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 ЛАБОРАТОРНОЙ РАБОТЕ**

**«Структура хранения данных: Обратная польская запись»**

**Выполнил:** студент группы 381706-2

Мышкин Андрей Александрович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Руководитель:**

ассистент кафедры МОСТ ИИТММ,

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018

**Содержание**

1. Введение………………………………………………………………………...3

2. Постановка задачи……………………………………………………...………4

3. Руководство пользователя…………………………………………...………...5

4. Руководство программиста…………………………………………...………..7

4.1. Описание структур данных…..……………………………………….7

4.2. Описание алгоритмов………………………………………………….7

4.3. Описание структуры программы….……………………………..…...8

6. Заключение……………………………………………………………………..9

7. Литература……………….…………………………………………...……….10

**Введение**

Обратная польская запись (англ. Reverse Polish notation, RPN) — 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.

Стековой машиной называется алгоритм, проводящий вычисления по обратной польской записи.

Отличительной особенностью обратной польской нотации является то, что все аргументы расположены перед знаком операции. 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.

Например, рассмотрим вычисление выражения 7 2 3 \* − (эквивалентное выражение в инфиксной нотации: 7 − 2 \* 3).

1. Первый по порядку знак операции — «\*», поэтому первой выполняется операция умножения над операндами 2 и 3 (они стоят последними перед знаком). Выражение при этом преобразуется к виду 7 6 − (результат умножения — 6, — заменяет тройку «2 3 \*»).
2. Второй знак операции — «−». Выполняется операция вычитания над операндами 7 и 6.
3. Вычисление закончено. Результат последней операции равен 1, это и есть результат вычисления выражения.

Очевидное расширение обратной польской записи на унарные, тернарные и операции с любым другим количеством операндов: при использовании знаков таких операций в вычислении выражения операция применяется к соответствующему числу последних встретившихся операндов.

**Постановка задачи**

Основная задача данной лабораторной работы – это создание и реализация структуры хранения данных такой, как обратная польская запись. Также разработка программы тестирования работы данной структуры, в которой будут проведен подсчет заданного арифметического выражения.

-

**Руководство пользователя**

При запуске тестирования программа просит пользователя вести его арифметическое выражение на консоль. Особенности записи в том, чтобы не было пропусков между числами и знаками арифметических операций, иначе программа может неправильно считать выражение.

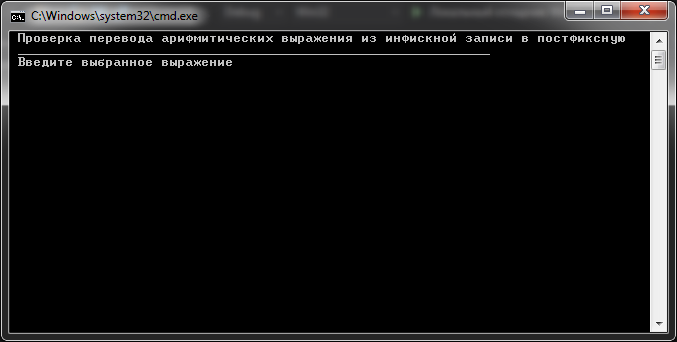


Рисунок 1. Тестирование работы обратной польской записи. Начальный этап

После записи выражения необходимо поставить знак «=», чтобы подтвердить окончание выражения.

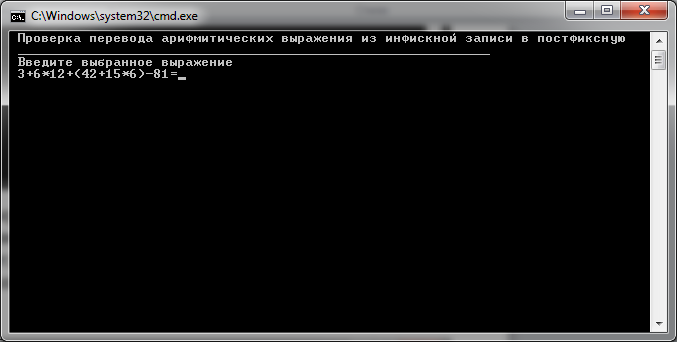


Рисунок 2. Тестирование работы обратной польской записи. Записанное выражение

Далее на экран будет повторно выведено вписанное выражение для проверки, а за ним уже это же выражение в обратной польской записи. Для проверки правильности алгоритмов в конце будет произведен подсчет указанного арифметического выражения.

