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 3822Б1ПМ1

Кубышкин Владимир

**Проверил**:

Преподаватель ИИТММ

Волокитин В.Д.

Нижний Новгород

2023

Содержание

[Введение 3](#_Toc153816886)

[1. Постановка задачи 4](#_Toc153816887)

[2. Руководство пользователя 5](#_Toc153816888)

[3.1 Описание структуры программы 6](#_Toc153816889)

[3.2 Описание алгоритмов 7](#_Toc153816890)

[Заключение 8](#_Toc153816893)

[Литература 9](#_Toc153816894)

# Введение

В данной лабораторной работе будет рассматриваться программа, которая на выходе будет выдавать результат вычислений, принимая на вход обычную строку. Алгоритм обратной польской записи, реализованный через стек, будет обрабатывать данную строку.

# Постановка задачи

Входные данные:

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

Выходные данные:

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

Задачи:

* Разработка интерфейса шаблонного класса **TStack**.
* Реализация методов шаблонного класса **TStack**.
* Разработка интерфейса класса **Postfix** для работы с постфиксной формой.
* Реализация методов класса **Postfix**.
* Разработка и реализация тестов для классов **TStack** и **Postfix** 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\* data – указатель на память, выделенную для хранения стека.
* size – хранит размер выделенной памяти (тип size\_t).
* index – хранит индекс элемента массива, в котором хранится последнее добавленное значение стека (тип int).
* Класс TStack:
* TStack(size\_t n), TStack() – конструкторы класса
* TStack(const TStack&a) – конструктор копирования
* ~TStack() – деструктор класса
* IsEmpty – проверка на пустоту стека
* IsFull – проверка на полноту стека
* push – добавление элемента в стек
* clear – пересоздает массив
* pop – удаление элемента из стека

**Класс Postfix:**

* operationPriority – приоритет операций
* checkBrackets – кол-во скобок
* isDigit – проверка операнда на цифру
* checkingLine – проверка строки
* calculate - расчеты

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.

# Заключение

Разработаны и реализованы интерфейсы и методы шаблонных классов TStack и Postfix, написаны тесты для них на базе Google Test.

Написана 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>
2. Лекция Сысоева А.В. : <https://cloud.unn.ru/s/6g44ey6HFB4ncDy>