**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

Тарканов А. В.

Принял: преподаватель

Башаримов Ю. С.

Гомель 2024

**Цель работы**: необходимо разработать свой класс, в который нужно добавить событие. В соответствии с заданием.

|  |  |
| --- | --- |
| № | 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 главное окно приложения.

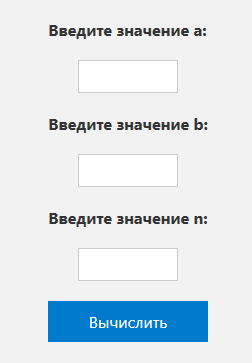


Рисунок 1 – Главное меню

В пользовательском интерфейсе мы видим, что у нас есть три поля для ввода значений для вычисления определенного интеграла, а также кнопка вычислить, при нажатии на которую вычисляется значение интеграла.

Идея метода заключается в разбиении области интегрирования на небольшие отрезки и аппроксимации каждого отрезка трапецией. Затем площади этих трапеций суммируются, чтобы получить приближенное значение интеграла.

Шаги метода трапеций:

1.Разбейте интервал интегрирования на n равных частей (отрезков).

2.Для каждого отрезка, используйте значения функции на концах отрезка для аппроксимации трапеции.

3.Вычислите площадь каждой трапеции.

4.Сложите площади всех трапеций, чтобы получить приближенное значение интеграла.

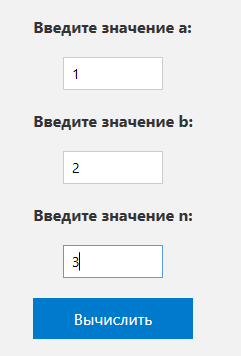


Рисунок 2 – Ввод значений

После нажатия на кнопку вычислить у нас вычисляется интеграл и выводится на экран ответ решения данного интеграла.

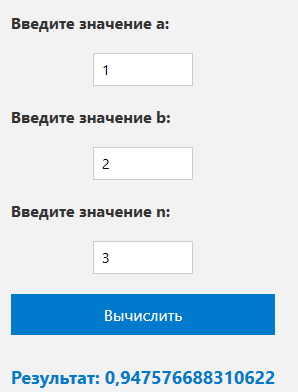


Рисунок 3 – Вычисление

При неправильном формате ввода отработает исключение, которое не допускает неправильный формат ввода в поля ввода, например строку. Это можно увидеть на рисунке 4.

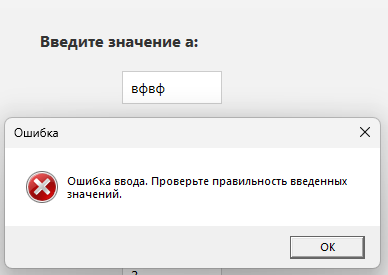


Рисунок 4 – Ошибка ввода

На рисунке 5 представлены результаты *unit*-тестов.

