МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

##### «Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского» (ННГУ)

##### Институт информационных технологий, математики и механики

Направление подготовки: «Программная инженерия»

##### ОТЧЕТ

По лабораторной работе

на тему:

##### «Реализация калькулятора с использованием постфиксной (обратной польской) формы для подсчета алгебраических выражений»

Выполнил: студент группы 3823Б1ПР2

Якимов И. В.

Научный руководитель:

Воденеева А.А.

Нижний Новгород

2024

**Оглавление**

[Введение 3](#_Toc184974134)

[1. Основные понятия и определения 4](#_Toc184974135)

[2. Методы решения 5](#_Toc184974136)

[3. Программная реализация 6](#_Toc184974137)

[3.1. Файловая структура программы 11](#_Toc184974139)

[4. Результаты тестов 12](#_Toc184974140)

[Заключение 13](#_Toc184974141)

# Введение

В реализации калькулятора используются операции «\*», «/», «-», «+», «^», «log(znach)», «log10(znach)», «sin(znach)», «cos(znach)». Обратная польская запись реализуется на стеке, стек основан на динамическом массиве, его основные методы Pop() – удаление элемента с вершины стека и Push() - добавление элемента на вершину стека.

Парсер внутри метода postfiks() реализован на основе словаря имеющего пары «ключ – значение».

Метод scobki() проверяет выражение на корректность скобок, в случае когда скобки не хватает/много выкидывает исключение

Константы выражения запоминаются в словаре, их значение запрашивается у пользователя через консоль до начала вычислений.

# Основные понятия и определения

1. Стек – структура данных в основе которой лежит принцип LIFO (last in first out).
2. Парсер – блок программы разбивающий выражение на типы лексем.
3. Динамический массив – структура данных, память которой находится в куче, с возможностью ее переопределения во время выполнения программы.
4. Map – структура данных, для хранения пар «ключ-значение»

