**25.** **Технология Windows Presentation Foundation**

Задание №1

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

Листинг программы:

<Grid>

<Button Content="Перетасваемая Кнопка" HorizontalAlignment="Left" Margin="195,217,0,0" VerticalAlignment="Top"/>

<Button Content="BoubleClick" Margin="395,217,100,100"/>

<Button Content="XAML добавленная кнопка" Margin="100,100,500,300"/>

</Grid>

Таблица 25.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

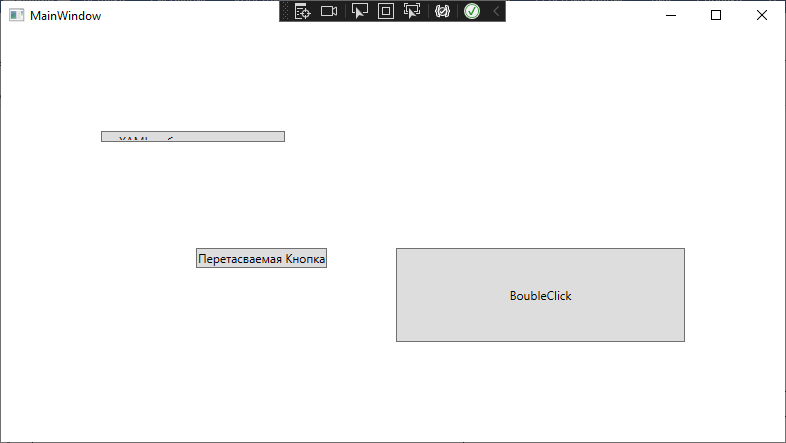


Рисунок 25.1 – Результат выполнения программы задание №1

Задание №2

Переделать задание 1 таким образом чтобы главная форма приняла следующий вид и все элементы выполняли необходимые действия.

Листинг программы:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="buttonReference"

Click="buttonReference\_Click"

Grid.Column="3"

Margin="0,5"

Content="Спаравка"/>

<Label x:Name="labelGreeting"

Grid.Column="1"

Grid.Row="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="40"

Content="Hello, world!"/>

<TextBox x:Name="textBoxName"

Grid.Column="1"

Grid.Row="2"

Grid.ColumnSpan="2"

HorizontalContentAlignment="Center"

VerticalContentAlignment="Center"

Margin="20"/>

<Button x:Name="buttonSayName"

Click="buttonSayName\_Click"

Grid.Column="0"

Grid.Row="3"

Grid.ColumnSpan="2"

Margin="80,10"

Content="Сказать имя"/>

<Button x:Name="buttonClose"

Click="buttonClose\_Click"

Grid.Column="2"

Grid.Row="3"

Grid.ColumnSpan="2"

Margin="80,10"

Content="Закрыть"/>

</Grid>

using System.Windows;

namespace z2

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

if(textBoxName.Text.Length != 0)

{

labelGreeting.Content = $"Hello, {textBoxName.Text}";

}

else

{

MessageBox.Show(

"Введите имя.",

"Пустое значение",

MessageBoxButton.OK);

}

}

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

{

this.Close();

}

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

{

ReferenceView rw = new ReferenceView();

rw.Show();

}

}

}

Таблица 25.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Антон | Hello, Атон |

Анализ результатов:

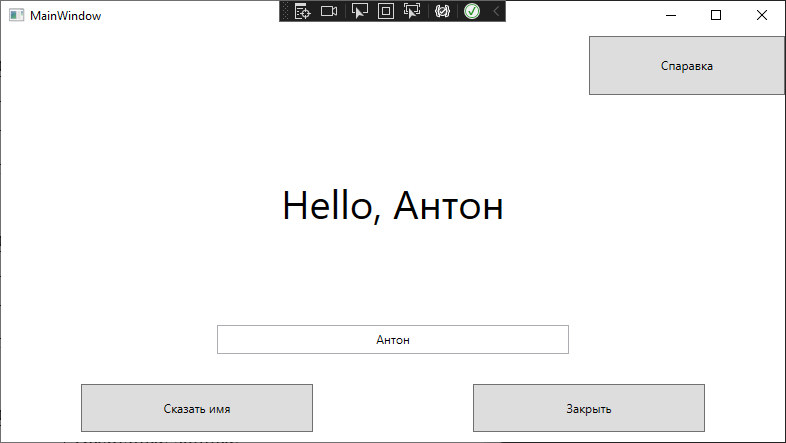


Рисунок 25.2 – Результат выполнения программы задание №2

Задание №3

Создайте проект WPF и выполните решение следующего алгоритма. Элементы управления Button TextBlock, TextBox. Расположение элементов продумайте самотоятельно. Построить график функции. Таблицу данных получить путём изменения параметра X с шагом h. Самостоятельно выбрать удобные параметры настройки.

