;

newButton.Click += Button\_Click;

// Добавление новой кнопки на Grid

MainGrid.Children.Add(newButton);

}

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

{

Button clickedButton = sender as Button;

if (clickedButton != null)

{

MainGrid.Children.Remove(clickedButton);

}

}

}

}

Анализ результатов:

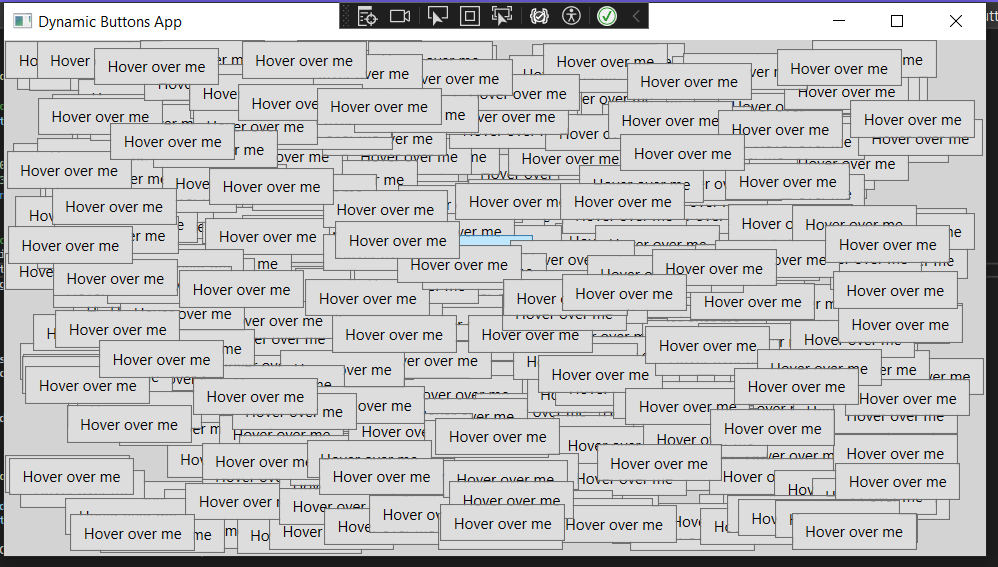


Рисунок 26.4 – Результат задания