# Методы решения

Посмотреть лекцию и потыкаться

# Программная реализация

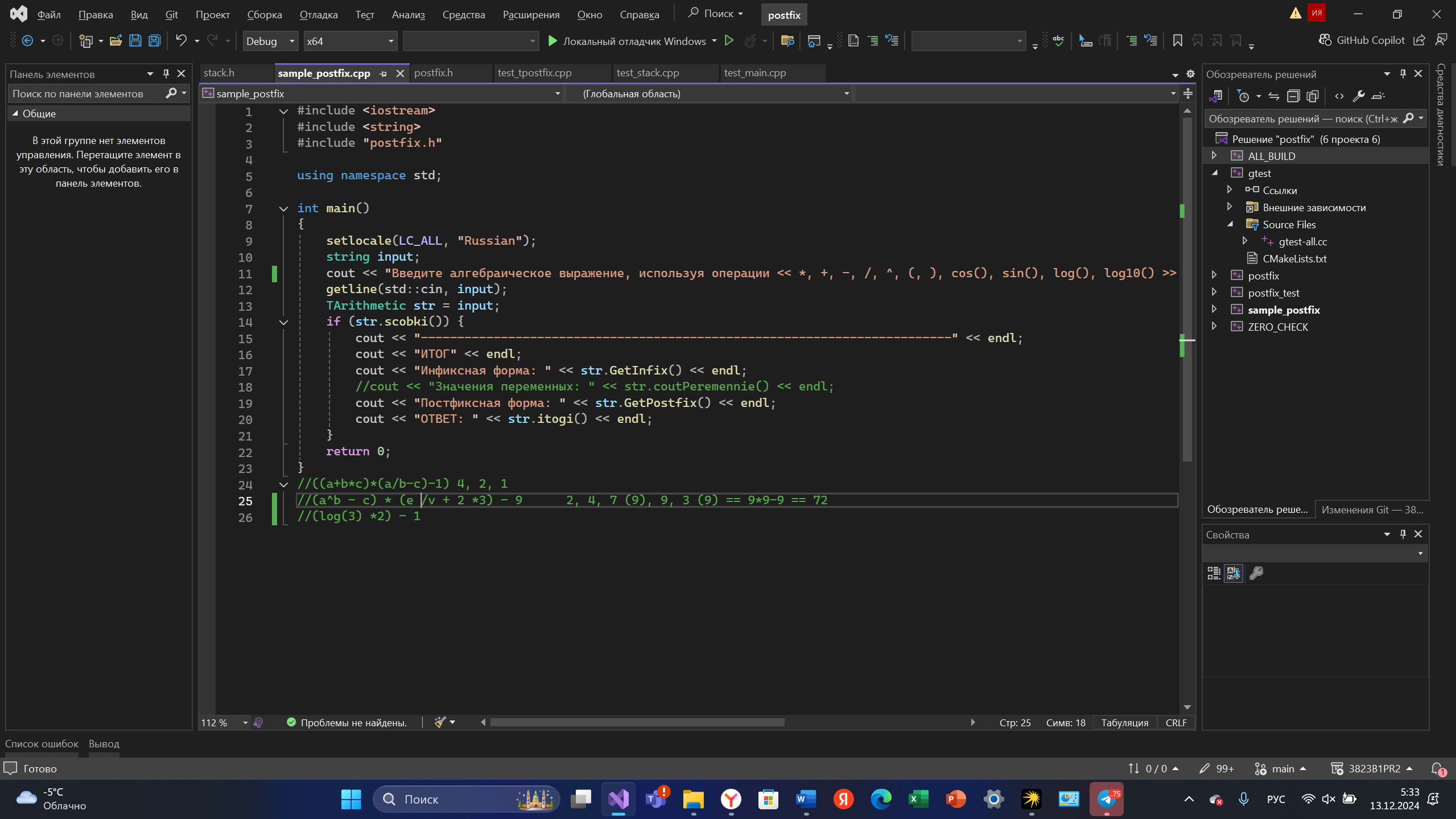


Рисунок 1. Sample\_postix.cpp

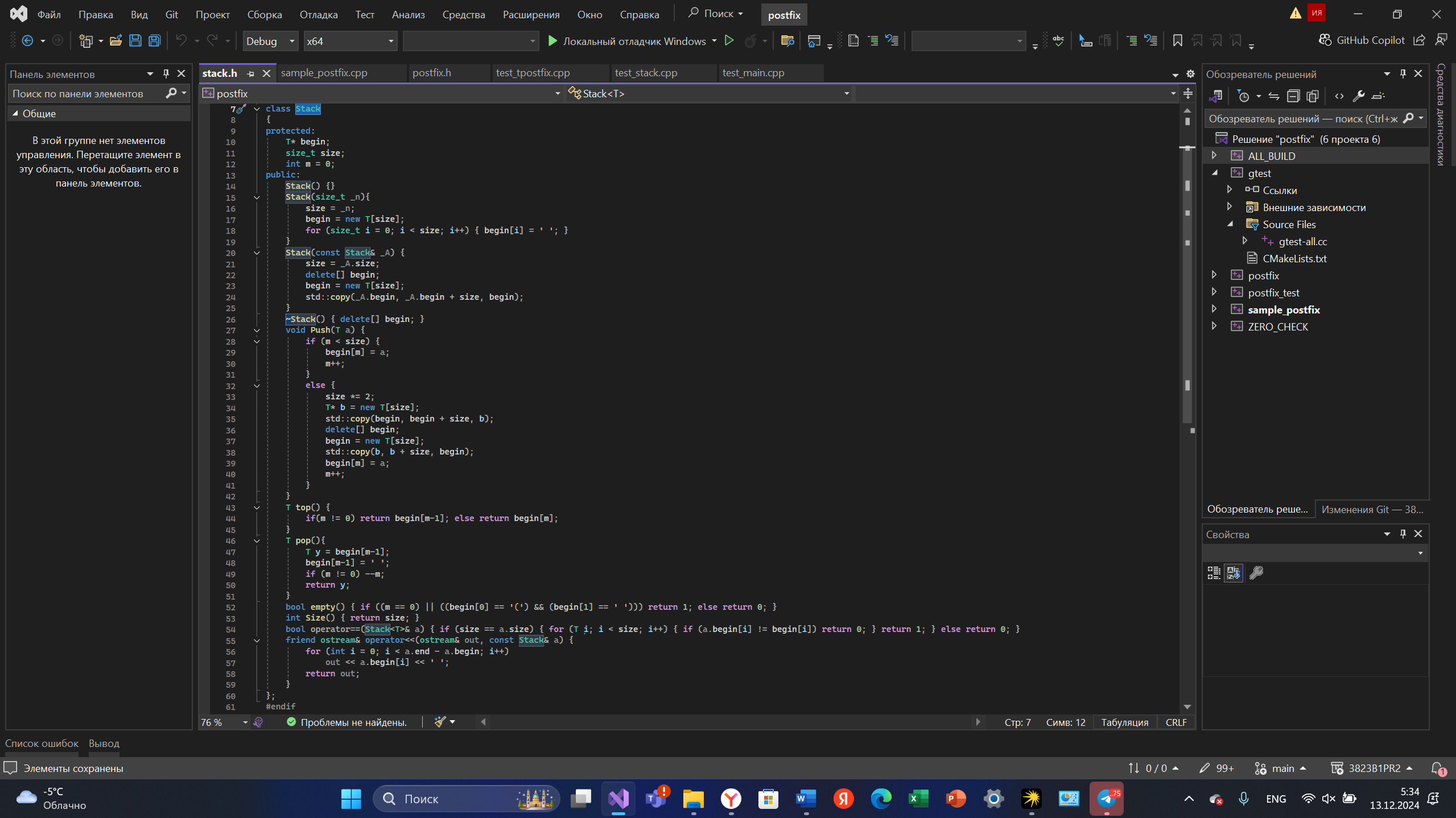


