МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

**«Преобразование арифметических операций в обратную польскую запись»**

**Выполнил:** студент группы 381706-1

Голованова Елена Александровна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018.

Оглавление

[1. Введение 3](#_Toc1946076)

[2. Постановка задачи 5](#_Toc1946077)

[3. Руководство пользователя 6](#_Toc1946078)

[4. Руководство программиста 7](#_Toc1946079)

[a. Описание структуры программы 7](#_Toc1946080)

[b. Описание структур данных 7](#_Toc1946081)

[c. Описание алгоритмов 8](#_Toc1946082)

[5. Заключение 10](#_Toc1946083)

[6. Список литературы 11](#_Toc1946084)

# Введение

В основе появления языков пpогpаммиpования высокого уровня лежит множество причин. Одной из них являются вычислительные задачи, требующие больших объёмов рутинных вычислений. Чаще всего пользователь не может наблюдать в среде способ разбора и вычисления необходимого выражения. Вследствие чего перед нами была поставлена такая задача – изучить и реализовать алгоритм трансляции и вычисления арифметического выражения. Данный алгоритм кроме сложной структуры, имеющей много нюансов, имеет и важное практическое применение, а именно он необходим для создания новых компиляторов высокого уровня.

К языкам пpогpаммиpования предъявляются требования максимального приближения формы записи вычислений к естественному языку математики. В связи с этим одной из первых областей системного пpогpаммиpования, сфоpмиpовалась такая, как исследование способов трансляции выражений. В этой области получены многочисленные результаты.

Всем известно, что в математике существует древняя традиция помещать оператор между операндами (x+y), а не после операндов (xy+). Форма с оператором между операндами называется инфиксной записью. Форма с оператором после операндов называется постфиксной, и именно она получила наибольшее pаспpостpанение. Такая запись обеспечивает подготовленность операндов до того как необходимо выполнять операцию, кроме этого эта запись может выразить любую формулу без скобок, а самое главное она удобна для вычисления в машинах со стеками, что позволяет ускорить процесс вычисления в несколько раз.

Обра́тная по́льская нота́ция (ОПН) — форма записи математических выражений, в которой операнды расположены перед знаками операций. Также именуется как обратная польская запись, обратная бесскобочная запись (ОБЗ), постфиксная нотация, бесскобочная символика Лукасевича, польская инверсная запись, ПОЛИЗ.

Достоинством обратной польской записи является то, что она обобщается n-адическим оператором, а инфиксная нотация на самом деле работает только с двумя операндами, т.е. по своей природе подходит только для бинарных операций. Еще одним достоинством обратной польской записи служит то, что приоритетом операций служит порядок их появления, при этом никогда не понадобятся скобки.



Рисунок 1. Постфиксная форма

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация метода перевода выражений из инфиксной в постфиксную нотацию ConvertToPol.
2. Разработка и реализация метода, вычисляющего результат выражения, записанного в постфиксной нотации..
3. Создать класс TException во избежания ошибок.
4. Выполнение примера использования программы.
5. Реализация простых тестов на базе Google Test.

# Руководство пользователя

Работа пользователя происходит так:

1. При запуске программы пользователя просят ввести математическое выражение для дальнейшего преобразования в обратную польскую запись и вычисления.
2. После этого на экран выводится выражение, введенное пользователем.
3. Далее на экране появляется это же выражение, но после перевода в обратную польскую запись.

# 4. Руководство программиста

## Описание структуры программы

Для реализации лабораторной работы создается несколько модулей:

* Модуль Polish

Этот модуль содержит реализацию примера использования программы.

* Модуль PolishLib

Содержит файл Polish.h, в котором описаны методы перевода в обратную польскую запись и вычисления результата в ней, а также два вспомогательных метода. В файле Polish.cpp происходит реализация методов. Есть также файл Stroka.h и Stroke.cpp, в которых описаны методы с реализацией вспомогательного класса TString

* Модуль PolishTest

Этот модуль содержит тесты, проверяющие работу всех методов .

* Модуль Exception

Этот модуль содержит класс исключений.

## Описание структур данных

Класс TString –шаблонный.

1. В зоне protected определены два поля:

* char\* string - массив для хранения строки;
* int length - длина строки.

2. В зоне public:

* TString() - конструктор;
* TString(TString &str) - конструктор копирования;
* TString(char\* \_string) - конструктор копирования;
* ~TString() - деструктор;
* int GetLength() - получить длину строки;
* void DeleteString(); - удалить строку
* TString operator + (TString &str);
* TString& operator = (TString &str);
* Char& operator [] (int i);
* friend istream& operator >> (istream &istr, TString &str);
* friend ostream& operator << (ostream &ostr, TString &str).

Класс Polish.h.

1. В зоне public определены несколько методов:

* int GetPriority(const char prior); - вычисление приоритета операции;
* TQueue<char> ConvertToPol(TString str); - преобразует выражение из инфиксной формы в префиксную;
* double Res(TQueue<char> que); вычисление результата операции, записанного в обратной польской нотации;
* bool IsPolish(char s).

## Описание алгоритмов

* Преобразование выражения из инфиксной формы в обратную польскую запись.
  + - 1. Приоритет операций:

|  |  |
| --- | --- |
|  |  |

* + - 1. Для хранения операций используется стек st, для хранения выражения в обратной польской записи используется очередь q.
      2. Исходное выражение просматривается слева направо.
      3. Операнды по мере их появления помещаются в очередь q.
      4. Символы операций и левые скобки помещаются в стек st.
      5. При появлении правой скобки последовательно изымаются элементы из стека st и переносятся в очередь q. Данные действия продолжаются либо до опустошения стека st, либо до попадания в стеке st на левую скобку.

7. Если текущая операция op, выделенная при обходе выражения, имеет больший вес, чем операция на вершине стека st, то она кладется в стек st. Иначе последовательно изымаются элементы из стека st и переносятся в очередь q. Данные действия продолжаются либо до опустошения стека st либо до того, как приоритет операции вершины стека будет не больше приоритета текущей операции op. Как только эти действия прекращаются текущая операция op кладется в стек st.

# 5. Заключение

В результате выполнения данной лабораторной работы была разработана программа «Обратная польская запись».

Помимо реализации в лабораторной работе есть пример использования.

Получены новые знания и навыки.

Были реализованы тесты для проверки работоспособности вышеперечисленных классов на базе GoogleTest.

# 6. Список литературы

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
3. Fb.ru: Обратная польская запись: алгоритм, методы и примеры http://fb.ru/article/321181/obratnaya-polskaya-zapis-algoritm-metodyi-i-primeryi.
4. Википедия: свободная электронная энциклопедия: на русском языке: [https://ru.wikipedia.org/wiki/Обратная-польская\_запись