Сборки, библиотеки, атрибуты, директивы

Задание №1. Изучить теоретический материал. Выполнить на практике все примеры создания WPF приложения.

Задание №2. Переделать задание 1 таким образом чтобы главная форма приняла следующий вид и все элементы выполняли необходимые действия.

Анализ программы:

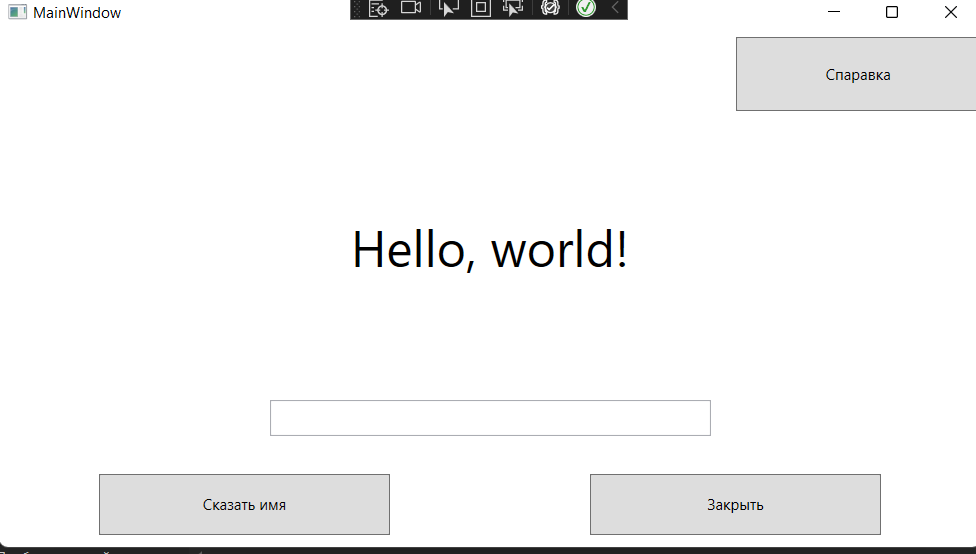


Рисунок 1.1 - Анализ программы

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.

Листинг программы:

using LiveCharts.Defaults;

using LiveCharts.Wpf;

using LiveCharts;

using System;

using System.Windows;

using System.Windows.Media;

namespace z2

{

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);

int a = 1;

int b = 100;

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.Pow(i, 3)));

}

}

chart.Series = new SeriesCollection { new LineSeries

{

Values = points,

PointGeometrySize = 0,

StrokeThickness = 4,

Fill = Brushes.Transparent

}

};

}

}

catch (FormatException)

{

MessageBox.Show(

"Неправильные данные!",

"Неправильный формат!",

MessageBoxButton.OK);

}

}

}

}

Анализ программы:

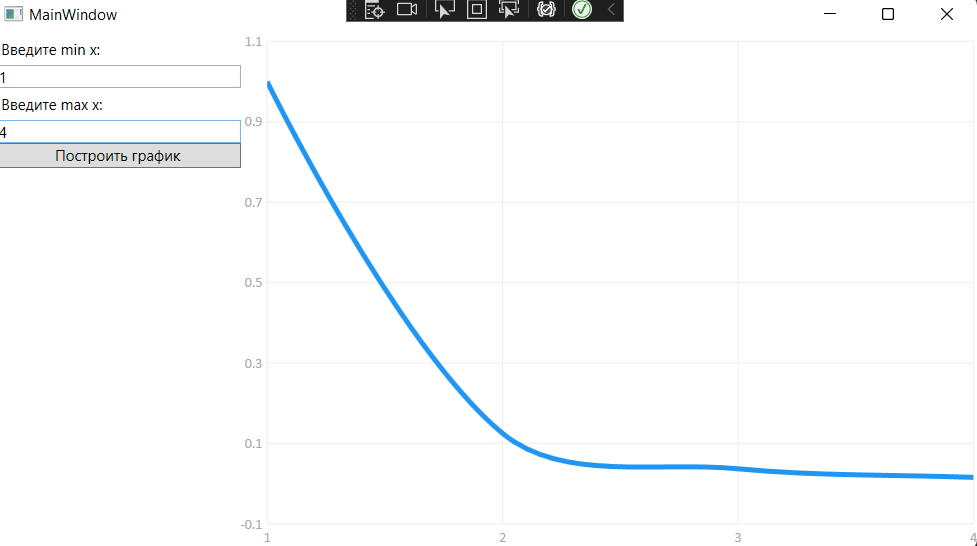


Рисунок 1.2 - Анализ программы

Источник: собственная разработка