**Технология WINDOWS PRESENTATION FOUNDATION**

**Задание №1-2.** Изучить теоретический материал. Выполнить на практике все примеры создания WPF приложения. Переделать задание 1 таким образом чтобы главная форма приняла следующий вид и все элементы выполняли необходимые действия.

Листинг программы:

public MainWindow()

{

InitializeComponent();

buttonRun.Click += ButtonRun\_Click;

buttonAbout.Click += ButtonAbout\_Click;

buttonClose.Click += ButtonClose\_Click;

textBoxEnterName.GotFocus += TextBoxEnterName\_GotFocus;

}

private void TextBoxEnterName\_GotFocus(object sender, RoutedEventArgs e)

{

var txt = (TextBox)sender;

if (txt.Text.Contains("Введите"))

{

txt.Text = string.Empty;

}

}

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

{this.Close();}

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}!";

}

Анализ результатов:

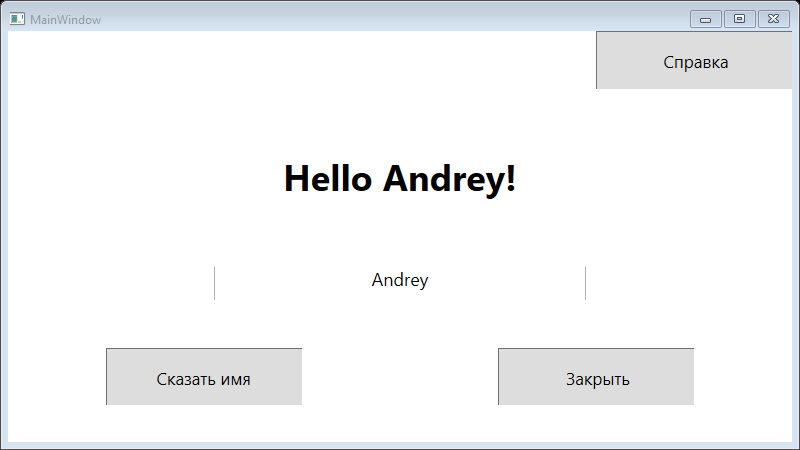


Рисунок 1.1 – Результат работы программы

**Задание №3.** Создайте проект WPF и выполните решение следующего алгоритма. Элементы управления Button TextBlock, TextBox. Расположение элементов продумайте самоcтоятельно.Построить график функции. Таблицу данных получить путём изменения параметра X с шагом h. Самостоятельно выбрать удобные параметры настройки.

Листинг программы:

public MainWindow()

{

InitializeComponent();

LineSeries = new LineSeries();

SeriesCollection = new SeriesCollection

{

new LineSeries

{

Values = new ChartValues<ObservableValue>

{

new ObservableValue(3),

new ObservableValue(4),

new ObservableValue(6),

new ObservableValue(8),

new ObservableValue(7),

new ObservableValue(5)

},

PointGeometrySize = 0,

StrokeThickness = 4,

Fill = Brushes.Transparent

}

};

DataContext = this;

}

public LineSeries LineSeries { get; set; }

public SeriesCollection SeriesCollection { get; set; }

private void UpdateAllOnClick(object sender, RoutedEventArgs e)

{

try

{

var x0 = int.Parse(textBoxEnterX0.Text);

var xn = int.Parse(textBoxEnterXn.Text);

var h = int.Parse(textBoxEnterH.Text);

var r = new Random();

foreach (var series in SeriesCollection)

{

var i = x0;

foreach (var observable in series.Values.Cast<ObservableValue>())

{

if (i >= xn)

{

break;

}

observable.Value = Math.Pow(i, 2);

i += h;

}

}

}

catch (FormatException)

{

}

Анализ результатов:

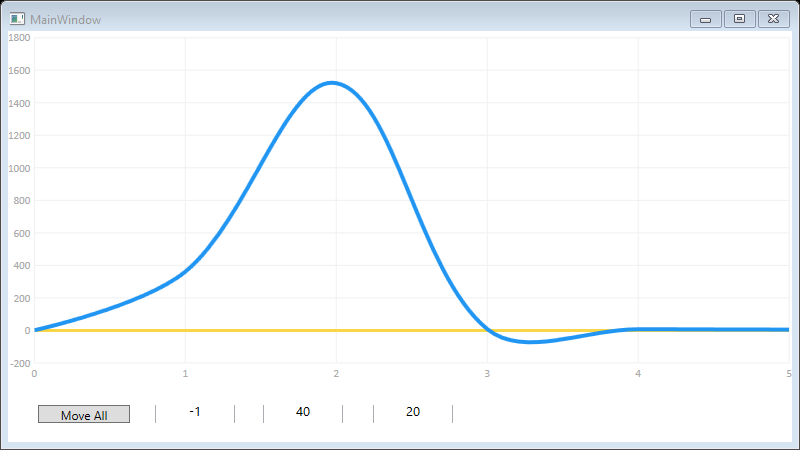


Рисунок 1.2 – Результат работы программы