

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

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

Студентка гр. 1373

Марков М. М.

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.

Ход работы

1. Односвязный список

Для реализации структуры односвязного списка был создан шаблонный класс LinkedList. В приватное поле класса входит:

- 1) Структура узла. Содержит шаблон данных и указатель на следующий узел;
- 2) Размер списка;
- 3) Указатель на первый элемент списка.
- 4) Указатель на последний элемент списка

Реализованные функции класса:

- 1) Деструктор;
- 2) Вставка элемента в конец списка;
- 3) Вставка элемента по индексу;
- 4) Поиск индекса элемента по его значению;
- 5) Удаление элемента по индексу;
- 6) Получение размера списка;
- 7) Очистка списка с удалением всех элементов
- 8) Слияние двух списков в один копированием элементов
- 9) Обращение к элементу по индексу

2. Динамический массив

Для реализации динамического массива был создан класс `DynamicArray`. В приватное поле класса входит:

- 1) Указатель на массив данных;
- 2) Количество элементов в массиве;
- 3) Вместимость массива;
- 4) Множитель увеличения вместимости массива
- 5) Функция расширения размера массива.

Реализованные функции класса:

- 1) Деструктор;
- 2) Добавление элемента по индексу;
- 3) Добавление элемента в конец;
- 4) Получение индекса элемента по данным;
- 5) Получение элемента по индексу;
- 6) Удаление элемента по индексу;
- 7) Получение размера массива;
- 8) Получение вместимости массива;
- 9) Очистка массива;
- 10) Слияние двух массивов;
- 11) Сужение вместимости под количество элементов;
- 12) Изменение вместимости массива по пользовательским параметрам;
- 13) Смена элементов по индексам;

3. Стек

Для реализации стека был создан класс `Stack`. Данный класс был реализован на основе ранее реализованного односвязного списка. В приватное поле класса входит переменная типа `LinkedList`, так же шаблонная.

Реализованные функции:

- 1) Снятие элемента с вершины стека;
- 2) Запись элемента на вершину стека;
- 3) Получение элемента на вершине стека;
- 4) Получение размера стека.

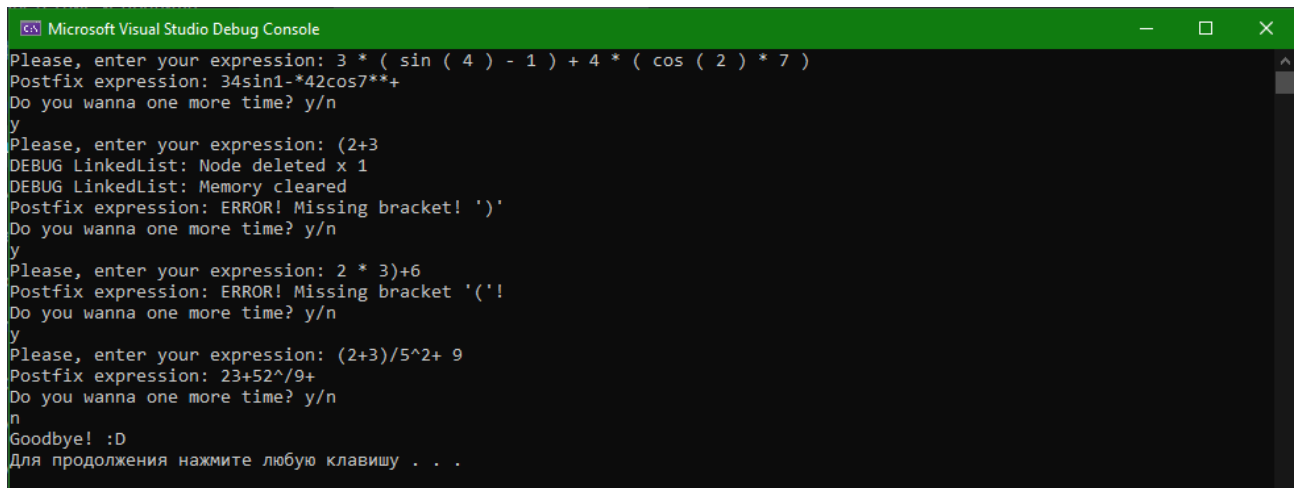
4. Алгоритм сортировочной станции

Реализация была выполнена в файле Shunting-yard.cpp

Реализация алгоритма была взята из интернета, в частности из Википедии.

В качестве вспомогательных функций были реализованы checkPriority (проверяет приоритет операции по отношению к другой) и isOperator (проверяет, является ли токен оператором). В данном случае, программе не важно, есть ли пробелы между токенами, проверка выполняется как с ними, так и без. Алгоритм предусматривает проверку на отсутствие или излишек скобок и сообщает об этом.

Ниже – пример работы программы на различных входных данных:



```
Microsoft Visual Studio Debug Console
Please, enter your expression: 3 * ( sin ( 4 ) - 1 ) + 4 * ( cos ( 2 ) * 7 )
Postfix expression: 34sin1-*42cos7**+
Do you wanna one more time? y/n
y
Please, enter your expression: (2+3
DEBUG LinkedList: Node deleted x 1
DEBUG LinkedList: Memory cleared
Postfix expression: ERROR! Missing bracket! ')'
Do you wanna one more time? y/n
y
Please, enter your expression: 2 * 3)+6
Postfix expression: ERROR! Missing bracket '('!
Do you wanna one more time? y/n
y
Please, enter your expression: (2+3)/5^2+ 9
Postfix expression: 23+52^/9+
Do you wanna one more time? y/n
n
Goodbye! :D
Для продолжения нажмите любую клавишу . . .
```

Выводы

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 инфиксной записи в постфиксную.