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 хранения данных: Стек на массиве»**

**Выполнил:** студент группы 381706-1

Остапович Денис Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533083472)

[2. Постановка задачи 4](#_Toc533083473)

[3. Руководство пользователя 5](#_Toc533083474)

[4. Руководство программиста 6](#_Toc533083475)

[4.1. Описание структуры программы 6](#_Toc533083476)

[4.2. Описание структур данных 6](#_Toc533083477)

[4.3. Описание алгоритмов 7](#_Toc533083478)

[5. Заключение 8](#_Toc533083479)

[6. Литература 9](#_Toc533083480)

# Введение

**Стек на массиве** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) stack — стопка; читается стэк) — [абстрактный тип данных](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D1%8B%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), представляющий собой [список элементов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), организованных по принципу [LIFO](https://ru.wikipedia.org/wiki/LIFO) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) last in — first out, «последним пришёл — первым вышел»).

Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю.

В [цифровом вычислительном комплексе](https://ru.wikipedia.org/wiki/%D0%A6%D0%92%D0%9A) стек называется магазином — по аналогии с [магазином в огнестрельном оружии](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD_(%D1%87%D0%B0%D1%81%D1%82%D1%8C_%D0%BE%D1%80%D1%83%D0%B6%D0%B8%D1%8F)) (стрельба начнётся с патрона, заряженного последним).

**Цель лабораторной работы** – реализация структуры хранения стека на массиве.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Разработать класс стека – TStack.
2. Создать класса обработки исключений – MyException, которые могут возникнуть при выполнении различных операций.
3. Продемонстрировать работу классa TStack.
4. Реализовать набор автоматических тестов на Google C++ Testing Framework.

# Руководство пользователя

Рассмотрим пример использования класса TStack.

После запуска программы пользователю предлагается ввести размеры стеков . Создаются два стека одинакового размера Stack и Stack2. Полностью заполняются натуральными числами от 0 до , и выводятся в консоль.

Затем из Stack извлекается элемент. Этот элемент выводится на экран и выводится Stack полученный после извлечения.

Поскольку теперь Stack и Stack2 несравнимы, то при корректной работе программы, на экран будет выведено сообщение что они различны. Программа завершена.

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Stack. Реализация в файле *main.cpp.*
* Модуль StackLib – статическая библиотека. Содержит файл stack.h, в котором реализован шаблонный класс *TStack.*
* Модуль StackTest. Содержит 18 тестов, описанных в файле *test.cpp* написанных наGoogle C++ Testing Framework.

## Описание структур данных

#### 4.2.1 Класс TStack

**protected:**

* *int size* –размер стека.
* *int top* –вершина стека.
* *T\* mas* – указатель на область памяти для хранение стека.

**public:**

1. Описаны 2 конструктора и деструктор:

* *TStack(int n = 0)* – конструктор по значению.
* *TStack(TStack<T> &S)* – конструктор копирования.
* *virtual ~TStack()* – деструктор.

1. Описаны методы

* *void Push(T A)* – добавить новый элемент A в стек.
* *Int GetSize()* – вернуть размер стека.
* *T GetTop()* – скопировать элемент из вершины стека.
* *T Pop()* – изъять с удалением элемент находящийся на вершине стека.
* *void Print()* – вывод элементов стека на экран.
* *bool IsFull()* – проверка стека на полноту.
* *bool IsEmpty()* – проверка стека на пустоту.
* *int operator!=(const TStack<T>& stack) const* – проверка стеков на различие
* *int operator==(const TStack<T>& stack) const* - проверка стеков на идентичность.
* *TStack& operator=(const TStack<T>& stack)* – присваивание стека.

## Описание алгоритмов

**Добавление элемента в стек.**

При добавлении элемента в стек, размещаем его в первую свободную ячейку массива, выделенного под хранение данных (если она имеется). На эту ячейку указывает *top* – вершина стека. Затем значение *top* увеличиваем на 1.

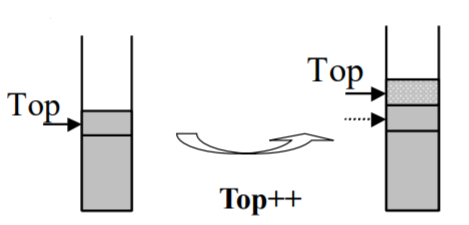


Рисунок 1. Добавление элемента в стек.

**Удаление элемента из стека.**

При удалении элемента из стека уменьшаем значение *top* на 1 – спускаемся к последнему существующему элементу и возвращаем его значение (если стек не пуст). Само значение не затирается, т.к. оно будет перезаписано при следующем добавлении другого элемента.

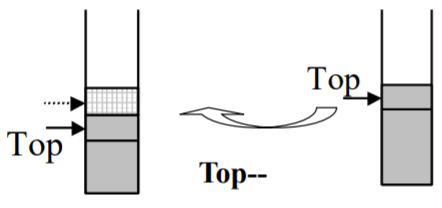


Рисунок 2. Удаление элемента из стека

# Заключение

В ходе выполнения лабораторной была разработана библиотека StackLib, реализующая шаблонный класс TStack. В ней реализованы методы работы со стеком описанные в разделе «Структуры данных».

Написаны тесты с использованием Google C++ Testing Framework.

# Литература

* Книги

1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
2. Bjarne Stroustrup The C++ programming language

* Ссылки в Internet

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [<http://www.itmm.unn.ru/files/2018/10/Primer-1.3.-Struktury-hraneniya-steka.pdf>], 2015.
2. Википедия [https://ru.wikipedia.org/wiki/Стек ]