

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Информационные Системы**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Списочные структуры**

Студент гр. 1373

\_\_\_\_\_

Кабернюк И.С.

Преподаватель

\_\_\_\_\_

Бондаренко Б.Е.

Санкт-Петербург

2022

### **Цель работы.**

Реализовать следующие структуры: односвязный список, динамический массив и стек. Стек можно реализовать как на базе списка, так и отдельно. Использовать стек для реализации алгоритма сортировочной станции. Разрешённые символы в исходном выражении: +, -, \*, /, ^, sin, cos, (, ), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Для упрощения разбиения входной строки на токены разрешается отделять каждый символ пробелом.

## Ход работы

### 1) Динамический массив (вектор)

Для создания динамического массива создаем класс Vector

Стандартный конструктор Vector()

Перегруженный Vector(const unsigned input\_size)

Функции класса:

- 1) Обращение по индексу Vector::operator[](int position)
- 2) Вывод массива Vector::print()
- 3) Добавление в конец Vector::push\_back(const int last\_obj)
- 4) Добавление по индексу Vector::add(const unsigned position, const int value)
- 5) Удаление по индексу Vector::erase(const unsigned position)
- 6) Поиск по индексу Vector::find(const unsigned position)
- 7) Размер массива Vector::get\_size()

### 2) Список (лист)

Для создания списка создаем класс List

В private поле создается еще один класс Node

Функции класса:

- 1) Обращение по индексу List::operator[](int position)
- 2) Добавление в конец List::push\_back(int data)
- 3) Размер списка List::GetSize()
- 4) Поиск по индексу List::find(const unsigned position)
- 5) Вывод массива List::print()
- 6) Очистить список List::clear()
- 7) Добавление в начало List::push\_front(const int data)
- 8) Добавление по индексу List::add(const unsigned position, const int data)
- 9) Удаление по индексу List::ereas(const unsigned position)

### 3) Стэк

Стэк реализован на базе списка (Лист).

Функции класса:

- 1) Очистить список void clear()
- 2) Размер списка GetSize()
- 3) Вывод массива print()
- 4) Добавление в стэк push(std::string value)
- 5) Удаление из стэка pop()
- 6) Показать верхний элемент стэка peak()

#### 4) Алгоритм сортировочной станции

Пока есть ещё символы для чтения:

Читаем очередной символ.

Если символ является числом, добавляем его к выходной строке..

Если символ является открывающей скобкой, помещаем его в стек.

Если символ является закрывающей скобкой:

До тех пор, пока верхним элементом стека не станет открывающая скобка, выталкиваем элементы из стека в выходную строку. При этом открывающая скобка удаляется из стека, но в выходную строку не добавляется. Если стек закончился раньше, чем мы встретили открывающую скобку, это означает, что в выражении либо неверно поставлен разделитель, либо не согласованы скобки.

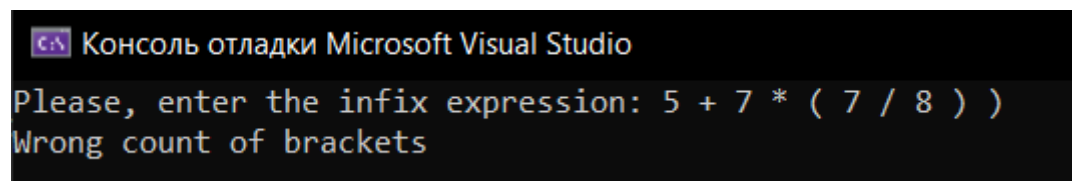
Если символ является оператором *oI*, тогда:

1) пока приоритет *oI* меньше либо равен приоритету оператора, находящегося на вершине стека выталкиваем верхние элементы стека в выходную строку;

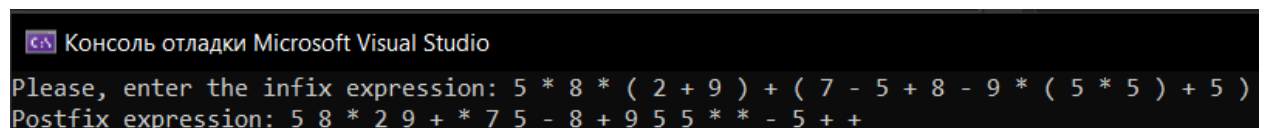
2) помещаем оператор *oI* в стек.

Когда входная строка закончилась, выталкиваем все символы из стека в выходную строку. В стеке должны были остаться только символы операторов; если это не так, значит в выражении не согласованы скобки.

Демонстрация работы программы:



```
Консоль отладки Microsoft Visual Studio
Please, enter the infix expression: 5 + 7 * ( 7 / 8 ) )
Wrong count of brackets
```



```
Консоль отладки Microsoft Visual Studio
Please, enter the infix expression: 5 * 8 * ( 2 + 9 ) + ( 7 - 5 + 8 - 9 * ( 5 * 5 ) + 5 )
Postfix expression: 5 8 * 2 9 + * 7 5 - 8 + 9 5 5 * * - 5 + +
```

### **Вывод**

В результате работы были реализованы следующие структуры данных:

- 1) Односвязный список;
- 2) Динамический массив;
- 3) Стек.

На основе реализованных структур, был написан алгоритм сортировочной станции, переводящий выражение из префиксной нотации в постфиксную.