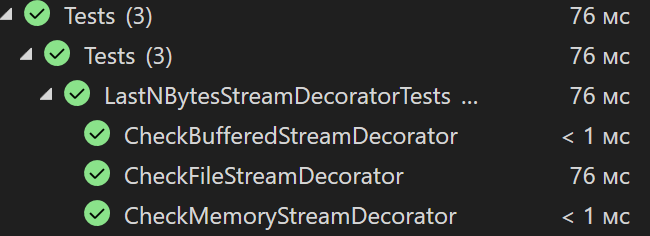


Рисунок 5 – Тестирование

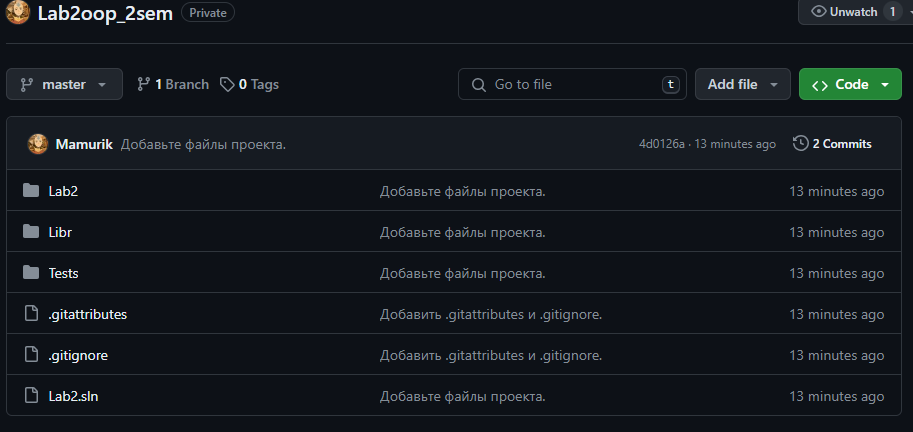


Рисунок 6 – *Github*

В приложении А представлен код программы.

**Вывод:** в результате выполнения лабораторной работы разработали свой класс, в который нужно добавили событие. Вычисляющее интегралл.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**MainForm.cs**

using System;

using System.Windows;

namespace Lab5oop

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

/// <summary>

/// Обработчик события нажатия кнопки CalculateButton.

/// </summary>

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

{

try

{

// Получаем значения из TextBox'ов

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

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

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

// Создаем экземпляр класса Integral

Integral integral = new Integral();

// Создаем делегат и передаем функцию

FunctionDelegate function = Math.Sin;

// Вычисляем интеграл

double result = integral.Calculate(function, a, b, n);

// Выводим результат

ResultTextBlock.Text = $"Результат: {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.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch (Exception ex)

{

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

}

}

}

public delegate double FunctionDelegate(double x);

}

**Integral.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5oop

{

/// <summary>

/// Класс для вычисления интеграла.

/// </summary>

public class Integral

{

/// <summary>

/// Метод для вычисления интеграла с использованием метода трапеций.

/// </summary>

/// <param name="function">Функция, подынтегральное выражение.</param>

/// <param name="a">Нижний предел интегрирования.</param>

/// <param name="b">Верхний предел интегрирования.</param>

/// <param name="n">Количество разбиений.</param>

/// <returns>Значение интеграла.</returns>

public double Calculate(FunctionDelegate function, double a, double b, int n)

{

double h = (b - a) / n;

double sum = 0;

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

{

double x0 = a + i \* h;

double x1 = a + (i + 1) \* h;

double y0 = function(x0);

double y1 = function(x1);

sum += (y0 + y1) \* h / 2;

}

return sum;

}

}

}

**MainWindow.Xaml:**

<Window x:Class="Lab5oop.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Lab5oop"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Window.Resources>

<Style TargetType="TextBlock">

<Setter Property="Margin" Value="10"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Foreground" Value="#333333"/>

<Setter Property="FontSize" Value="16"/>

</Style>

<Style TargetType="TextBox">

<Setter Property="Margin" Value="10"/>

<Setter Property="Width" Value="100"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="BorderBrush" Value="#CCCCCC"/>

<Setter Property="Padding" Value="6"/>

</Style>

<Style TargetType="Button">

<Setter Property="Margin" Value="10"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="Padding" Value="10"/>

<Setter Property="Background" Value="#007ACC"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Cursor" Value="Hand"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#005F8C"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="TextBlock" x:Key="ResultTextBlockStyle">

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Foreground" Value="#007ACC"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="Margin" Value="10,20,10,0"/>

</Style>

</Window.Resources>

<Grid Background="#F2F2F2">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Введите значение a:"/>

<TextBox x:Name="AValueTextBox"/>

<TextBlock Text="Введите значение b:"/>

<TextBox x:Name="BValueTextBox"/>

<TextBlock Text="Введите значение n:"/>

<TextBox x:Name="NValueTextBox"/>

<Button Content="Вычислить" Click="CalculateButton\_Click"/>

<TextBlock x:Name="ResultTextBlock" Style="{StaticResource ResultTextBlockStyle}"/>

</StackPanel>

</Grid>

</Window>