Рисунок 2. Stack.h

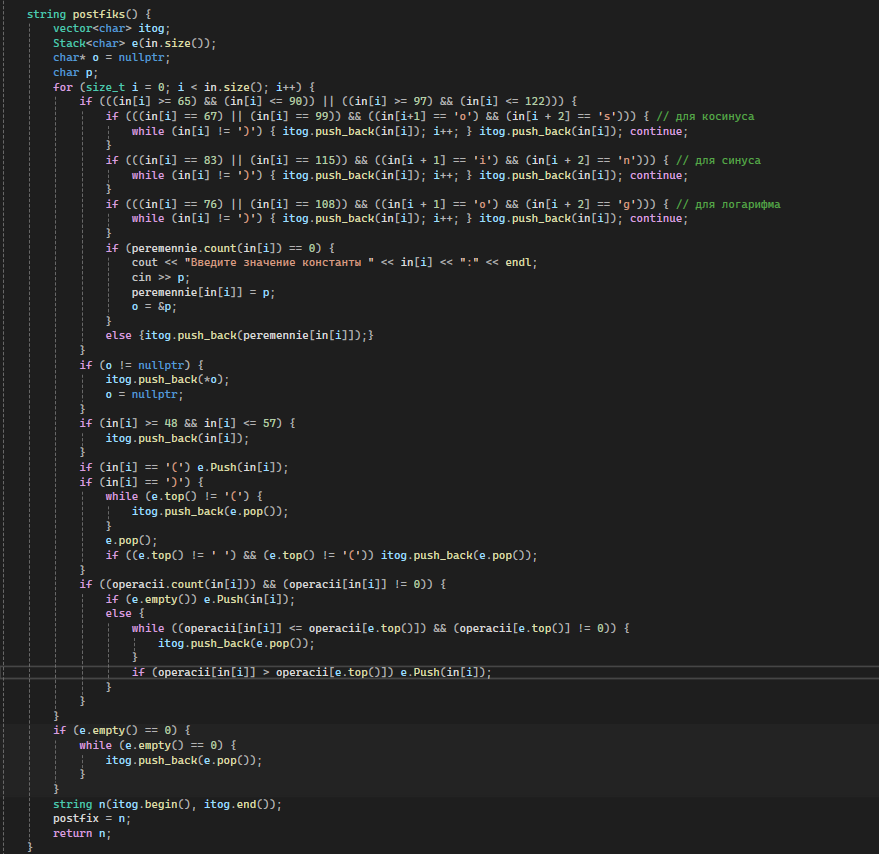


Рисунок 3. Метод Postfix()

