Федеральное государственное автономное образовательное учреждение

высшего образования

«Нижегородский государственный университет

им. Н.И. Лобачевского»

Факультет вычислительной математики и кибернетики

Отчёт по лабораторной работе

Вычисление арифметических выражений.

Выполнил:

студент ИИТММ гр. 381703-1

Девликамов В.О

Проверил:

Волокитин В.Д.

Нижний Новгород

2017 г.

Содержание

[Введение 3](#_Toc499930784)

[Постановка задачи 4](#_Toc499930785)

[Руководство пользователя 5](#_Toc499930786)

[Руководство программиста 6](#_Toc499930787)

[Описание структуры программы 6](#_Toc499930788)

[Описание структур данных 6](#_Toc499930789)

[Описание алгоритмов 7](#_Toc499930790)

[Заключение 10](#_Toc499930791)

[Литература 11](#_Toc499930792)

# Введение

# Дана строка, представляющая собой математическое выражение, содержащее числа, переменные, различные операции. Требуется вычислить его значение за O(n) , где n — длина строки.

Обратная польская нотация — это форма записи математических выражений, в которой операторы расположены после своих операндов.

Например, следующее выражение: 2 + 3\*2 + (2 – 1)\*(4 + 2\*3)

Обратная польская нотация была разработана австралийским философом и специалистом в области теории вычислительных машин Чарльзом Хэмблином в середине 1950-х на основе польской нотации, которая была предложена в 1920 г. польским математиком Яном Лукасевичем.

Данный алгоритм реализован в моей лабораторной работе.

# 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. Опционально - наличие математических функций (sin, соs, ln, exp, и т.д.) Программа должна выполнять предварительную проверку корректности выражения и сообщать пользователю вид ошибки и номера символов строки, в которых были найдены ошибки.

# Руководство пользователя

Необходимо ввести корректное арифметическое выражение. Пользователь может ввести параметр (любая буква латинского алфавита), также может ввести функцию (sin, cos, tan, arctan, log, sqrt). На выходе пользователь получает посчитанное выражение, или сообщение об ошибке (вид ошибки и номер символа строки)

# Руководство программиста

**Описание структур данных**

**main\_arithmetic.cpp**

string s – строка, которую вводит пользователь

**arithmetic.cpp**

pair <string, int> error – пара строка – содержит вид ошибки , число – содержит номер символа строки

Stack <pair<char, int>> temp – стек для проверки правильной скобочной последовательности

char v – значение i символа строки

Stack <Lexem> s, s1 – стеки для перевода в обратную польскую нотацию

map <string, int> perem – map для запоминания значений переменных

set <string> was – set для проверки был ли ранее введен такой же параметр

**arithmetic.h**

enum Lex\_Type – для определения типа введенного символа

bool may\_unary – определяет унарная ли операция (true – унарная, false –бинарная)

int pos – позиция символа в строке

**Описание алгоритмов**

Перевод строки в массив лексем:

Избавляемся от всех пробелов, для каждого элемента строки определяем вид лексемы и помещаем его в массив лексем

Перевод в обратную польскую нотацию происходит следующим способом:

1. Если лексема это число, то кладем во 2 стек
2. Если лексема это операция, то перекладываем из 1 стека во второй, пока приоритет текущей лексемы меньше или равна приоритету лексемы на вершине стека
3. Если лексема это открывающая скобка – то кладем в стек 1
4. Если лексема это закрывающая скобка – то перекладываем из стека 1 во 2 стек, пока не встретим открывающуюся скобку

Вычисление по обратной польской нотации происходит следующим способом:

1. Если лексема это число, то кладем в стек
2. Если лексема это бинарная операция – достаем из стека 2 числа, вычисляем и кладем обратно в стек
3. Если лексема это унарная операция – достаем из стека число, вычисляем и кладем обратно в стек

В конце в стеке останется одно число, которое и будет результатом посчитанного выражения.

# Заключение

В результате проделанной работы были достигнуты следующие результаты:

1. Разработан шаблонный класс Stack
2. Написан алгоритм, вычисляющий значение арифметического выражения
3. Функции тестируются с помощью Google Tests.

# Литература

1. <https://e-maxx.ru/algo/expressions_parsing>
2. https://ru.wikipedia.org/wiki/Обратная\_польская\_запись