Листинг программы:

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<StackPanel>

<Label Grid.Row="0" Content="Введите min x: "/>

<TextBox x:Name="textBoxMinX"/>

<Label Grid.Row="0" Content="Введите max x: "/>

<TextBox x:Name="textBoxMaxX"/>

<Button x:Name="buttonCaculate"

Click="buttonCaculate\_Click"

Content="Построить график"/>

</StackPanel>

<lvc:CartesianChart x:Name="chart" Grid.Column="1" >

<lvc:CartesianChart.AxisY>

<lvc:Axis Name="Axis">

<lvc:Axis.Sections>

<lvc:AxisSection>

<lvc:AxisSection.Fill>

<SolidColorBrush Color="#A3A3FF" Opacity=".4"></SolidColorBrush>

</lvc:AxisSection.Fill>

</lvc:AxisSection>

</lvc:Axis.Sections>

</lvc:Axis>

</lvc:CartesianChart.AxisY>

</lvc:CartesianChart>

</Grid>

using LiveCharts.Defaults;

using LiveCharts.Wpf;

using LiveCharts;

using System;

using System.Windows;

using System.Windows.Media;

namespace z3

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

chart.Series = new SeriesCollection { new LineSeries

{

Values = new ChartValues<ObservablePoint>(),

PointGeometrySize = 0,

StrokeThickness = 4,

Fill = Brushes.Transparent

}

};

DataContext = this;

}

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

{

try

{

int minX = Convert.ToInt32(textBoxMinX.Text);

int maxX = Convert.ToInt32(textBoxMaxX.Text);

if (minX < maxX)

{

ChartValues<ObservablePoint> points = new ChartValues<ObservablePoint>();

for (int i = minX; i <= maxX; i++)

{

if (i != 0)

{

points.Add(new ObservablePoint(i, 1 / Math.Tan(i)));

}

}

chart.Series = new SeriesCollection { new LineSeries

{

Values = points,

PointGeometrySize = 0,

StrokeThickness = 4,

Fill = Brushes.Transparent

}

};

}

}

catch (FormatException)

{

MessageBox.Show(

"Неправильные данные введенные в поля",

"Неправильный формат",

MessageBoxButton.OK);

}

}

}

}

Таблица 25.3 – Входные и выходные данные программы задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Min x: 1  Max x: 20 |  |

Анализ результатов:

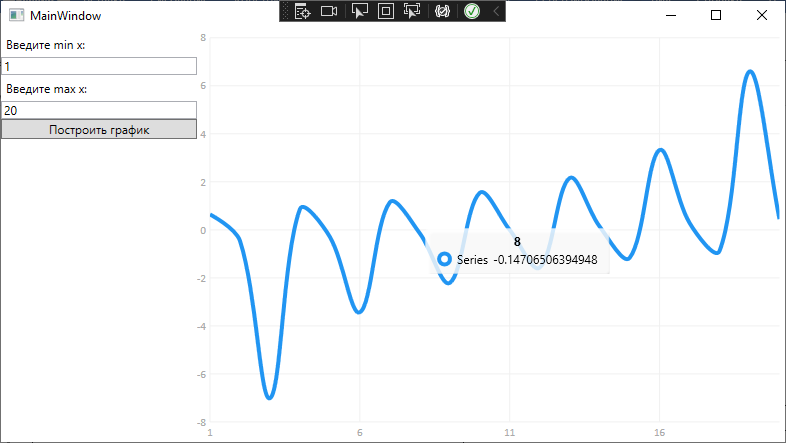


Рисунок 25.1 – Результат выполнения программы задание №3

Задание №4

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

Листинг программы:

<Grid >

<Canvas x:Name="c">

<Button PreviewMouseMove="Button\_PreviewMouseMove" Width="100" Height="20" Click="Button\_Click"/>

</Canvas>

</Grid>

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace z4

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

Button bt = new Button();

bt.Width = 100;

bt.Height = 20;

Random r = new Random();

bt.Margin = new Thickness(r.Next(0,1000), r.Next(0, 1000),0,0);

bt.PreviewMouseMove += Button\_PreviewMouseMove;

bt.Click += Button\_Click;

c.Children.Add(bt);

}

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

{

Button bt = (Button)sender;

c.Children.Remove(bt);

}

}

}

Анализ результатов:

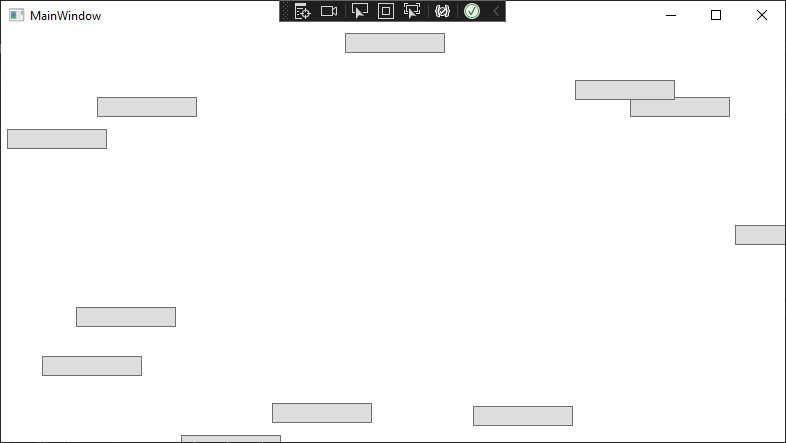


Рисунок 25.1 – Результат выполнения программы задание №4