МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

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

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

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

**Выполнил:** студент группы 381706-1

Шашкин Евгений Вадимович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

**Оглавление**

[1. Введение. 2](#_Toc1244445)

[2. Постановка задачи. 3](#_Toc1244446)

[3. Руководство пользователя. 4](#_Toc1244447)

[4. Руководство программиста. 5](#_Toc1244448)

[4.1. Описание структуры программы. 5](#_Toc1244449)

[4.2. Описание структур данных. 5](#_Toc1244450)

[4.3. Описание алгоритмов. 5](#_Toc1244451)

[5. Заключение. 7](#_Toc1244452)

[6. Литература. 8](#_Toc1244453)

# Введение.

**Обратная польская запись**— форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Также именуется как обратная польская запись, обратная бесскобочная запись, постфиксная нотация, бесскобочная символика Лукасевича, польская инверсная запись, ПОЛИЗ.

Чтобы дать индуктивное определение постфиксной нотации, обозначим выражения в инфиксной нотации , , , эквивалентные им выражения в постфиксной нотации , , , соответственно; — произвольный бинарный оператор, тогда:

1. Если — переменная или константа, то есть .

2. Если — выражение вида , то есть .

3. Если — выражение вида , то есть .

Например, выражение вида в ОПЗ будет иметь вид: .

# Постановка задачи.

В данной лабораторной работе нужно разработать эффективную библиотеку для преобразования обыкновенных математических выражений в обратную польскую запись.

Для этого нам нужно:

* Описать и реализовать класс строк **TMyString**.
* Описать и реализовать функции для преобразования математических выражений в ОПЗ.
* Описать и реализовать функцию для вычисления выражения по ОПЗ.
* Протестировать функции для преобразования выражений в ОПЗ и их последующее вычисление с помощью Google Test.
* Реализовать класс **TMyException** для обработки исключений, которые могут возникнуть в результате работы.
* Продемонстрировать работу функций для преобразования и вычисления математических выражений в ОПЗ.

# Руководство пользователя.

**C:\Users\Евгений\Desktop\polish.png**

*Рис 1. Пример работы программы*

Программа работает следующим образом:

Данное выражение преобразуется в ОПЗ (45 55 + 10 / 5 5 \* -) и затем вычисляется.

# Руководство программиста.

# Описание структуры программы.

Программа состоит из модулей:

* **PolishNotation** – содержит в себе файл **polish\_main.cpp** с реализацией примера использования класса **ОПЗ**.
* **PolishNotationLib –** содержит в себе файлы **Polish.h** и **MyString.h**, в которых описаны и реализованы **ОПЗ** и класс **TMyString** соответственно.
* **PolishNotationTest** – содержит в себе файл **test\_polish.cpp**, в котором находится набор тестов, для проверки работоспособности класса **ОПЗ**.
* **MyExceptionLib** –содержитв себе файл **MyException.h** с реализацией класса исключений **TMyException**.

# Описание структур данных.

**Класс TMyString.**

Класс **TMyString** содержит два поля со спецификатором **protected**:

* **char\* str** – строка для хранения символов.
* **int len** – длина строки.

Далее идут методы класса со спецификатором доступа **public**:

* **TMyString()** – конструктор по умолчанию.
* **TMyString(char\* \_str)** – конструктор-инициализатор.
* **TMyString(TMyString &obj)** – конструктор копирования.
* **~TMyString()** – деструктор.
* **int GetLen()** – метод, возвращающий длину строки.
* **void DeleteStr()** – метод, удаляющий строку.
* **TMyString operator+(TMyString &obj)** – оператор сложения строк.
* **TMyString& operator=(TMyString &obj)** – оператор присваивания строк.
* **char& operator[](int \_index)** – оператор доступа у символам строки.
* **friend std::istream& operator>>(std::istream &istr, TMyString &obj)** – дружественная функция ввода строки.
* **friend std::ostream& operator<<(std::ostream &ostr, const TMyString &obj)** – дружественная функция вывода строки.

**Класс TMyException.**

Класс **TMyException** содержит одно поле со спецификатором доступа **private**:

* **string str** – строка, хранящая сообщение об ошибке.

Далее идут методы класса со спецификатором доступа **public**:

* **TMyException(std::string \_str)** – конструктор инициализатор.
* **void what()** – метод вывода ошибки на экран.

# Описание алгоритмов.

**Преобразования математического выражения в польскую запись.**

В начале каждой арифметической операции даётся приоритет ( ‘(‘,’)’ – 1; ‘+’, ‘-‘ – 2; ‘\*’, ‘/’ – 3;). После чего, создается стек для хранения операций. Затем создается очередь для хранения самого математического выражения. Проходя по выражению слева направо, если встречаем операцию, то кладем ее в стек, если операнды выражения – они помещаются в очередь. Если проходя по выражению, мы встретили операцию с большим приоритетом, чем у той операции, которая на вершине стека, то она кладется в стек. В противном случае, элементы из стека последовательно переносятся в очередь. Этот процесс продолжается либо до опустошения стека, либо до того, как приоритет операции вершины стека будет не больше приоритета текущей операции. Как только эти действия прекращаются, текущая операция кладется в стек.

# Заключение.

В данной лабораторной работе мне удалось реализовать библиотеку для преобразования обыкновенных математических выражений в обратную польскую запись, а именно:

* Удалось реализовать вспомогательный класс строки **TMyString**.
* Удалось реализовать функции для преобразования и вычисления выражения в **ОПЗ**.
* Удалось протестировать функции **ОПЗ**, а также обеспечить их работоспособность.
* Удалось реализовать класс для обработки исключений **TMyException**.

Таким образом, я смог познакомится с понятием обратной польской записи математического выражения и реализовать библиотеку для преобразования и вычисления математический выражений в **ОПЗ**.

# Литература.

1. Ссылка из Википедии про Обратную польскую запись:

<https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C>

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015.