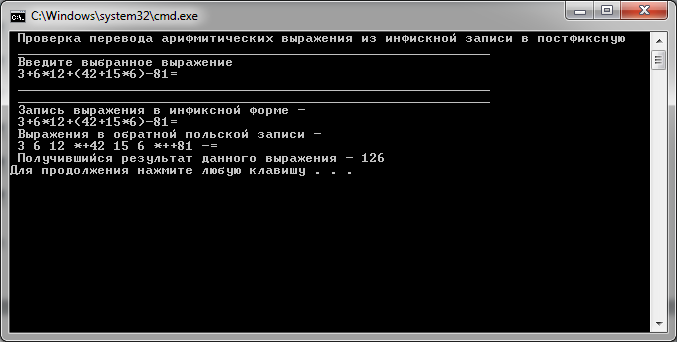


Рисунок3. Тестирование работы обратной польской записи. Калькулятор и обратная польская запись

**Руководство программиста**

**Описание структур данных.**

**Класс TInfixToPolish**

Полей у этого класса нет, так как к нему подключены библиотеки с классами стек и стек на списке, на которых построена программа.

Методы

int GetOpePrt(char ope) – метод класса, возвращающий приоритет указанной операции. Операция определяется переменной ope

int OpeMark(char ope) – метод класса, делающий проверку на знак операции, то есть определяющий её

double Calculator(char\* memory, int length) – метод класса, производящий подсчет данного арифметического выражения

char\* ConversionToPolish(char\* exp, int length) – метод класса, делающий преобразование обычной записи выражения к обратной польской записи

**Описание алгоритмов.**

Описание работы преобразования арифметического выражения в обратную польскую запись.

Создаем два стека, один для хранения операций, которые указаны в выражение(ope), и второй как конченый вариант записи выражения(polish).

В цикле пока не дойдем до конца

Если элемент выражения не операция, тогда добавляем элементы в polish, пока не встретится знак. Как только получаем знак, добавляем в стек polish пробел(«\_»).

Иначе если получили открывающую скобку(«(»), добавляем её в стек ope.

Иначе если получили закрывающую скобку(«)»), то извлекаем символы из стека ope до тех пор, пока не встретим в стеке открывающуюся скобку

Если мы получили в виде символа знак операции, тогда проверяем приоритет этой операции. Операции умножения и деления имеют наивысший приоритет. Операции сложения и вычитания меньший приоритет. Наименьший приоритет имеет открывающаяся скобка.

Если стек ope все ещё пуст или символы в нем имеют меньший приоритет, чем приоритет текущего символа, то помещаем этот символ в стек.

Если же символ на вершине стека имеет приоритет выше или равного нашего приоритета, то извлекаем символы из стека ope в стек polish до тех пор, пока выполняется это условие. После этого возвращаемся к предыдущему пункту.

**Описание структуры программы.**

infixtopolish – модуль для тестирования работы структуры данных типа обратная польская запись. Главным, файлом которого является main.cpp

infixtopolishlib – модуль для хранения реализации данного класса. В нем содержится заголовочный файл – TInfixPolish.h, и файл с реализацией методов класса – TInfixPolish.cpp

infixtopolish test – модуль, содержащий тесты данного класса для прохождения их с помощью использования Google C++ Testing Framework. В файле infixtopolish \_tests.cpp содержатся написанные тесты, а в test\_main.cpp код, запускающий тестирование

**Заключение**

В ходе проведение данной лабораторной работы была создана и протестирована такая структура хранения данных как обратная польская запись. В данном классе были реализованы все необходимые методы для комфортной работы Были усвоены все тонкости, с которыми пришлось столкнуться при выполнении данной лабораторной работы.

Также были освоены инструменты разработки программного обеспечения, как система контроля версий Git и фрэймворк для разработки автоматических тестов Google Test.

**Литература**

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015
2. Обратная польская запись. Википедия - <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>