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**

**«Структура данных: Стек на массиве»**

**Выполнил:** студент группы 381706-1

Власов Андрей Сергеевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc532665011)

[2. Постановка задачи 4](#_Toc532665012)

[3. Руководство пользователя 5](#_Toc532665013)

[4. Руководство программиста 6](#_Toc532665014)

[4.1 Описание структуры программы 6](#_Toc532665015)

[4.2 Описание структур данных 6](#_Toc532665016)

[4.3 Описание алгоритмов 7](#_Toc532665017)

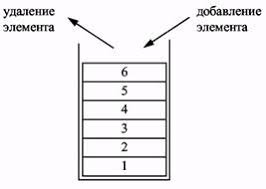
[5. Заключение 8](#_Toc532665018)

[6. Литература 9](#_Toc532665019)

# Введение

Стек (от англ. stack – стопка) — структура данных, представляющая из себя упорядоченный набор элементов, в которой добавление новых элементов и удаление существующих производится с одного конца, называемого вершиной стека. Притом первым из стека удаляется элемент, который был помещен туда последним, то есть в стеке реализуется принцип LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

В стеке мы имеем доступ только к последнему добавленному элементу. Принцип работы стека сравнивают со стопкой листов бумаги: чтобы взять второй сверху лист, нужно снять верхний.



*Рисунок 1 Схема стека*

Цель данной лабораторной работы: реализация стека на массиве.

# Постановка задачи

В данной лабораторной работе нам необходимо осуществить:

1. Разработку и реализацию класса стека.
2. Создание класса для обработки исключений.
3. Пример использования класса стека.
4. Набор автоматических тестов.

# Руководство пользователя

Рассмотрим пример написания глобальной функции main:

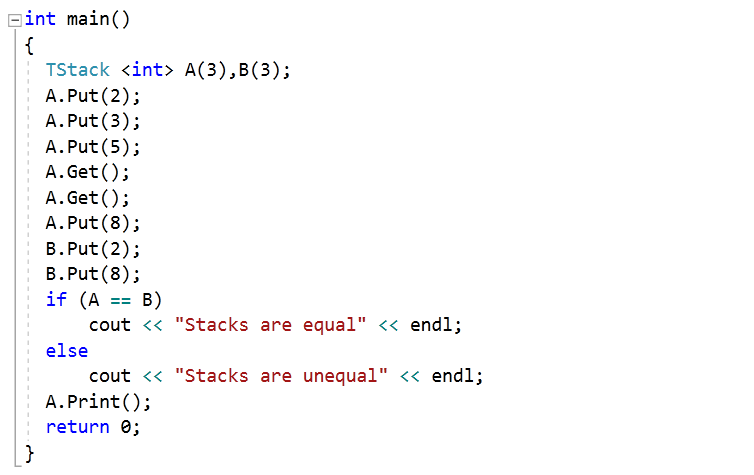


Рисунок 2 Пример использования класса TStack

В данном случае мы создаем два стека с максимальным размером 3, добавляем и удаляем элементы, сравниваем два стека и выводим один из них.

# Руководство программиста

## 4.1 Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Stack. Содержит пример использования стека.
* Модуль StackLib – статическая библиотека. Содержит файл Stack.h, в котором описан интерфейс и реализация шаблонного класса *TStack*.
* Модуль StackTest. Содержит набор тестов, разработанных с помощью использования Google C++ Testing Framework.
* Модуль MyExceptionsLib – библиотека исключений.

## Описание структур данных.

#### Класс TStack – стек.

template <class T> class TStack – класс стека является шаблонным

Он содержит 3 поля со спецификатором доступа protected:

1. int size – максимальный размер стека.
2. int top – позиция вершины стека.
3. T\* mas – указатель на область памяти, выделяемую под хранение стека.

А также 2 конструктора, 1 деструктор, 6 методов и 3 перегруженных оператора со спецификатором доступа public:

* TStack(int \_size = 0) – конструктор инициализации.
* TStack(TStack<T> &A) – конструктор копирования.
* ~TStack() – деструктор.
* void Put(T e) – метод добавления нового элемента в стек.
* T Get() – метод изъятия элемента из вершины стека с удалением.
* void Print() – метод вывода элементов стека.
* bool IsFull() – метод проверки стека на полноту.
* bool IsEmpty() – метод проверки стека на пустоту.
* *int operator!=(const TStack<T>& stack) const* – оператор сравнения на неравенство стеков.
* *int operator==(const TStack<T>& stack) const* – оператор сравнения на равенство стеков.
* *TStack& operator=(const TStack<T>& stack)* – оператор присваивания.

## Описание алгоритмов

Рассмотрим подробнее алгоритмы добавления и удаления элемента:

1. **Добавление элемента в стек.**

Сначала проверяем стек на полноту, если он неполный, то размещаем элемент в первую свободную ячейку массива, выделенного под хранение данных. На эту ячейку указывает параметр top – вершина стека. Затем значение top увеличиваем на 1.

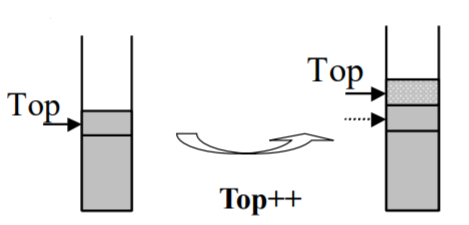


Рисунок 3 Добавление элемента в стек.

1. **Удаление элемента из стека.**

Сначала проверяем стек на пустоту, если он непустой, то уменьшаем значение topна 1 – спускаемся к последнему существующему элементу и возвращаем его значение. Само значение в ячейке с индексом, по которому производилось возращение никак не затирается, т.к. оно будет перезаписано при следующем добавлении другого элемента.

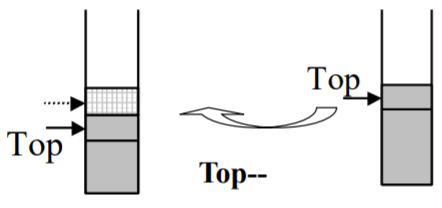


Рисунок 4 Удаление элемента из стека

# Заключение

В данной лабораторной работе была выполнена реализация такой структуры данных как стек на массиве. А именно была разработана библиотека, реализующая шаблонный класс стека, был создан класс для обработки исключений, показан пример использования, а также написан набор тестов с использованием Google C++ Testing Framework.

# Литература

Internet – ресурсы:

1. <https://ru.wikipedia.org/wiki/Стек>
2. <https://neerc.ifmo.ru/wiki/index.php?title=Стек>

Книги:

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ. М.: Издат. дом «Вильямс», 2000. С. 58–76