Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение высшего образования "Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского" (ННГУ)

Институт информационных технологий, математики и механики

Отчет по учебной практике

**Алгоритм преобразования выражения из инфиксной формы в постфиксную и вычисления выражения по постфиксной форме**

Выполнил:

Студент группы 381806-3

Макарычев С.Д.

Проверил:

Доцент кафедры МОСТ ИИТММ

Кустикова В.Д.

Нижний Новгород

2018 г.

**Содержание**

[Введение 4](#_Toc24378315)

[1 Постановка задачи 5](#_Toc24378316)

[2 Руководство пользователя 6](#_Toc24378317)

[3 Руководство программиста 8](#_Toc24378318)

[3.1 Структура программы 8](#_Toc24378319)

[3.2 Описание алгоритмов 8](#_Toc24378320)

[3.2.1 Алгоритм преобразования инфиксной формы в постфиксную 8](#_Toc24378321)

[3.2.2 Алгоритм вычисления выражения по постфиксной форме при определенных значениях операндов. 9](#_Toc24378322)

[4 Описание функций 10](#_Toc24378323)

[4.1 Функции класса TStack 10](#_Toc24378324)

[4.2 Функции класса Postfix 10](#_Toc24378325)

[4.3 Функции класса Exception 11](#_Toc24378326)

[Заключение 12](#_Toc24378327)

# Введение

Постфиксная запись представляет собой такую запись арифметического выражения, в которой сначала записываются операнды, а затем – знаки операции. Постфиксная форма позволяет избавиться от скобок, присутствующих в инфиксной форме, тем самым помогая избежать неоднозначности выполнения действий.

В данной работе представлена реализация алгоритма, позволяющего преобразовать выражение в постфиксную форму и вычислить полученное в постфиксной форме выражение.

# Постановка задачи

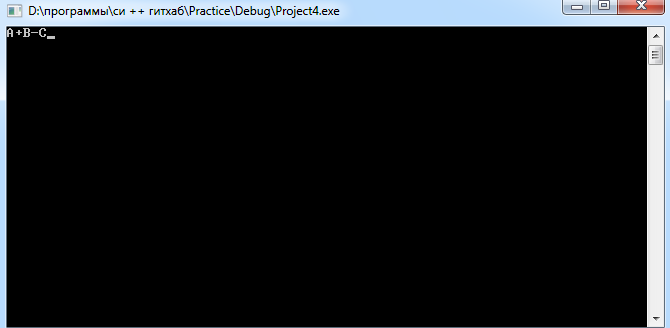
В данной работе необходимо, используя структуру данных “Стек”, реализовать программу, позволяющую получать из выражения в инфиксной форме выражение в постфиксной форме и вычислять значение этого выражения

Входные данные: выражение в инфиксной форме; значения переменных, входящих в это выражение.

Выходные данные: выражение в постфиксной форме; значение выражения при определенных значениях переменных.

# Руководство пользователя

После запуска программы пользователю необходимо заглавными буквами латинского алфавита ввести выражение, которое он хочет преобразовать. Вводим в поле наше выражение и нажимаем Enter.(рис.1)



1. Ввод выражения

Если выражение введено корректно, то в консоли появится выражение в постфиксной форме.(рис.2)

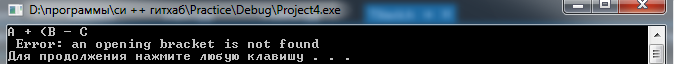


1. Вывод выражения в постфиксной форме

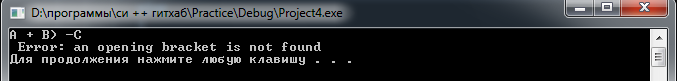
Если же выражение введено неправильно, то сработает одна из исключительных ситуаций. (рис.3), (рис.4), (рис.5), (рис.6)



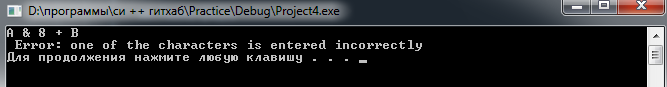
1. Исключительная ситуация: несколько переменных стоят рядом друг с другом



1. Исключительная ситуация: в выражении не найдена закрывающаяся скобка

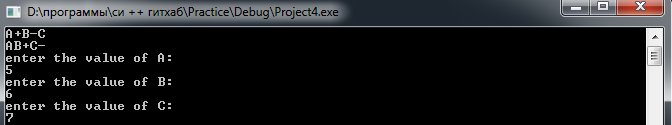


1. Исключительная ситуация: в выражении не найдена открывающаяся скобка



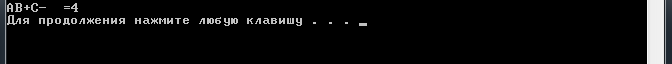
1. Исключительная ситуация: в выражении присутствует недопустимый символ

Далее после вывода выражения в постфиксной форме пользователю нужно ввести значения переменных, входящих в выражение.(рис.7)



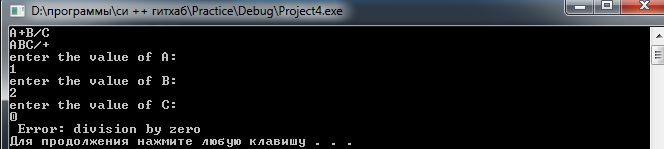
1. Ввод значений переменных.

После этого в консоль выведется значение исходного выражения. (рис.8)



1. Вывод значения выражения при заданных параметрах.

Если в исходном выражении присутствует частное, и делитель в нем равен нулю, то срабатывает исключительная ситуация: деление на ноль.



1. Исключительная ситуация: деление на ноль.