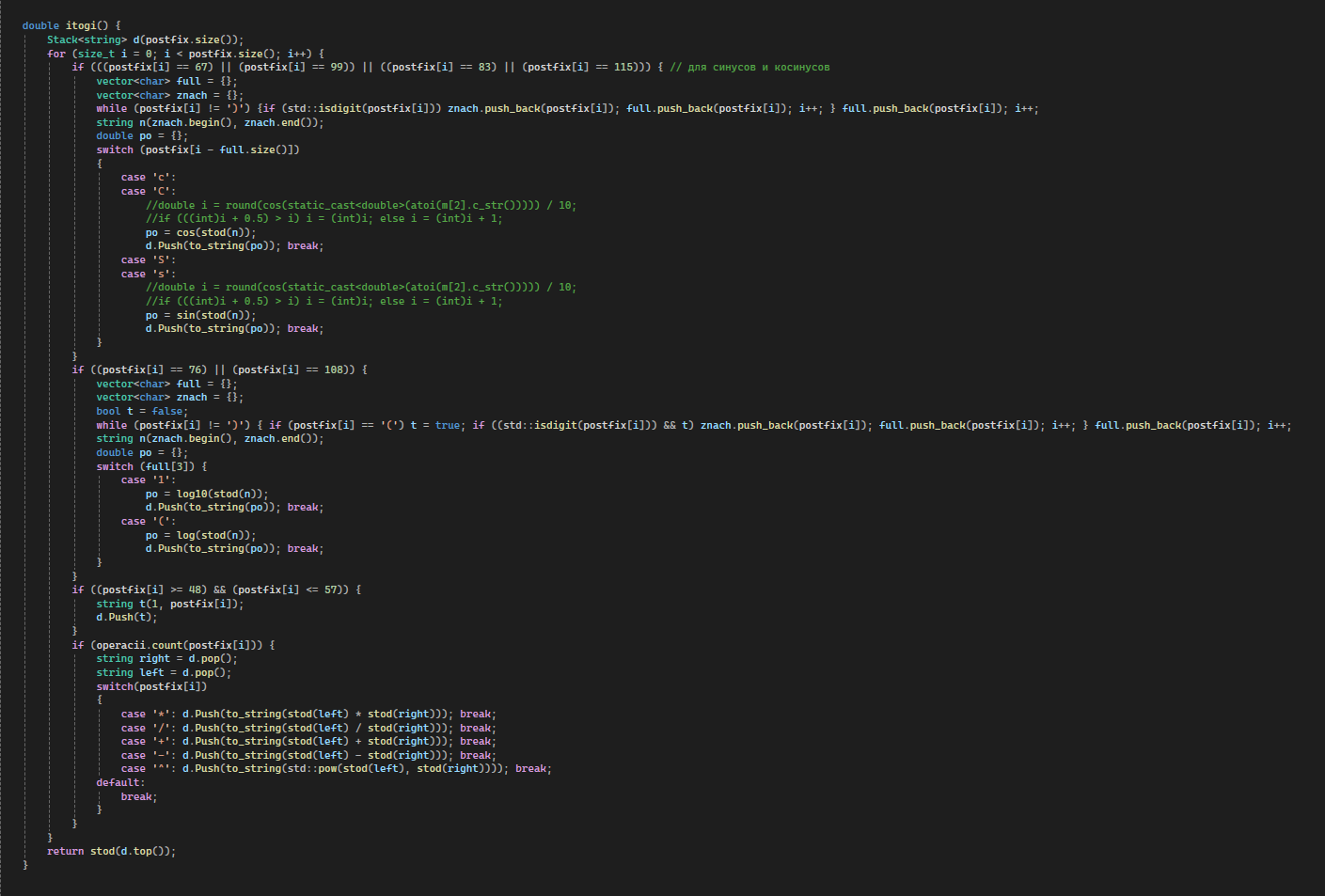


Рисунок 4. Метод itogi()

