1. **Технология Windows Presentation Foundation**

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

Листинг программы:

namespace Task\_1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

buttonRun.Click += ButtonRun\_Click;

buttonAbout.Click += ButtonAbout\_Click;

}

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

{

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

AboutApp aboutApp = new AboutApp(info);

aboutApp.ShowDialog();

}

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

{

string hello = "Hello";

string input = string.Empty;

if (string.IsNullOrEmpty(textBoxEnterName.Text) || string.IsNullOrWhiteSpace(textBoxEnterName.Text))

{

input = "World";

}

else

{

input = textBoxEnterName.Text;

}

textBlockHello.Text = $"{hello} {input}";

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Slava | Hello Slava |

Анализ результатов:

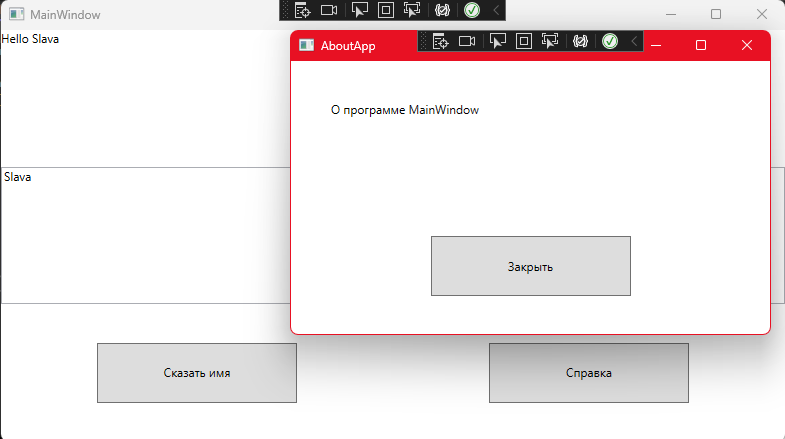


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

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».

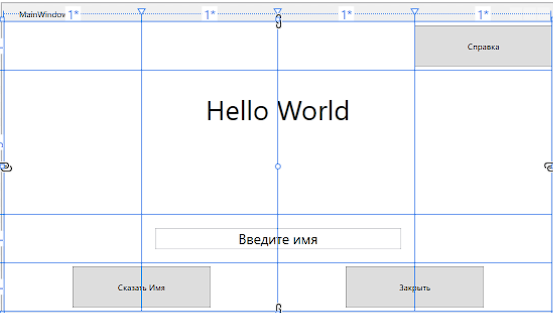


Рисунок 2.1 – Задание

Источник: собственная разработка

Листинг программы:

namespace Task\_2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

buttonRun.Click += ButtonRun\_Click;

buttonClose.Click += ButtonClose\_Click;

buttonAbout.Click += ButtonAbout\_Click;

textBoxEnterName.MouseEnter += TBMouse\_Enter;

}

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

{

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

AboutApp aboutApp = new AboutApp(info);

aboutApp.ShowDialog();

}

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

{

this.Close();

}

private void TBMouse\_Enter(object sender, RoutedEventArgs e)

{

textBoxEnterName.Text = "";

}

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

{

string hello = "Hello";

string input = string.Empty;

if (string.IsNullOrEmpty(textBoxEnterName.Text) || string.IsNullOrWhiteSpace(textBoxEnterName.Text) || textBoxEnterName.Text == "Введите имя")

{

input = "World";

textBoxEnterName.Text = "Введите имя";

}

else

{

input = textBoxEnterName.Text;

}

textBlockHello.Text = $"{hello} {input}";

}

}

}

Таблица 2.1 – Входные и выходные данные

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

Анализ результатов:

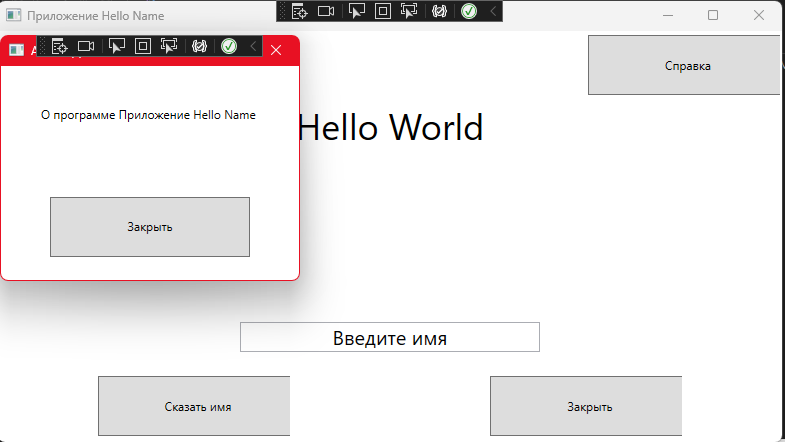


Рисунок 2.2 – Результат работы программы

Источник: собственная разработка

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. y=|x|.

