**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине: «Объектно-ориентированное проектирование и программирование»

на тему: **«**Делегаты и события. Создание своих событий**»**

Выполнил: студент гр. ИТИ-21

Бахонько А. М.

Принял: преподаватель

Башаримов Ю. С.

Гомель 2024

**Цель работы**: изучить работу с делегатами; научится создавать собственные события.

**Задание:**

• Необходимо разработать свой класс, в который нужно добавить событие, в соответствии с заданием.

• Создать *WPF* приложение для созданного класса.

• События должны добавляться методом, реализующий делегат.

|  |  |
| --- | --- |
| № | Условие |
| 2 | Разработать класс «Интеграл» который вычисляет интеграл любым численным методом. Функция передается в качестве аргумента. Для передачи функции объявить свой собственный делегат. |

**Ход работы**

На рисунке 1 представлено окно калькулятора интегралов в приложении *WPF*.

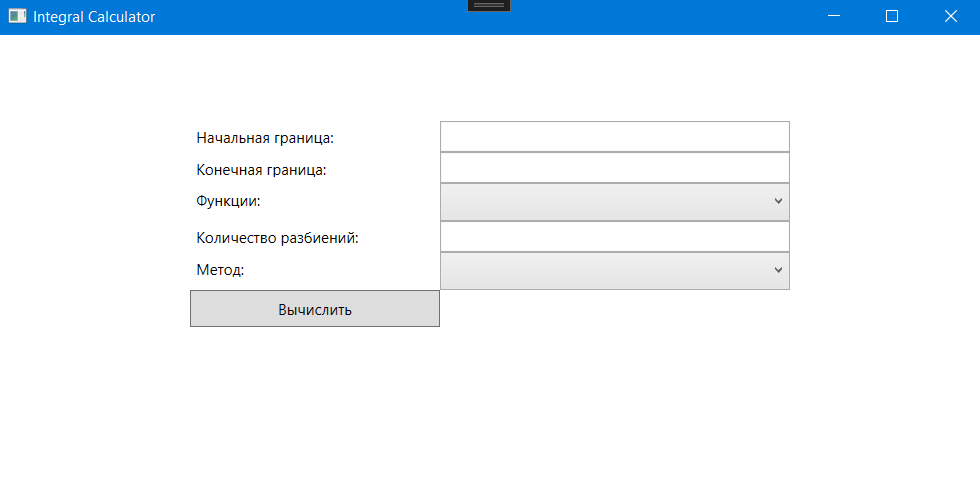


Рисунок 1 – Калькулятор интегралов

В калькуляторе есть поля для ввода интервала интегрирования и количества разбиений, выпадающий список для выбора метода подсчёта интеграла, кнопка, что реализует событие с помощью метода, реализующий делегат. Результат нажатия на кнопку «Выполнить**»** представлен на рисунке 2.

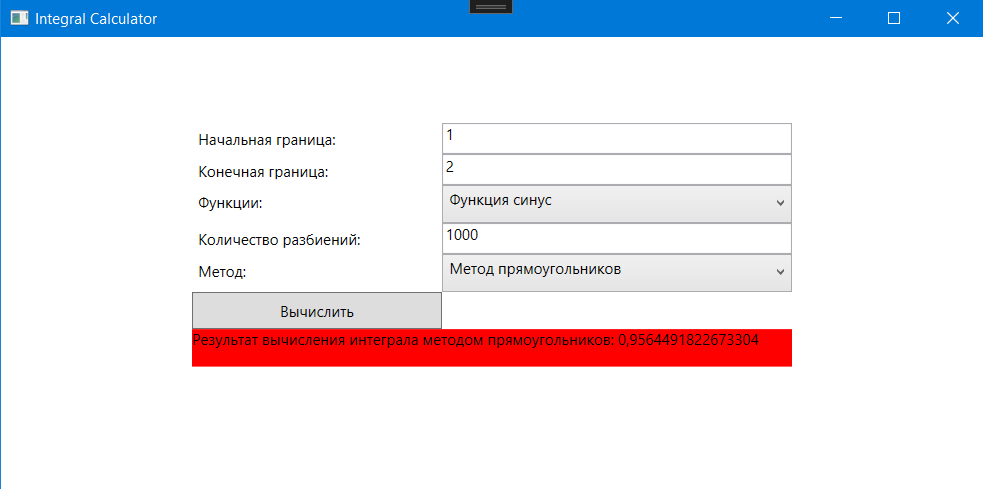


Рисунок 2 – Результат подсчёта методом прямоугольников

После выбора метода трапеции и нажатия на кнопку «Выполнить**»** в окне будет выведен новый результат. Результат представлен на рисунке 3.