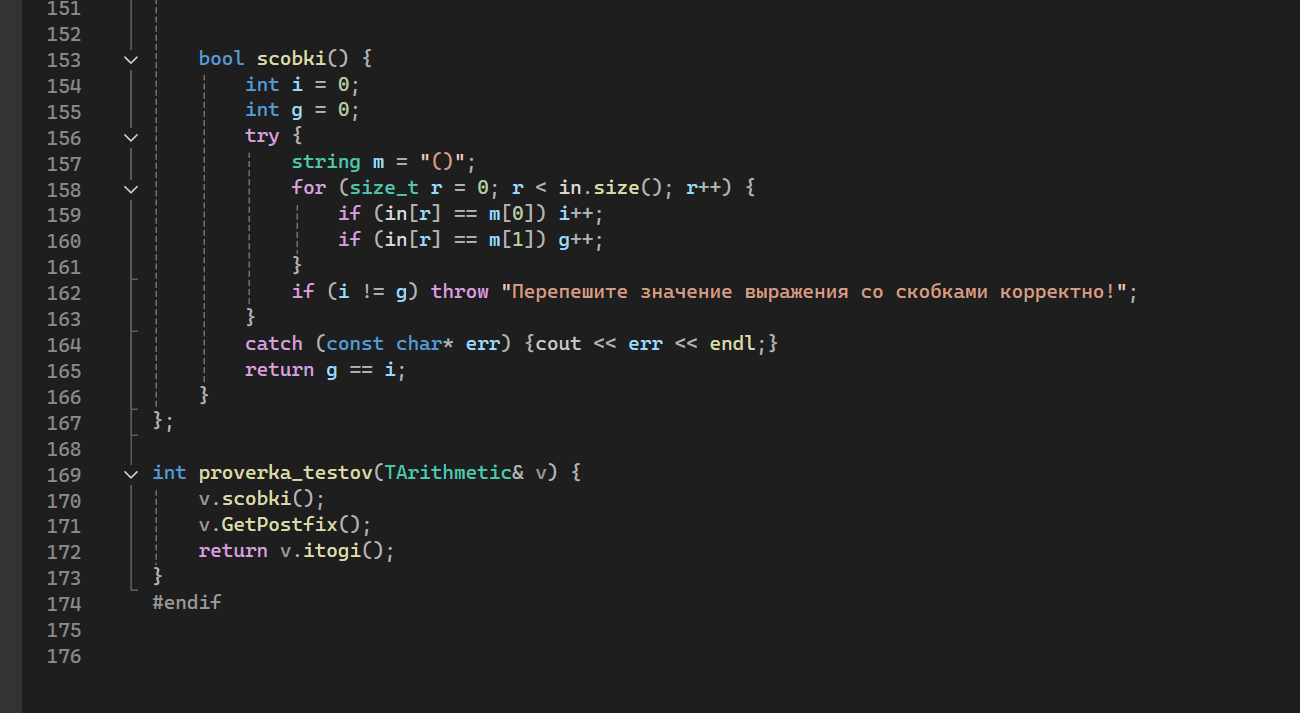


Рисунок 5. Метод Scobki и proverka\_testov

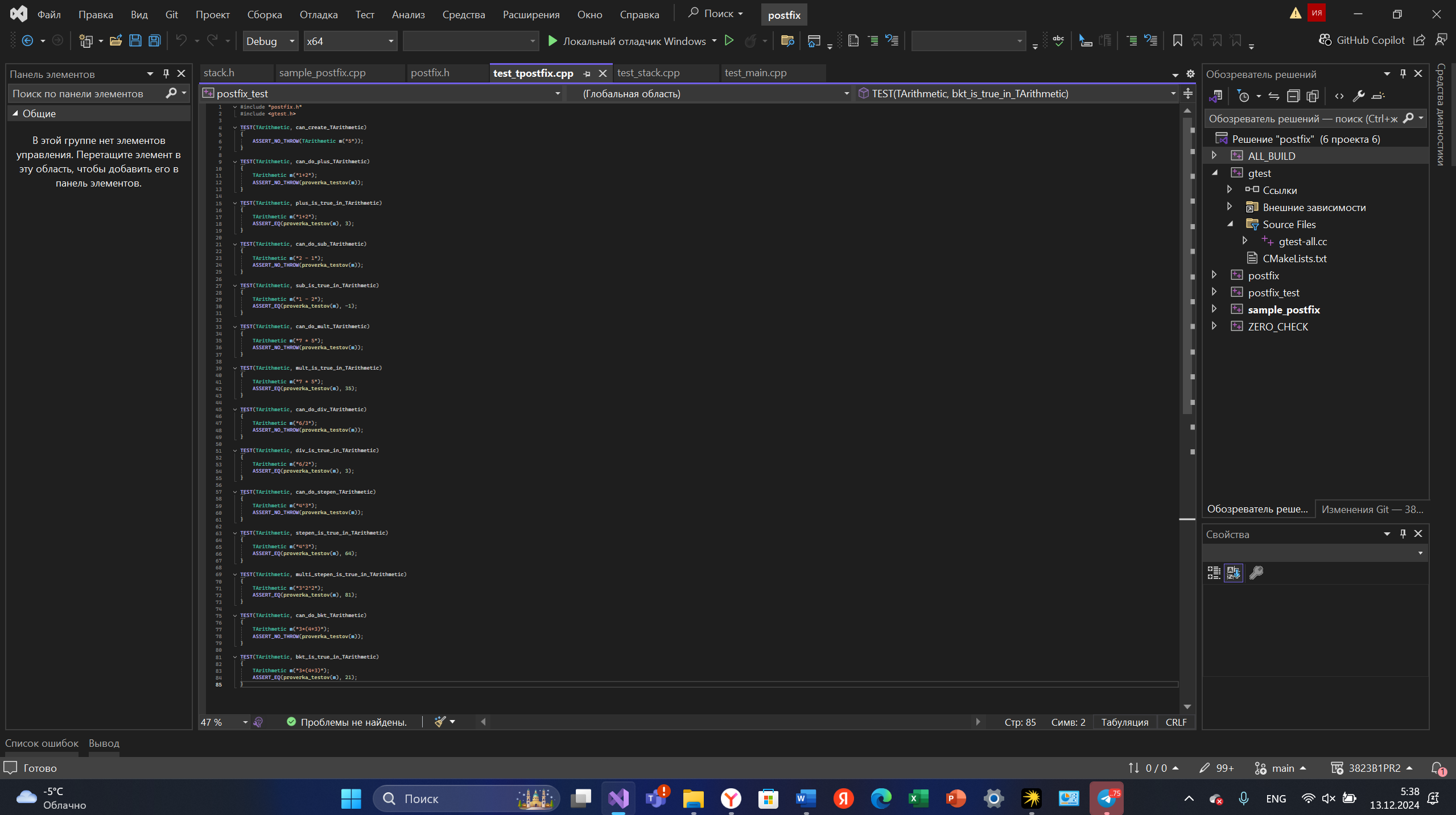


Рисунок 6. Tесты test\_tpostfix

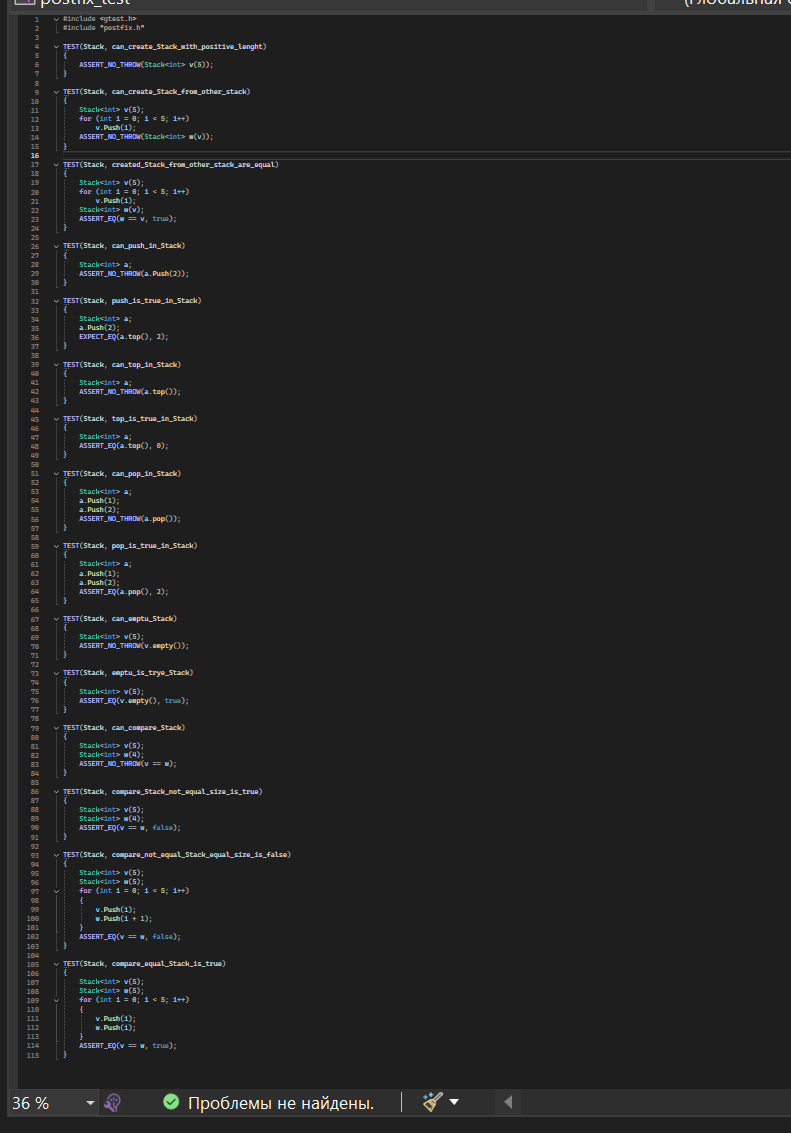


Рисунок 7. Test\_Stack

## Файловая структура программы

1. Stack.h – файл реализации класса Stack
2. Sample\_postfix.cpp – файл функции main
3. postfix.h– файл с реализацией класса TArithmetic
4. Test\_tpostfix.cpp – файл тестов класса TArithmetic
5. Test\_stack.cpp – файл тестов класса Stack
6. Test\_main.cpp– файл запуска тестов.

