Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление арифметических выражений»**

**Выполнил**:

студент группы 381903-2

Гусев Никита Станиславович

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2020

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Разработать программу, выполняющую вычисление арифметического выражения с вещественными числами. Выражение в качестве операндов может содержать переменные и вещественные числа. Допустимые операции известны: +, -, /, \*. Допускается наличие знака "-" в начале выражения или после открывающей скобки. Программа должна выполнять предварительную проверку корректности выражения и сообщать пользователю вид ошибки и номера символов строки, в которых были найдены ошибки.

# Метод решения

Программа использует такие стандартные библиотеки, как: iostream, cmath, string. Также программа использует пространство имен std.

Реализован класс Lexem. В private объявлены поля: переменная типа double, переменная типа string (строка) и переменная типа bool, определяющая какой символ подается. В части public происходит проверка введенного на то, является ли введенное пользователем числом или операндом. Каждому операнду сопоставим свой приоритет операнда, позволяющий производить расчеты по всем правилам математики. Приоритет также определяется здесь. Класс используется для создания массива лексем. Методы возвращают число или операцию.

Реализован шаблонный класс Tstack. Private содержит int top, size – параметры стека. Public часть содержит конструкторы, деструктор. Функции (push, show, pop) «положить в стек», «взять из стека», «показать стек», булевая функция, которая определят заполненность стека, функция «очистки» стека (при top = 0). Класс Tstack используется для перевода в польскую запись и вычисления выражения.

Алгоритм перевода в ОПЗ

На вход получаем массив лексем (размер массива N). Далее в цикле (от 1 до N) берем лексему, если это число, то кладем в новый массив лексем, если же это операция, то кладем ее в стек (предварительно проверив его на пустоту); Если пустой, то кладем в стек, если нет, то сравниваем приоритет операции с приоритетом той операции, которая лежит в стеке. Если приоритет операции уже лежавшей в стеке больше приоритета поступающей, то достаем ее и кладем в новый массив. После выполнения этого цикла добавляем все операции из стека в конец нового массива. На выходе получаем массив лексем в ОПЗ.  
Метод find\_des:  
На вход подается массив лексем в ОПЗ и размер массива. В цикле (по массиву) достаем лексему, если это операнд, то кладем в стек, если бинарная операция, то достаем из стека два числа, производим с ними это операцию и кладем обратно в стек, если унарная операция, то из стека достаем только одно число. После завершения цикла, результат будет лежать на вершине стека.

Метод work:

На вход получает строку, разбирает строку на лексемы, записывая из в массив, проводит проверку правильности математического выражения, переводит в ОПЗ и вычисляет значение.

Проверка реализована с помощью switch(checking). Выдает разные результаты при разных ошибках

**Руководство пользователя**

1. Запустить программу
2. Ввести математическое выражение без пробелов.
3. Возможные операторы: +, -, /, \*, унарный –
4. При некорректном введении данных программа об этом сообщит, соответственно пользователь должен ввести все правильно

# Описание программной реализации

Проект содержит файлы arifmetika.cpp, Tstack.cpp,(содержат основной код) labb33, (в которой находится функция main), arifmetika.h, Tstack.h (Файлы-заголовки), Test.cpp, Artest.cpp (реализация google test’ов).

Подключенные библиотеки: iostream, string, cmath.

Также подключено namespace std.

В файле Tstack.cpp реализован шаблонный класс TStack. Файл arifmetika.cpp содержит реализацию основных методов(например, work) и класса Lexem. С помощью файлов Test.cpp и Artest.cpp реализована проверка корректности вычисления через google test’ы.

# Подтверждение корректности

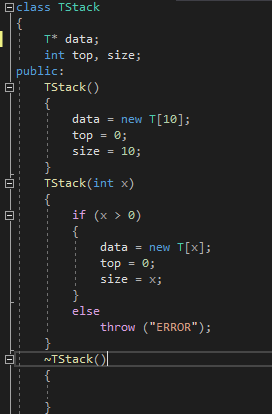
Для подтверждения корректности в программе реализован метод checking.