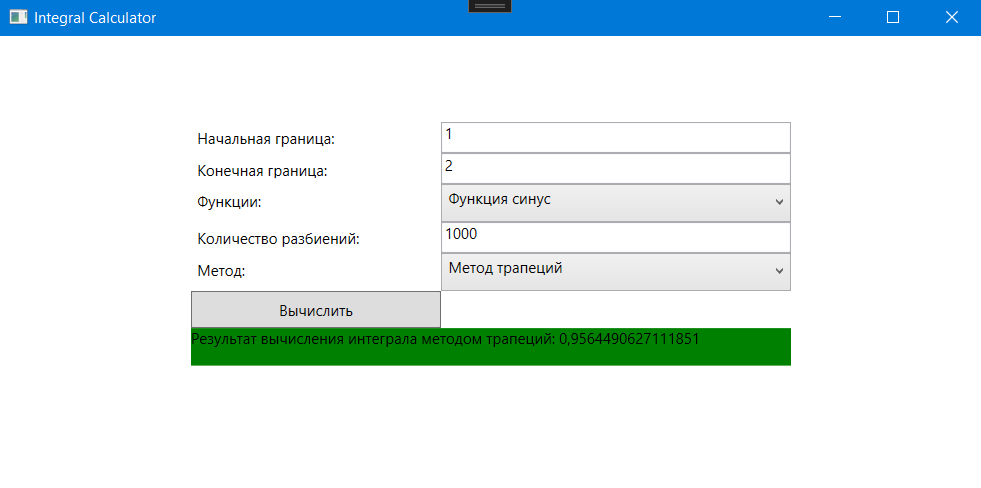


Рисунок 3 – Результат подсчёта методом трапеций

В приложении А представлен код программы.

На рисунке 4 представлен репозиторий проекта.

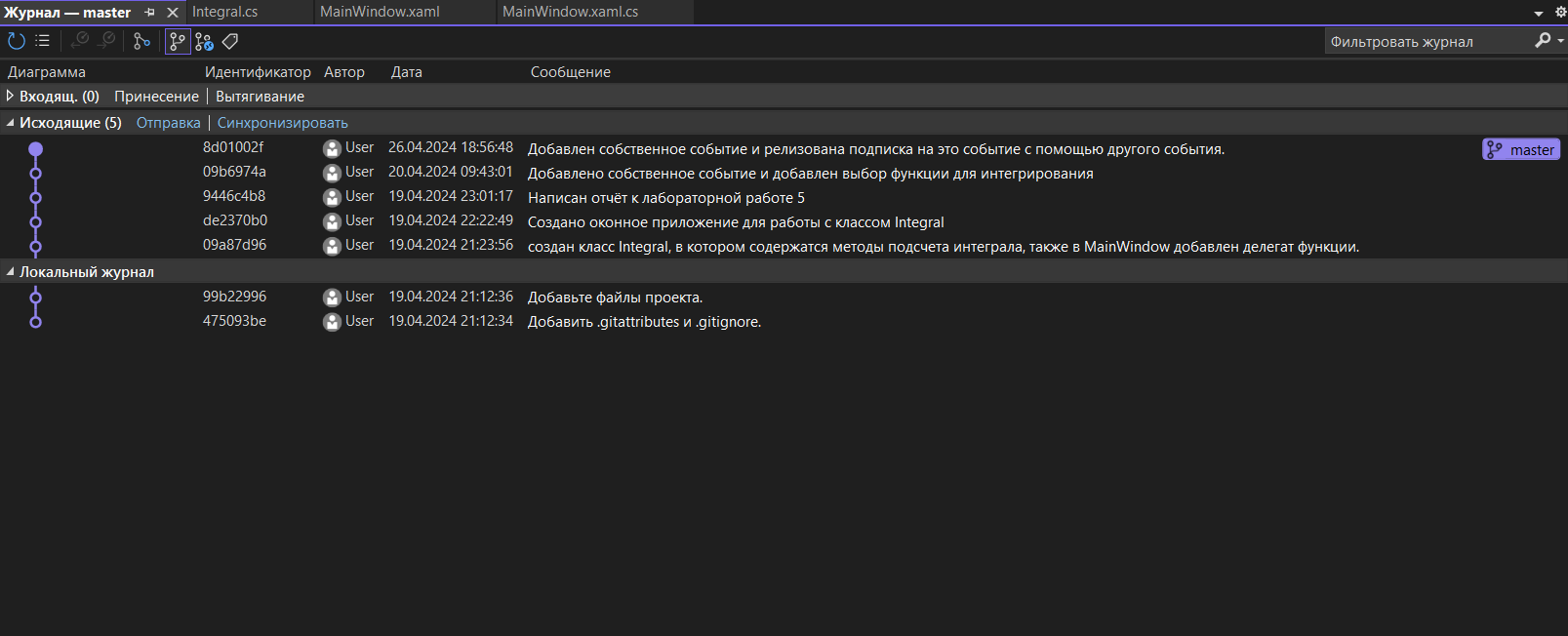


Рисунок 4 – Репозиторий проекта

**Вывод:** в результате выполнения лабораторной работы создали приложение *WPF*; реализовали делегаты для методов; осуществили работу собственных событий.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**Integral.cs**

namespace IntegralCalculator

{

// класс Integral с методами для вычисления интеграла

public class Integral

{

public event EventHandler СalculationIntegral;

