Федеральное государственное автономное образовательное учреждение

высшего образования

«Нижегородский государственный университет

им. Н.И. Лобачевского»

Институт информационных технологий, математики и механики

Отчёт по лабораторной работе

**Вычисление арифметических выражений**

Выполнил:

студент ИИТММ ПМИ гр. 382003-1

Котихин А.А

Проверил:

Ассистент кафедры МОСТ ИИТММ

Волокитин В. Д.

Нижний Новгород

2021 г.

**Содержание**

1. Постановка задачи
2. Руководство пользователя
3. Описание структур данных
4. Описание структуры работы программы
5. Описание алгоритмов
6. Приложения

**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.

**Руководство пользователя**

После запуска программы пользователь может ввести с клавиатуры необходимое арифметическое выражение.

Вод выражения выполняется по правилам:

1. Допускаются символы «1234567890.+-/\*xy»
2. Нельзя ставить две операции подряд «4+\*5»

После получения результата, пользователь может ввести ещё одно выражение.

**Описание структур данных**

В программе используются классы:

Symbol

TStack

Arithmetic

Также подключены следующие библиотеки:

<vector>

<iostream>

<string>

**Описание структуры работы программы**

1. Пользователь вводит выражение, которое записывается в строку.
2. Производится проверка строки на соответствие правилам ввода.
3. Происходит перевод строки в вектор лексем, в котором каждый элемент – это отдельная лексема.
4. Далее, программа переводит вектор лексем в обратную польскую запись.
5. После перевода, производится подсчет выражения в обратной польской записи.
6. Результат выводится в консоль.
7. Можно продолжить вводить выражения.

**Описания алгоритмов**

**Перевод строки в вектор**

1)Если встречается символ +,\*,/ то создаётся объект типа Symbol, в который записывается операция

2)Если встретился -, то сначала необходимо выяснить, унарный или нет: Если минус находится в начале строки, либо перед минусом не стоит число, или закрывающая скобка, то минус унарный, создаётся соответствующий объект типа Symbol. Если предыдущие условия не выполнены, то минус бинарный и создаётся соответствующий объект.

4)Если в строке встречается пробел, то он просто пропускается. Если встречается символ или точка, то идём дальше по строке, пока встречаются символы. По завершении этой операции создаётся объект Symbol, с соответствующим значением числа.

**Перевод в польскую запись**

Рассматриваем поочередно каждый элемент вектора:

1. Создаем дополнительный вектор символов, в который будем складывать готовую польскую запись.
2. Идем в цикле по элементам вектора:
   * Если этот элемент - число (или переменная), то просто помещаем его в выходную строку.
   * Если знак операции (+, -, \*, /), проверяется приоритет операции (в соответствии с правилами математики).
   * Проверка стека. Если стек все еще пуст, или находящиеся в нем символы имеют меньший приоритет, чем приоритет текущего символа, то помещаем текущий символ в стек. Если символ, находящийся на вершине стека имеет приоритет, больший или равный приоритету текущего символа, то извлекаем символы из стека в выходную строку до тех пор, пока выполняется это условие; затем переходим к пункту а).
   * Если текущий символ - открывающая скобка, то помещаем ее в стек.
   * Если текущий символ - закрывающая скобка, то извлекаем символы из стека в выходную строку до тех пор, пока не встретим в стеке открывающую скобку.

**Подсчет результата**

Создаётся стек символов (TStack<Symbol>)

1. В цикле проходим по вектору, где хранится польская запись
2. Если элемент – число, то вставляем в стек это число
3. Если элемент – унарный минус, то берем из стека последнее значение, умножаем его на -1 и помещаем обратно в стек
4. Если все предыдущие условия не выполнились, то берем из стека два операнда. Если текущая операция – плюс, то складываем операнды. Если бинарный минус – то вычитаем из первого операнда второй и так далее (для деления так же проверяется, равенство второго операнда нулю).
5. Ответом будет являться значение, которое останется в последнем объекте Symbol в стеке.

**Проверки.**

Для проверки нам понадобится строка, в которой хранятся все позволенные символы (Symbols).

Проходя по всей строке, мы сравниваем каждый символ строки со всеми символами строки позволенных символов. Если хотя бы один символ исходной строки не найдется в строке дозволенных символов, то программа выдаст соответствующее сообщение.

**Заключение**

Разработана программа, способная вычислить произвольное арифметическое выражение с вещественными числами. Работа программы протестирована с помощью Google Tests.