**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Дисциплина «Объектно-ориентированное проектирование и программирование»

ОТЧЕТ

Лабораторная работа № 5

Делегаты и события. Создание своих событий

Выполнил студент

Группы ИТД-21

Чайдаков И.М.

Принял преподаватель

Башаримов Ю. С.

Гомель 2025

Делегаты и события. Создание своих событий

**Цель работы:** Создание консольного приложения на C# для выполнения арифметических операций, заданных в текстовом файле.

**Задание:** Необходимо разработать свой класс, в который нужно добавить событие. Используя коллекцию Dictionary сохранить по ключам (+,-,\*,/) функции выполняющие соответствующие операции. Реализовать класс калькулятора, который считывает построчно файл, и выполняет операции в зависимости от символа. Пример ввода: 123 + 234 434 \* 555 123 / 324 Данные в файле корректные. Рекомендуется использовать регулярные выражения для разбора файла.

**Ход выполнения задания:**

**1. Разработка логики вычислений**

Был создан класс Calculator, в котором используется коллекция Dictionary<char, Func<double, double, double>> для хранения арифметических операций. Каждой операции (+, -, \*, /) соответствует делегат Func, выполняющий соответствующее вычисление между двумя числами.

**2. Чтение и обработка файла**

Метод ProcessFile читает входной файл построчно. Для разбора каждой строки используется регулярное выражение (\d+)\s\*([+\-\*/])\s\*(\d+), которое извлекает два операнда и оператор. Если оператор присутствует в словаре, вычисляется результат и выводится на экран. Программа рассчитана на корректный ввод, как указано в ТЗ.

**3.** **Тестирование и демонстрация результата**

Был создан текстовый файл input.txt с несколькими выражениями. После запуска программа построчно вывела результаты вычислений (Рисунок 1).

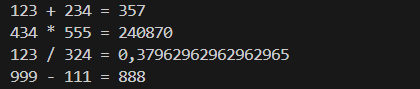


Рисунок 1 — Результат работы программы.

**4.** **Создание удаленного репозитория**

Создаем удаленный репозиторий в который отправляем файлы проетк (Рисунок 2).

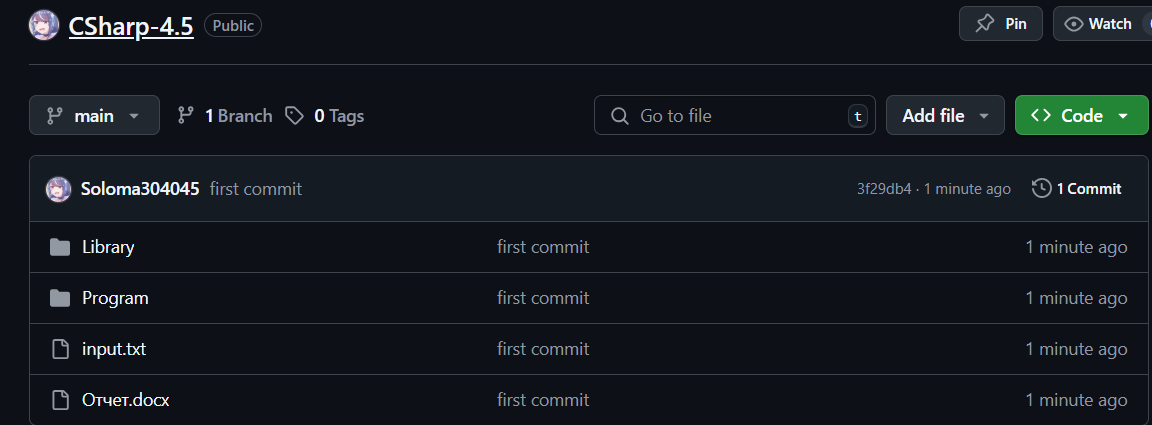


Рисунок 2 — Удаленный репозиторий.

**Вывод:** В ходе работы над проектом были освоены ключевые технологии языка C#, включая использование делегатов, коллекций Dictionary, регулярных выражений и работы с файлами. Применение этих инструментов позволило реализовать модульную и расширяемую архитектуру калькулятора, способного обрабатывать арифметические выражения из текстового файла.

**ПРИЛОЖЕНИЕ А**

**Код программы**

namespace Library;

using System;

using System.Collections.Generic;

using System.IO;

using System.Text.RegularExpressions;

public class Calculator

{

    private readonly Dictionary<char, Func<double, double, double>> operations;

    public Calculator()

    {

        operations = new Dictionary<char, Func<double, double, double>>

        {

            ['+'] = (a, b) => a + b,

            ['-'] = (a, b) => a - b,

            ['\*'] = (a, b) => a \* b,

            ['/'] = (a, b) => a / b

        };

    }

    public void ProcessFile(string filePath)

    {

        if (!File.Exists(filePath))

        {

            Console.WriteLine($"Файл '{filePath}' не найден.");

            return;

        }

        string pattern = @"(\d+)\s\*([+\-\*/])\s\*(\d+)";

        var regex = new Regex(pattern);

        foreach (var line in File.ReadLines(filePath))

        {

            var match = regex.Match(line);

            if (match.Success)

            {

                double operand1 = double.Parse(match.Groups[1].Value);

                char op = match.Groups[2].Value[0];

                double operand2 = double.Parse(match.Groups[3].Value);

                if (operations.TryGetValue(op, out var operation))

                {

                    double result = operation(operand1, operand2);

                    Console.WriteLine($"{operand1} {op} {operand2} = {result}");

                }

                else

                {

                    Console.WriteLine($"Неизвестная операция: {op}");

                }

            }

            else

            {

                Console.WriteLine($"Неверный формат строки: {line}");

            }

        }

    }