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

Голубева А. С.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2018.

**Содержание**

[1. Введение 3](#_Toc991355)

[2. Постановка задачи 4](#_Toc991356)

[3. Руководство пользователя 5](#_Toc991357)

[4. Руководство программиста 6](#_Toc991358)

[4.1 Описание структуры программы 6](#_Toc991359)

[4.2 Описание структур данных 6](#_Toc991360)

[4.3 Описание алгоритмов 7](#_Toc991361)

[5. Заключение 9](#_Toc991362)

[6. Литература 10](#_Toc991363)

1. **Введение**

Обратная польская запись (англ. Reverse Polish notation, RPN) — форма записи математических и логических выражений, в которой операнды расположены перед знаками операций.

В общем виде запись выглядит следующим образом:

* Запись набора операций состоит из последовательности операндов и знаков операций. Операнды в выражении при письменной записи разделяются пробелами.
* Выражение читается слева направо. Когда в выражении встречается знак операции, выполняется соответствующая операция над двумя последними встретившимися перед ним операндами в порядке их записи. Результат операции заменяет в выражении последовательность её операндов и её знак, после чего выражение вычисляется дальше по тому же правилу.
* Результатом вычисления выражения становится результат последней вычисленной операции.

Целью данной лабораторной работы является разработка программы преобразования арифметических выражений в польскую форму записи.

1. **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Реализация класса TString.
2. Разработка и реализация следующих функций:
3. GetPriority – получение приоритета операций;
4. IsOperation – проверка символа;
5. ConvertToPolish - преобразование выражения из инфиксной в постфиксную нотацию;
6. Res – вычисление конечного результата выражения.
7. Реализация класса TException для обработки исключений.
8. Реализация тестов и обеспечение их работоспособности.
9. Пример использования и обеспечение его работоспособности.
10. **Руководство пользователя**

Рассмотрим пример работы программы.

* При запуске программы пользователю предлагается ввести математическое выражение;
* Выражение переводится в обратную польскую запись;
* Затем вычисляется результат выражение и выводится на экран.

1. **Руководство программиста**
   1. **Описание структуры программы**

Программа состоит из следующих модулей:

* Модуль PolishLib – содержит заголовочный файл Polish.h, в котором определен интерфейс класса TString и описаны прототипы методов для работы с обратной польской записью. Также содержит файл Polish.cpp, в нем содержится реализация методов класса TString и методов, предназначенных для работы с обратной польской записью.
* Модуль Polish– содержит файл Polish\_main.cpp, в котором реализован пример использования программы;
* Модуль QueueLib – содержит реализацию класса TQueue;
* Модуль StackLib – содержит реализацию класса TStack;
* Модуль PolishTest – содержит файл PolishTest.cpp, в котором реализованы тесты;
* Модуль Exception – содержит заголовочный файл Exception.h, в котором реализован класс исключений, и файл Exception.cpp.
  1. **Описание структур данных**

**Класс TString**

Protected:

int size – длина строки;

char \*mas – массив элементов строки.

Public:

TString() – конструктор по умолчанию;

TString(TString &A) – конструктор копирования;

TString(char \*mas) - конструктор;

~TString() - деструктор;

int GetSize() – получить длину строки;

TString operator + (TString &A) – перегрузка оператора сложения;

TString & operator = (TString &A) - перегрузка оператора присваивания;

char & operator [] (int i) - перегрузка оператора индексации;

friend ostream & operator << (ostream &ostr, TString &A);

friend istream& operator >> (istream &istr, TString &A);

Методы для работы с обратной польской записью:

int GetPriority(const char op) – получить приоритет операции;

bool IsOperation(char с) – проверить символа;

TQueue<char> ConvertToPolish(TString str) – преобразовать в обратную

польскую запись;

double Res(TQueue<char> Q) – получить конечный результат;

* 1. **Описание алгоритмов**

**Преобразование выражения из инфиксной формы в польскую запись.**

Стек St используется для хранения операций и открывающихся скобок, для хранения выражения в обратной польской записи используется очередь Q. Для операций вводится приоритет:

* «+», «-» – приоритет 2
* «\*», «/» – приоритет 3
* «(», «)» – приоритет 1
* «=» – приоритет 0

При появлении закрывающейся скобки элементы из стека St переносятся в очередь Q. Продолжаем пока стек St не пуст, либо не появилась открывающаяся скобка.

Если приоритет текущей операции p имеет приоритет больше приоритета операции на вершине стека St, то она кладется в стек St. Иначе элементы из стека St переносятся в очередь Q. Продолжаем пока стек St не пуст, либо приоритет операции вершины стека будет не больше приоритета текущей операции p. После того, как эти действия завершаются текущая операция p кладется в стек St.

1. **Заключение**

В результате лабораторной работы была разработана программа, преобразовывающая арифметические выражения в обратную польскую запись и вычисляющая результат этих выражений.

Успешно выполнены тесты, проверяющие работоспособность программы. Приведен пример использования программы.

1. **Литература**
   1. Википедия: свободная электронная энциклопедия: на русском языке:

URL: <https://ru.wikipedia.org/wiki/Обратная_польская_запись>

(Дата обращения: 15.02.2019)

* 1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015.
  2. Национальный открытый университет «Интуит». Лекция 31: Динамические структуры данных: очередь и стек.

URL: <https://www.intuit.ru/studies/courses/648/504/lecture/11457?page=2>

(Дата обращения: 15.02.2019)