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 3821Б1ПМ2

Тышкевич А.Р.

**Проверил**:

Преподаватель ИИТММ

Волокитин В.Д.

Нижний Новгород

2022

Содержание

[Введение 3](#_Toc86935064)

[1. Постановка задачи 4](#_Toc86935065)

[2. Руководство пользователя 5](#_Toc86935066)

[3.1 Описание структуры программы 6](#_Toc86935067)

[3.2 Описание алгоритмов 6](#_Toc86935068)

[4. Результаты экспериментов 8](#_Toc86935071)

[Заключение 9](#_Toc86935072)

[Литература 10](#_Toc86935073)

# Введение

В данной лабораторной работе будет рассматриваться возможность вычисления выражений с помощью программы, которая будет принимать на вход обычную строку, а на выход будет выдавать результат вычислений. Строка будет обработана с помощью алгоритма обратной польской записи, который реализован через стек.

# Постановка задачи

Входные данные:

* Арифметическое выражение в инфиксной форме, которое может содержать такие операторы, как **«**+**»**, **«**-**»**, **«**\**»**, **«**\***»**.

Выходные данные:

* Арифметическое выражение в постфиксной форме («Обратная польская запись»).
* Вычисленное арифметическое выражение.

То есть при выполнении работы мы решим следующие задачи:

* Разработка интерфейса шаблонного класса **TStack**.
* Реализация методов шаблонного класса **TStack**.
* Разработка интерфейса класса **TPostfix** для работы с постфиксной формой.
* Реализация методов класса **TPostfix**.
* Разработка и реализация тестов для классов **TStack** и **TPostfix** на базе Google Test.

# Руководство пользователя

1. Программа имеет вид консольного приложения.

2. Сначала программа предложит ввести пользователю арифметическое выражение в инфиксной форме (возможен ввод и чисел, и переменных).

1. Далее программа выведет выражение в постфиксной форме.
2. В итоге будет выведено вычисленное арифметическое выражение.

## Описание структуры программы

**Класс TStack:**

* T\* pMem – указатель на память, выделенную для хранения стека.
* size – хранит размер выделенной памяти (тип int).
* top – хранит индекс элемента массива, в котором хранится последнее добавленное значение стека (тип int).
* DataCount – хранит количество запомненных в стеке значений (тип int).
* Класс TStack:
* TStack(int \_size) – конструктор класса
* ~TStack(int \_size) – деструктор класса
* GetNextIndex – возвращает индекс следующего элемента в стеке
* IsEmpty – проверка на пустоту стека
* IsFull – проверка на полноту стека
* Put – добавление элемента в стек

**Класс TPostfix:**

* infix – хранит строку в виде инфиксной формы
* postfix – хранит строку в виде постфиксной формы
* Priority – возвращает приоритет оператора
* GetInfix – возвращает строку в инфиксной форме
* GetPostfix – возвращает строку в постфиксной форме
* ToPostfix – переводит из инфиксной формы в постфиксную
* Calculate – вычисление по постфиксной форме

3.2 Описание алгоритмов

Алгоритм перевода выражения в постфиксную запись:

1.Константы и переменные кладутся в формируемую запись в порядке их появления в исходном массиве. При появлении операции в исходном массиве:

* если в стеке нет операций или верхним элементом стека является открывающая скобка, операции кладётся в стек;
* если новая операции имеет больший приоритет, чем верхняя операции в стеке, то новая операции кладётся в стек;
* если новая операция имеет меньший или равный приоритет, чем верхняя операции в стеке, то операции, находящиеся в стеке, до ближайшей открывающей скобки или до операции с приоритетом меньшим, чем у новой операции, перекладываются в формируемую запись, а новая операции кладётся в стек.

2. Открывающая скобка кладётся в стек.

3. Закрывающая скобка выталкивает из стека в формируемую запись все операции до ближайшей открывающей скобки, открывающая скобка удаляется из стека.

4. После того, как мы добрались до конца исходного выражения, операции, оставшиеся в стеке, перекладываются в формируемое выражение.

Алгоритм подсчета постфиксной записи:

1. Обработка входного символа:

* если на вход подан операнд, он помещается на вершину стека;
* если на вход подан знак операции, то соответствующая операция выполняется над требуемым количеством значений, извлечённых из стека, взятых в порядке добавления.

Результат выполненной операции кладётся на вершину стека.

2. Если входной набор символов обработан не полностью, перейти к шагу 1.

3. После полной обработки входного набора символов результат вычисления выражения лежит на вершине стека.

# Результаты экспериментов

Для выявления ошибок в работе программы были реализованы тесты на основе фреймворка для написания автоматических тестов Google Test. Все они пройдены успешно.

# 

# Заключение

Для поставленной задачи была разработана программа, реализующая вычисление арифметических операций с помощью специальной структуры хранения данных – стека и перевода инфиксной формы выражения в постфиксную.

# Литература

1. Сайт ru.wikipedia.org Обратная польская запись - https://ru.wikipedia.org/wiki/Обратная\_польская\_запись