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 стека»**

**Выполнил:** студент группы 381706-1

Полетуева Анастасия Николаевна

\_\_\_\_\_\_\_\_\_\_\_ 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** (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 стратегия «последним вошел — первым вышел» (last-in, first-out — LIFO). Примером стека в реальной жизни может являться стопка тарелок: когда мы хотим вытащить тарелку, мы должны снять все тарелки выше.

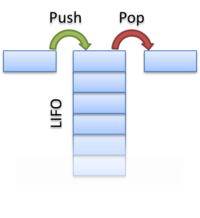


Рисунок 1. Стек.

# Постановка задачи

В рамках лабораторной работы ставится задача разработки стека (класс TStack). Для работы с этой структурой данных будут реализованы методы:

* добавления элемента в стек,
* извлечения элемента из стека (с удалением),
* проверка стека на полноту/пустоту.

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация класса стек – TStack.
2. Разработка программы, использующую класс TStack.
3. Реализация набора тестов, написанных с использованием GoogleTesting Framework.

# Руководство пользователя

При запуске программы c пользователя спрашивается размер стека для дальнейшей работы с ним. Затем создается Stack размером , заполненный натуральными числами от 0 до , и выводится на консоль.

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

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Stack. Содержит пример использования стека. Реализация в файле *main\_Stack.cpp.*
* Модуль StackLib – статическая библиотека. Содержит файл Stack.h, в котором описан интерфейс и реализация шаблонного класса *TStack.*
* Модуль StackTest. Содержит 9 тестов, описанных в файле *StackTest.cpp* и разработанных с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, содержащая класс исключений.

## Описание структур данных

#### 4.2.1 Класс TStack

Класс TStack является шаблонным классом. В нем находятся три поля со спецификатором доступа protected:

* int size – максимальный размер стека.
* int top – позиция вершины стека.
* T\* Elem – указатель на область памяти для хранение стека.

Далее в публичной зоне 2 конструктора, деструктор, 5 методов для работы со стеком (public):

* TStack(int n = 0) – конструктор с одним параметром.
* TStack(TStack<T> &m) – конструктор копирования.
* ~TStack() – деструктор.
* void Put(cons tint el) – добавить новый элемент в стек.
* T Get() – изъять с удалением элемент находящийся на вершине стека.
* bool IsFull() – проверка стека на полноту.
* bool IsEmpty() – проверка стека на пустоту.
* int GetSize() – количество элементов в стеке

## 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 параметр *top* – вершина стека. Затем значение *top* увеличиваем на 1.

**Удаление элемента из стека.**

При удалении элемента из стека уменьшаем значение *top* на 1 – спускаемся к последнему существующему элементу и возвращаем его значение. Само значение в ячейке с индексом не затирается, т.к. оно будет перезаписано при следующем добавлении другого элемента.



Рисунок 2. Добавление элемента в стек(1-5),

удаление элемента из стека(6-10).

# Заключение

В ходе выполнения лабораторной было установлено понятие стека на массиве. Была разработана программа, реализующая структуру хранения данных - стек. Реализованы методы работы со стеком. Программная реализация стека на массиве была продемонстрирована на примере. Разработаны тесты на основе Google C++ Testing Framework.

# Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования» (часть 1), 2015, -96с.
2. Википедия: свободная электронная энциклопедия: на русском языке: <https://ru.wikipedia.org/wiki/Стек>
3. Викиконспекты: <http://neerc.ifmo.ru/wiki/index.php?title=Стек>