Листинг программы:

namespace Task\_3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

buttonRun.Click += ButtonRun\_Click;

textBoxEnterX.MouseEnter += TBMouse\_Enter;

textBoxEnterH.MouseEnter += TBMouse\_EnterH;

textBoxEnterXN.MouseEnter += TBMouse\_EnterXN;

}

private void TBMouse\_Enter(object sender, RoutedEventArgs e)

{

textBoxEnterX.Text = "";

}

private void TBMouse\_EnterH(object sender, RoutedEventArgs e)

{

textBoxEnterH.Text = "";

}

private void TBMouse\_EnterXN(object sender, RoutedEventArgs e)

{

textBoxEnterXN.Text = "";

}

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

{

if (!string.IsNullOrEmpty(textBoxEnterX.Text) || !string.IsNullOrWhiteSpace(textBoxEnterX.Text) && textBoxEnterX.Text != "Введите X" &&

!string.IsNullOrEmpty(textBoxEnterX.Text) || !string.IsNullOrWhiteSpace(textBoxEnterX.Text) && textBoxEnterX.Text != "Введите H" &&

!string.IsNullOrEmpty(textBoxEnterX.Text) || !string.IsNullOrWhiteSpace(textBoxEnterX.Text) && textBoxEnterX.Text != "Введите XN")

{

double inputX = double.Parse(textBoxEnterX.Text);

double inputH = double.Parse(textBoxEnterH.Text);

double inputXN = double.Parse(textBoxEnterXN.Text);

double y = 0;

for (double i = inputX; i < inputXN; i = i + inputH)

{

y = Math.Abs(i);

textBlockHello.Text += $"y = {y}\n";

}

}

}

}

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -9 0,5 4 | 9 8,5 8 7,5 7 6,5 6 5,5 5 4,5 4 3,5 3 2,5 2 1,5 1 0 0,5 1 1,5 2 2,5 3 3,5 |

Анализ результатов:

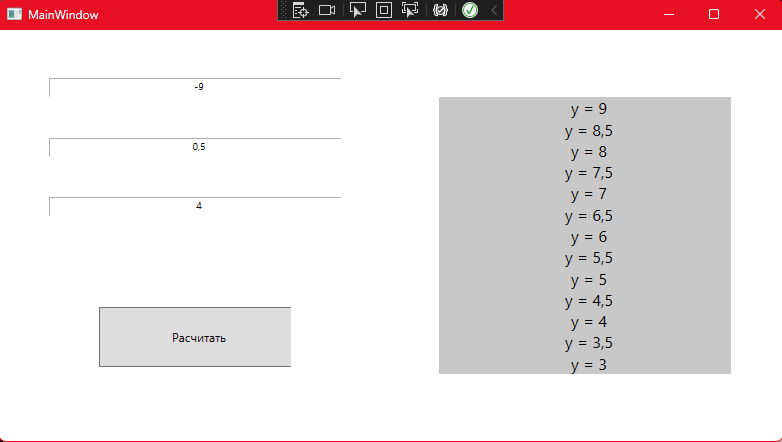


Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

namespace Task\_4

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Random random = new Random();

public MainWindow()

{

InitializeComponent();

}

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

{

Button button = new Button();

button.Content = "New Button";

button.Width = 100;

button.Height = 30;

button.Margin = new Thickness(10);

Grid.SetRow(button, random.Next(5));

Grid.SetColumn(button, random.Next(5));

Grid grid = (Grid)this.Content;

grid.Children.Add(button);

button.Click += (s, ev) => { myGrid.Children.Remove(button); };

}

}

}

Таблица 4.1 – Входные и выходные данные

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

Анализ результатов:

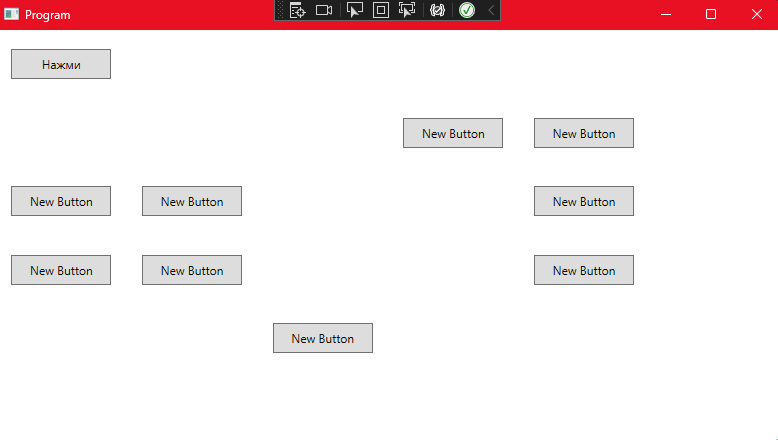


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка