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 №3**

**«Вычисление арифметических выражений»**

**Выполнил**:

студент группы 3822Б1ПР1

Ворошилов В. А.

**Проверил**:

доцент кафедры МОСТ, к.т.н.,

Сысоев А.В.

Нижний Новгород

2023

Содержание

[Введение 3](#_Toc118232948)

[1. Постановка задачи 4](#_Toc118232949)

[2. Руководство пользователя 5](#_Toc118232950)

[3. Руководство программиста 6](#_Toc118232951)

[3.1. Описание структуры программы 6](#_Toc118232952)

[3.2. Описание алгоритмов 6](#_Toc118232953)

[4. Результаты экспериментов 7](#_Toc118232954)

[Заключение 8](#_Toc118232955)

[Литература 9](#_Toc118232956)

[Приложение 10](#_Toc118232957)

# Введение

Арифметическое выражение – выражение, составленное из операндов, соединённых арифметическими операциями (+, -, \*, /).

Традиционная запись арифметического выражения, например, (a + b \* c) \* (c / d – e), подразумевает, что операнды отделяются друг от друга операциями, а порядок действий определяется расстановкой скобок и приоритетом операций. Такой способ записи называется инфиксной формой арифметического выражения.

Вычисление на компьютере выражения в инфиксной форме реализовывать неудобно. Обычно производят преобразование в другие, более удобные представления.

В 1950-ых годах австралийский учёный Чарльз Хэмблин разработал форму записи арифметических выражений, в которой операция располагается в выражении справа от её операндов. Так, арифметическому выражению (a + b \* c) \* (c / d – e) будет соответствовать запись abc\*+cd/e-\*. Такой способ записи называется постфиксной (обратной польской) формой арифметического выражения.

Такая форма записи более проста для реализации вычислений арифметического выражения в компьютерной программе.

# Постановка задачи

Цель данной работы — разработка структуры данных Стек и ее использование для расчета арифметических выражений с использованием обратной польской записи (постфиксной формы).

Выполнение работы предполагает решение следующих задач:

1. Разработка интерфейса шаблонного класса TStack.
2. Реализация методов шаблонного класса TStack.
3. Разработка интерфейса класса Postfix для работы с постфиксной формой.
4. Реализация методов класса Postfix.
5. Разработка и реализация тестов для классов TStack и Postfix на базе Google Test.
6. Публикация исходных кодов в личном репозитории на GitHub.

# Руководство пользователя

…

# Руководство программиста

## Описание структуры программы

…

## Описание алгоритмов

*Алгоритм перевода инфиксной (обычной) формы в постфиксную:*

* Для каждой лексемы в инфиксной форме:
  + Если лексема – операнд, поместить её в постфиксную форму.
  + Если лексема – открывающаяся скобка, поместить её в стек.
  + Если лексема – закрывающаяся скобка:
    - Пока на вершине стека не открывающаяся скобка:
      * Извлечь из стека элемент,
      * Поместить элемент в постфиксную форму,
    - Извлечь из стека открывающуюся скобку.
  + Если лексема – операция:
    - Пока приоритет лексемы меньше или равен приоритета верхнего элемента стека:
      * Извлечь из стека элемент,
      * Поместить элемент в постфиксную форму
    - Поместить лексему в стек
* По исчерпании лексем в инфиксной форме перенести все элементы из стека в постфиксную форму.

*Алгоритм вычисления выражения в постфиксной форме:*

* Для каждой лексемы в постфиксной форме:
  + Если лексема - операнд, поместить ее значение в стек.
  + Если лексема – операция:
    - Извлечь из стека значения двух операндов,
    - Выполнить операцию (верхний элемент из стека является правым операндом, следующий за ним – левым),
    - Положить результат операции в стек
* По исчерпании лексем из постфиксной формы на вершине стека будет результат вычисления выражения

# Результаты экспериментов

…

# Заключение

…

# Литература

1. Кнут Д. Э. Искусство программирования. Том 3. Сортировка и поиск = The Art of Computer Programming. Volume 3. Sorting and Searching / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). – 2-е изд. – Москва: Вильямс, 2007. – Т. 3. – 832 с.
2. Страуструп Бьерн Язык программирования C++. Специальное издание. Пер. с англ. – М.: Издательство Бином, 2017 г. – 1136 с.
3. Сайт Algolist. Сортировка выбором – <http://algolist.manual.ru/sort/select_sort.php>.
4. Сайт Algolist. Сортировка пузырьком – <http://algolist.manual.ru/sort/bubble_sort.php>.

…

# Приложение

**Postfix.h**

#ifndef \_\_POSTFIX\_H\_\_

#define \_\_POSTFIX\_H\_\_

#include <string>

#include <iostream>

#include "stack.h"

using namespace std;

class TPostfix

{

string infix;

string postfix;

string operands = "+-\*/";

//int arity[4] = { 2,2,2,2 };

int priority[4] = { 1,1,2,2 };

bool isCorrect(string str);

public:

TPostfix(string inf = "a+b")

{

if (!isCorrect(inf))

throw "Error";

for (int i = 0; i < inf.length(); i++)

if (inf[i] != ' ')

infix += inf[i];

if (infix.length() < 1)

throw "Error";

}

string GetInfix() { return infix; }

string GetPostfix() { return postfix; }

string ToPostfix();

double Calculate(); // Ввод переменных, вычисление по постфиксной форме

};

#endif

**Postfix.cpp**

#include "postfix.h"

#include "stack.h"

bool TPostfix::isCorrect(string str)

{

string unacceptable = "!@#^&.,<>{}[]|=\_?";

int count = 0, k = 0, countOperation = 0, length = str.length();

if ((operands.find(str[0]) != string::npos) || (operands.find(str[length - 1]) != string::npos))

return false;

for (int i = 0; i < length; i++)