# Руководство программиста

## Структура программы

Данная программа состоит из 5 файлов: TStack.h, в котором описан шаблонный класс, реализующий структуру данных “Стек”; файла Postfix.h, в котором описан класс со статическими методами, выполняющими алгоритмы преобразования; файла Postfix.cpp с реализациями статических функций класса; файла Exception.h с классом для обработки исключений; файла main.cpp, в котором реализуется основная программа.

## Описание алгоритмов

### Алгоритм преобразования инфиксной формы в постфиксную

Алгоритм представлен в виде функции, принимающей на вход строку с выражением в инфиксной форме и возвращающей строку в постфиксной форме.

Сначала создается 2 стека: в первом стеке будут храниться операнды, во втором – операции. Далее всем операциям присваивается приоритет: “\*” и “/” – приоритет 3; “+” и “-“ – приоритет 2; “(“ – приоритет 1.

Поступившая на вход строка начинает просматриваться слева направо с первого элемента. Если пришел операнд, то мы кладём его в первый стек. Если пришла операция, то если второй стек пуст, кладем операцию в данный стек. Если же во втором стеке уже имеются элементы, то сравниваем приоритет верхнего элемента стека и операции, которую мы хотим добавить. Если приоритет верхнего элемента стека больше или равен приоритета рассматриваемой операции, то пока это условие выполняется, мы удаляем элементы из второго стека и вставляем в первый стек.

Если пришла открывающаяся круглая скобка, то мы кладем ее во второй стек. Если пришла закрывающаяся круглая скобка, то мы начинаем перекладывать элементы из второго стека в первый, пока не встретится открывающаяся круглая скобка, ее мы из второго стека удаляем.

Когда дошли до конца исходной строки, начинаем перекладывать элементы из второго стека, начиная с самого верхнего и заканчивая самым нижним элементом, в первый. Из второго стека перекладываем элементы, начиная с самого нижнего, в новую строку. Таким образом, в новой строке содержится выражение в постфиксной форме.

1. Пример пошагового выполнения алгоритма для выражения: (A+B)\*C

|  |  |  |
| --- | --- | --- |
| Обрабатываемая лексема | Стек 1 | Стек 2 |
| ( |  | ( |
| A | A | ( |
| + | A | ( + |
| B | A B | ( + |
| ) | A B + | ( |
| ) | A B + |  |
| \* | A B + | \* |
| C | A B + C | \* |
|  | A B + C \* |  |

### Алгоритм вычисления выражения по постфиксной форме при определенных значениях операндов.

Алгоритм представлен в виде функции, принимающей на вход строку с выражением в постфиксной форме, множество операндов, входящих в это выражение, множество значений этих операндов и возвращающей значение выражения.

Сначала создается вспомогательный стек для хранения значений некоторой части выражения.

Постфиксная форма просматривается слева направо. Если пришел операнд, то выполняется поиск значения этого операнда, и значение кладется в стек. Если пришла операция, то вынимаем с вершины стека 2 значения и выполняем операцию в порядке 2, 1, где 1-верхнее значение из стека, 2-значение идущее после верхнего элемента стека. Результат операции кладется в стек.

Если дошли до конца выражения, то в стеке хранится результат вычисления выражения для заданных значений операндов.

1. Пример пошагового выполнения алгоритма для постфиксного выражения A B + C \* при A = 5, B = 4, C = 3.

|  |  |  |
| --- | --- | --- |
| Обрабатываемая лексема | Стек | Выражение |
| A | 5 |  |
| B | 5 4 |  |
| + | 9 | 4 + 5 |
| C | 9 3 |  |
| \* | 27 | 3 \* 9 |

# Описание функций

## Функции класса TStack

**Функция “**IsEmpty”

Описание: проверяет стек на пустоту.

Ничего не принимает. Возвращает 1, если стек пуст, 0, если в стеке имеются элементы.

**Функция “**IsFull”

Описание: проверяет стек на переполнение.

Ничего не принимает. Возвращает 1, если стек полон, 0, если стек не полон.

**Функция “**Push”

Описание: выполняет вставку элемента наверх стека.

Принимает элемент, который нужно вставить в стек. Ничего не возвращает.

**Функция “**Get\_top”

Описание: позволяет узнать индекс верхнего элемента стека.

Ничего не принимает. Возвращает индекс верхнего элемента стека.

**Функция “**GetElem”

Описание: позволяет получить элемент из стека с определенным индексом.

Принимает индекс элемента, к которому нужно обратиться. Возвращает элемент с указанным индексом.

**Функция “**Pop”

Описание: удаляет верхний элемент из стека.

Ничего не принимает. Возвращает удаленный элемент.

**Функция “**Top”

Описание: позволяет узнать верхний элемент стека.

Ничего не принимает. Возвращает верхний элемент стека.

## Функции класса Postfix

**Статическая Функция “**PostfixForm”

Описание: преобразует инфиксное выражение в постфиксное.

Принимает строку, содержащую выражение в инфиксной форме. Возвращает строку, содержащую выражение в постфиксной форме.

**Статическая Функция “**Calculating”

Описание: вычисляет значение выражения по постфиксной форме.

Принимает строку, содержащую выражение в постфиксной форме, строку с операндами этого выражения, массив значений этих операндов. Возвращает значение выражения.

**Статическая Функция “**Prioritet”

Описание: присваивает приоритет операциям.

Принимает операцию из строки. Возвращает приоритет операции.

## Функции класса Exception

**Функция “what”**

Описание: переопределяет функцию класса потомка exception, определяет сработавшее исключение.

Ничего не принимает. Возвращает строку с ошибкой.