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

Задание 1. 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:

namespace Task3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window, INotifyPropertyChanged

{

private double xMin;

private double xMax;

private double h;

private SeriesCollection seriesCollection;

public event PropertyChangedEventHandler PropertyChanged;

public ChartValues<ObservablePoint> DataPoints { get; set; }

public double XMin

{

get => xMin;

set

{

xMin = value;

OnPropertyChanged("XMin");

}

}

public double XMax

{

get => xMax;

set

{

xMax = value;

OnPropertyChanged("XMax");

}

}

public double H

{

get => h;

set

{

h = value;

OnPropertyChanged("H");

}

}

public MainWindow()

{

InitializeComponent();

DataContext = this;

DataPoints = new ChartValues<ObservablePoint>();

}

private void OnPropertyChanged(string propertyName)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));

}

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

{

double xMinValue, xMaxValue, hValue;

if (!double.TryParse(TextBoxXMin.Text, out xMinValue) ||

!double.TryParse(TextBoxXMax.Text, out xMaxValue) ||

!double.TryParse(TextBoxH.Text, out hValue))

{

MessageBox.Show("Invalid input values!");

return;

}

if (xMinValue >= xMaxValue || hValue <= 0)

{

MessageBox.Show("Invalid input values!");

return;

}

XMin = xMinValue;

XMax = xMaxValue;

H = hValue;

// Clear previous data points

DataPoints.Clear();

// Generate new data points

for (double x = XMin; x <= XMax; x += H)

{

double y = Math.Sqrt(x);

DataPoints.Add(new ObservablePoint(x, y));

}

}

}

}

Анализ результатов:

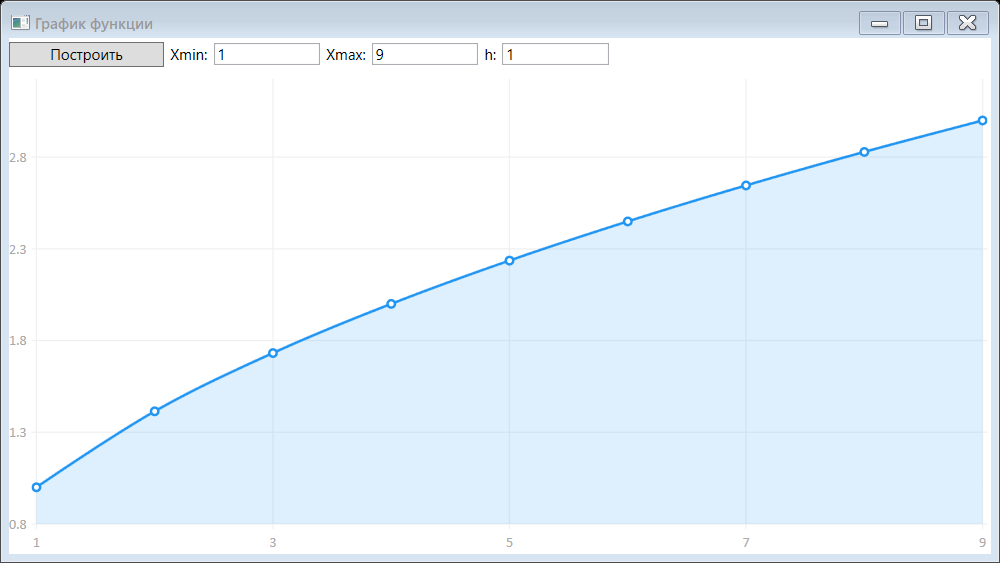


Рисунок 25.1 – Результат работы программы

Задание 2. Написать программу с кнопкой. При наведении курсора на элемент управления он должен создавать новую кнопку а при клике удалять себя.

Листинг программы:

namespace Task4

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

Button button = (Button)sender;

// Создаем новую кнопку

Button newButton = new Button

{

Content = "Нажмите на меня",

Width = 130,

Height = 30

};

// Устанавливаем обработчик события клика на новую кнопку

newButton.Click += NewButton\_Click;

// Добавляем новую кнопку в StackPanel

stackPanel.Children.Add(newButton);

}

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

{

Button button = (Button)sender;

// Удаляем кнопку из StackPanel

stackPanel.Children.Remove(button);

}

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

{

}

}

}

Анализ результатов:

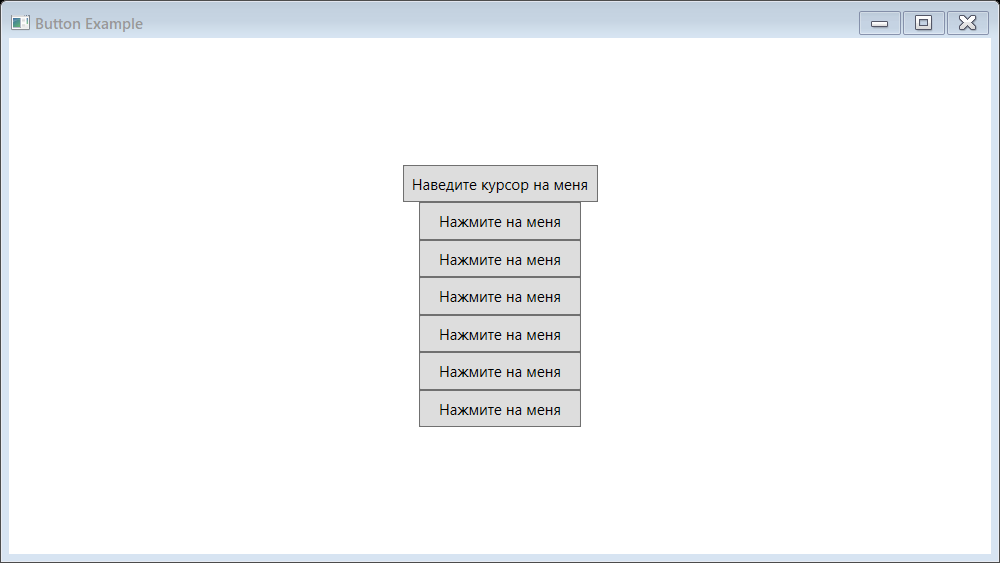


Рисунок 25.2 – Результат работы программы