# Результаты тестов

# 

Рисунок 8. Результаты тестов класса Stack

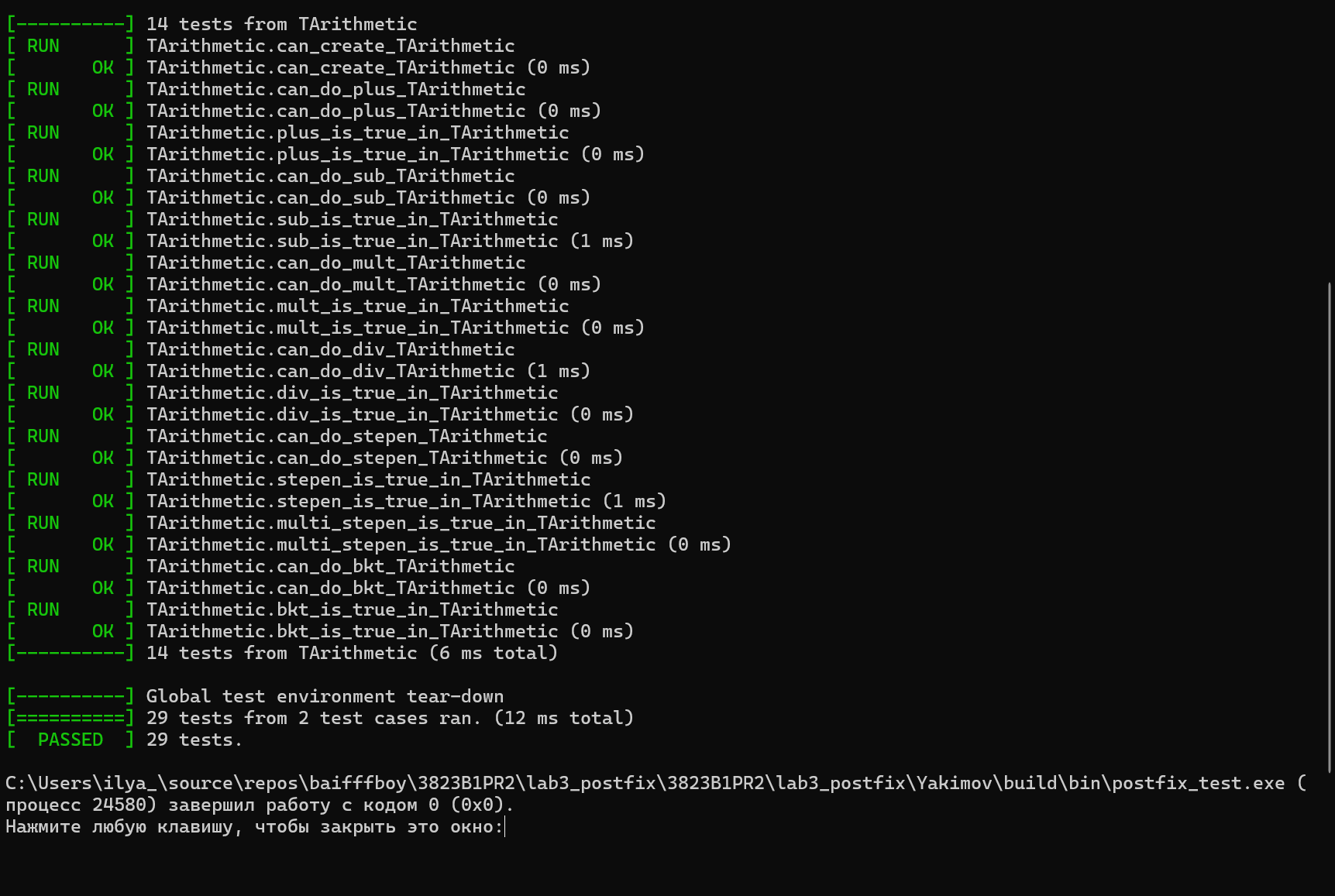


Рисунок 9. Результаты тестов класса TArithmetic

# Заключение

1. В лабораторной работе реализован калькулятор на стеке
2. Реализован стек

Варианты развития: реализация не только вычисления однозначных чисел