

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский

Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных
технологий

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

«Реализация класса стек»

Выполнил: студент группы
381706-2

Крюков Дмитрий Алексеевич

_____ Подпись

Научный руководитель:

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

_____ Подпись

Нижний Новгород

2018

Содержание

| | |
|---------------------------------------|---|
| Содержание..... | 2 |
| 1. Введение..... | 3 |
| 2. Постановка задачи..... | 4 |
| 3. Руководство пользователя..... | 5 |
| 4. Руководство программиста..... | 6 |
| 4.1 Описание структуры программы..... | 6 |
| 4.2 Описание структур данных..... | 6 |
| 4.3 Описание алгоритмов..... | 7 |
| 5. Заключение..... | 8 |
| 6. Литература..... | 9 |

1. Введение

Стек — это структура данных, которая работает по принципу **FILO** (first in — last out; первый пришел — последний ушел).

В стеке элемент, который вошел самый первый — выйдет самым последним. Получается, если вы добавили три элемента в стек первым будет удален последний добавленный элемент.

Программный вид стека используется для обхода структур данных, например, дерево или граф. При использовании рекурсивных функций также будет применяться стек, но аппаратный его вид. Кроме этих назначений, стек используется для организации стековой машины, реализующей вычисления в обратной инверсной записи.

Применение стека упрощает и ускоряет работу программы, так как идет обращение к нескольким данным по одному адресу.

2. Постановка задачи

Реализация класса стек:

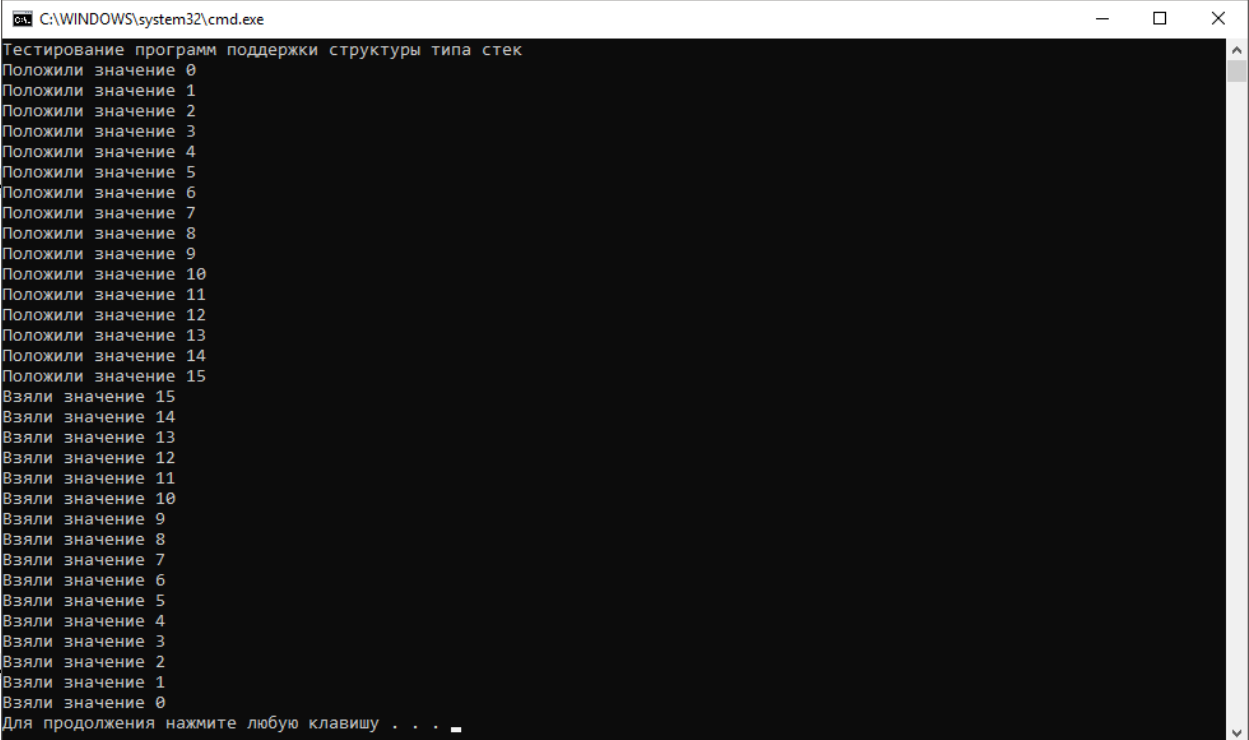
Для стека определены операции:

- проверка пустоты
- проверка полноты
- добавить элемент
- извлечь последний элемент

3. Руководство пользователя

Данная программа предназначена для тестирования динамической структуры стек. В стек поочередно загружаются 16 элементов, а затем они извлекаются из стека и выводятся в консоли.

4. Руководство программиста



```
C:\WINDOWS\system32\cmd.exe
Тестирование программ поддержки структуры типа стек
Положили значение 0
Положили значение 1
Положили значение 2
Положили значение 3
Положили значение 4
Положили значение 5
Положили значение 6
Положили значение 7
Положили значение 8
Положили значение 9
Положили значение 10
Положили значение 11
Положили значение 12
Положили значение 13
Положили значение 14
Положили значение 15
Взяли значение 15
Взяли значение 14
Взяли значение 13
Взяли значение 12
Взяли значение 11
Взяли значение 10
Взяли значение 9
Взяли значение 8
Взяли значение 7
Взяли значение 6
Взяли значение 5
Взяли значение 4
Взяли значение 3
Взяли значение 2
Взяли значение 1
Взяли значение 0
Для продолжения нажмите любую клавишу . . .
```

Структура

Поля:

mem - память для элементов стека

top - индекс последнего занятого в mem

memSize — размер памяти под стек

Методы:

`int IsEmpty()` - контроль пустоты

`int GetCount()` - число элементов в стеке

`int IsFull()` - контроль переполнения

`void Put(const int Val)` - добавить значение

`ValType Get()` - извлечь значение

4.3 Описание алгоритмов

Добавление в стек элемента **a**

Если стек полон

Ошибка

положить **a** в `mem[top]`

увеличить `top` на 1

Контроль переполнения

Вернуть булево значения равенства $top - 1 = memSize$

5. Заключение

В ходе работы был реализован класс стек, в нем реализованы функции контроля пустоты и переполнения, а так же добавления и извлечения элементов

6. Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Стек | программирование на С и С++ - Режим доступа:
<https://codelessons.ru/cplusplus/realizaciya-steka-stack-v-c.html>