// методы для вычисления интеграла

public double RectangleMethod(Function f, double a, double b, int n)

{

double h = (b - a) / n;

double result = 0.0;

for (int i = 0; i < n; i++)

{

result += f(a + h \* (i + 0.5));

}

result \*= h;

OnCalculationIntegral();

return result;

}

public double TrapezoidMethod(Function f, double a, double b, int n)

{

double h = (b - a) / n;

double result = 0.0;

for (int i = 1; i < n; i++)

{

result += f(a + h \* i);

}

result += (f(a) + f(b)) / 2;

result \*= h;

OnCalculationIntegral();

return result;

}

protected virtual void OnCalculationIntegral()

{

СalculationIntegral?.Invoke(this, EventArgs.Empty);

}

}

}

**MainWindow.xaml**

<Window x:Class="IntegralCalculator.MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Integral Calculator" Height="400" Width="800">

<Grid HorizontalAlignment="Center" VerticalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="280"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="25"/>

<RowDefinition Height="25"/>

<RowDefinition Height="25"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<!-- элементы для ввода данных -->

<Label Grid.Row="0" Grid.Column="0" Content="Начальная граница:"/>

<TextBox x:Name="textBoxA" Grid.Row="0" Grid.Column="1"/>

<Label Grid.Row="1" Grid.Column="0" Content="Конечная граница:"/>

<TextBox x:Name="textBoxB" Grid.Row="1" Grid.Column="1"/>

<Label Grid.Row="2" Grid.Column="0" Content="Количество разбиений:"/>

<TextBox x:Name="textBoxN" Grid.Row="2" Grid.Column="1"/>

<!-- выпадающий список для выбора метода -->

<Label Grid.Row="3" Grid.Column="0" Content="Метод:"/>

<ComboBox x:Name="comboBoxMethod" Grid.Row="3" Grid.Column="1">

</ComboBox>

<!-- кнопка для вычисления интеграла -->

<Button x:Name="buttonCalculate" Grid.Row="4" Grid.Column="0" Content="Вычислить" Click="buttonCalculate\_Click"/>

<!-- текстовая метка для вывода результата -->

<TextBlock x:Name="labelResult" Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="2"/>

</Grid>

</Window>

**MainWindow.xaml.cs**

using System.Windows;

using System.Windows.Media;

namespace IntegralCalculator

{

// объявление делегата

public delegate double Function(double x);

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

// добавление элементов в выпадающий список

comboBoxMethod.Items.Add("Метод прямоугольников");

comboBoxMethod.Items.Add("Метод трапеций");

comboBoxFunc.Items.Add("Функция синус");

comboBoxFunc.Items.Add("Функция косинус");

comboBoxFunc.Items.Add("Функция тангенс");

}

// обработчик нажатия кнопки "Вычислить"

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

{

try

{

// получение введенных пользователем данных

double a = Convert.ToDouble(textBoxA.Text);

double b = Convert.ToDouble(textBoxB.Text);

int n = Convert.ToInt32(textBoxN.Text);

// задание функции

Function f = Math.Sin;

if (comboBoxFunc.SelectedItem.ToString() == "Функция синус")

{

f = Math.Sin;

}

if (comboBoxFunc.SelectedItem.ToString() == "Функция косинус")

{

f = Math.Cos;

}

if (comboBoxFunc.SelectedItem.ToString() == "Функция тангенс")

{

f = Math.Tan;

}

// создание экземпляра класса Integral

Integral integral = new Integral();

integral.СalculationIntegral += Integral\_Methods;

double result = 0.0;

// проверка выбранного метода и вызов соответствующего метода класса Integral

if (comboBoxMethod.SelectedItem.ToString() == "Метод прямоугольников")

{

result = integral.RectangleMethod(f, a, b, n);

labelResult.Text = $"Результат вычисления интеграла методом прямоугольников: {result}";

}

else if (comboBoxMethod.SelectedItem.ToString() == "Метод трапеций")

{

result = integral.TrapezoidMethod(f, a, b, n);

labelResult.Text = $"Результат вычисления интеграла методом трапеций: {result}";

}

}

catch (FormatException)

{

MessageBox.Show("Ошибка ввода. Проверьте правильность введенных значений.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch (OverflowException)

{

MessageBox.Show("Ошибка ввода. Введено слишком большое значение в поле.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch (Exception ex)

{

MessageBox.Show($"Произошла ошибка: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void Integral\_Methods(object sender, EventArgs e)

{

if (comboBoxMethod.SelectedItem.ToString() == "Метод прямоугольников")

labelResult.Background = Brushes.Red;

else if (comboBoxMethod.SelectedItem.ToString() == "Метод трапеций")

labelResult.Background = Brushes.Green;

}

}

}