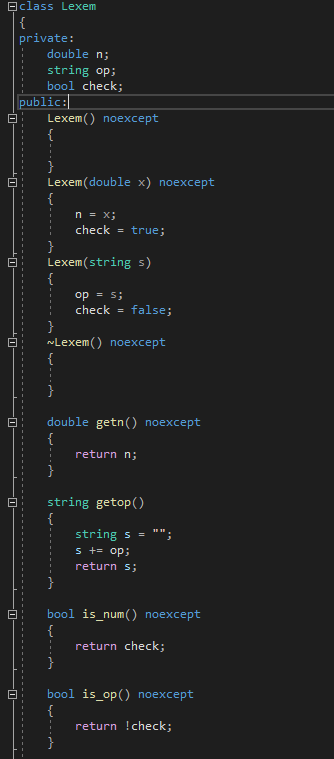
В нем на вход мы получаем массив лексем и его размер. В цикле проверяем, что следует после текущей лексемы, это может быть: бинарная операция, унарная операция, число. В случае как-либо несовпадения выдается ошибка.

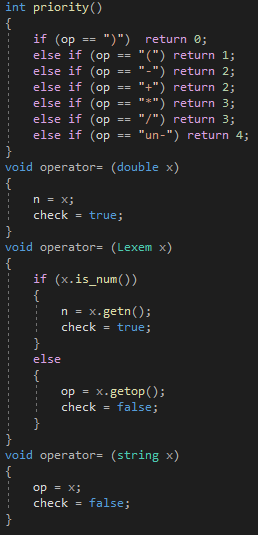
# Заключение

В ходе данной лабораторной работы была написана программа, которая вычисляет значение математического выражения с вещественными числами. Реализован шаблонный класс стек, класс лексем, алгоритм разбиения строки на лексемы, метод проверки, алгоритм перевода выражения в ОПЗ, алгоритм вычисления математического выражения, проведено тестирование через google tests.

# Приложение

 Рис.1

 Рис.2

 Рис.3

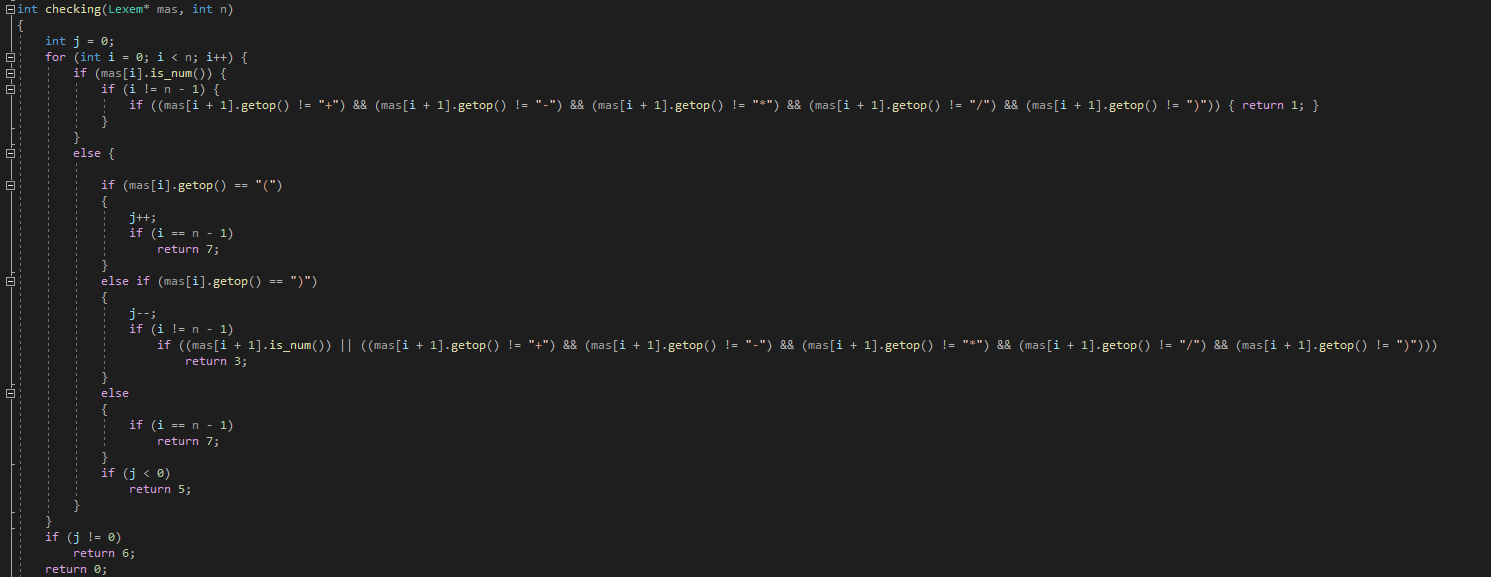


Рис.4