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 учебной практике**

**«Реализация класса стек»**

**Выполнил:** студент группы 381706-2

Банденков Даниил Викторович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

# Содержание

[Содержание 2](#_Toc534333789)

[1 Введение 3](#_Toc534333790)

[2 Постановка задачи 4](#_Toc534333791)

[3 Руководство пользователя 5](#_Toc534333792)

[4 Руководство программиста 6](#_Toc534333793)

[4.1 Описание структуры программы 6](#_Toc534333794)

[4.2 Описание структур данных 6](#_Toc534333795)

[4.3 Описание алгоритмов 6](#_Toc534333796)

[5 Заключение 8](#_Toc534333797)

[6 Литература 9](#_Toc534333798)

# Введение

Стек (англ. stack — стопка; читается стэк) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Принцип работы стека сравнивают со стопкой листов бумаги: чтобы взять второй сверху, нужно снять верхний.

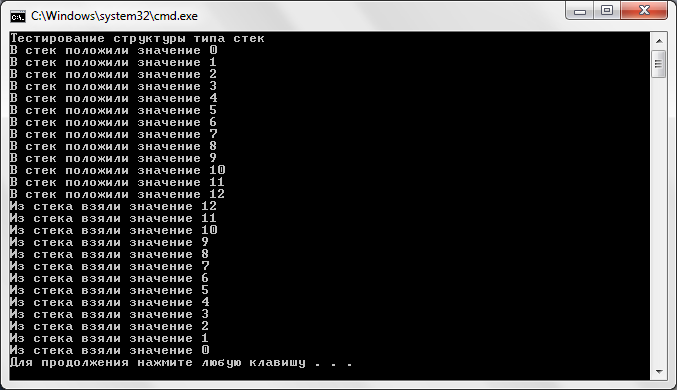
В стеке нет индексов как в массиве, а значит, вы не можете обратиться к определенному элементу. Все потому что, стек построен на связных списках. Зачем тогда использовать связные списки, если с таким же успехом можно было использовать простой массив? Все достоинство шаблонного стека заключается в добавлении и удалении элементов. Эти операции происходят за константное время (это хороший плюс).

# Постановка задачи

В данной работе требуется создать класс стек, в нем реализовать функции контроля пустоты и переполнения, а так же добавления и извлечения элементов.

# Руководство пользователя

Данная программа тестирует динамическую структуру - стек. В список поочередно загружаются 13 элементов, а затем они извлекаются из стека и выводятся в консоли (см Рис 1).

Рис 1.Пример использования программы.

# Руководство программиста

## Описание структуры программы

*В решении содержатся следующие модули:*

1. stacklib (TStack.h, TStack.cpp)– модуль с классом, реализующий структуру стек.
2. stack(main.cpp)– модуль реализующий тестирование динамической структуры стек.
3. stacktest(stack\_test.cpp)модуль тестирование класса стек при помощи Google C++ Testing Framework. .

## Описание структур данных

*Поля:*

mem - память для элементов структуры.

top - индекс последнего занятого в mem.

memSize — размер памяти под структуру.

*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.

## Описание алгоритмов

*Извлечение из стека элемента (псевдокод)*

Если:

Стек пуст

Ошибка

Вернуть mem[top--]

*Наглядный алгоритм*

На рисунке 2 изображены 6 чисел: 6, 5, 1, 2, 5, 9. Извлекать их будем в таком же порядке.

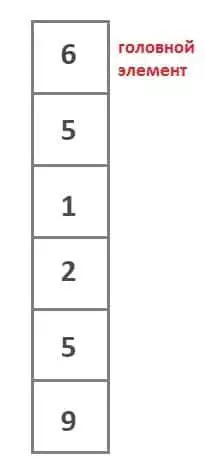


Рисунок 2.

Например чтобы извлечь число 1 нам придется сначала извлечь числа 6 и 5, а потом уже 1. Так как это стек, эти числа мы добавляли в обратном порядке. Если быть точным вот так: 9, 5, 2, 1, 5, 6..

# Заключение

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

# Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. <https://codelessons.ru/cplusplus/realizaciya-steka-stack-v-c.html>

(Дата обращения 02.01. 2019)

1. <https://ru.wikipedia.org/wiki/Стек> (Дата обращения 02.01. 2019)