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»**

**Выполнил:** студент группы 381706-1

Нечаева Екатерина Владимировна

\_\_\_\_\_\_\_\_\_\_\_ 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 – упорядоченный набор данных, в котором размещение новых элементов и удаление существующих производится только с одного его конца, который называют вершиной стека, т.е. стек – это список с одной точкой доступа к его элементам.

Стек – структура типа ***LIFO***(*Last In, First Out*) – последним вошел, первым выйдет.

Стек получил свое название из-за схожести с оружейным магазином с патронами (обойма): когда в стек добавляется новый элемент, то прежний проталкивается вниз и временно становится недоступным. Когда же верхний элемент удаляется из стека, следующий за ним поднимается вверх и становится опять доступным.

В 1946 Алан Тьюринг ввёл понятие стека. А в 1957 году немцы Клаус Самельсон и Фридрих Л. Бауэр запатентовали идею Тьюринга.

# Постановка задачи

В данной лабораторной работе ставится задача реализовать одну из структур данных - стек.

Размер стека — это фиксированная величина, и превышение лимита выделенной на стеке памяти приведёт к переполнению стека, поэтому элементы стека будем хранить в виде массива с динамической памятью. Размер задаётся при создании потока.

Для работы со стеком будут реализованы операции:

* добавления элемента в стек
* извлечения элемента из стека с последующим удалением
* проверка стека на полноту/пустоту

# Руководство пользователя

В начале запуска программы пользователю предлагается ввести размер стека. После этого есть два варианта: либо заполнить вручную, либо случайными числами. Выбрав один из вариантов, на консоль выводится полученный стек. Затем программой извлекается первый элемент и снова выводится уже новый стек.

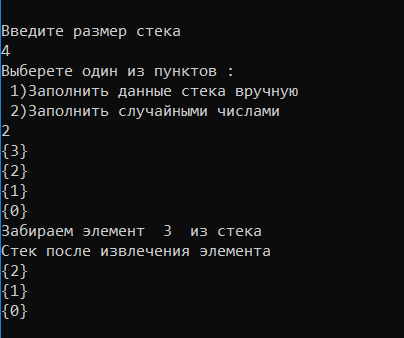


Рисунок 1. Пример использования стека и его методов.

# Руководство программиста

## 4.1 Описание структуры программы

Программа состоит из следующих модулей:

* Модуль stack. Содержит пример использования стека(*mainstack.cpp*)*.*
* Модуль libstack – статическая библиотека. Содержит заголовочный файл *stack.h*, в котором реализован шаблонный класс TStack.
* Модуль stackTest. Содержит 8 тестов, описанных в файле *stackTest.cpp*.
* Модуль libthrows – библиотека для исключений. Содержит заголовочный файл *throws.h*, в котором он и реализован.

## Описание структур данных

#### Класс TException – класс исключений.

Содержит одно защищенное поле *string str,* переменная строкового типа, которая хранит сообщение об исключении.

Содержит публичные методы

* TException(std::string \_str); //конструктор-инициализатор
* void Print(); //вывод данного исключения на консоль

#### Класс TStack – стек.

Содержит три защищенных поля

* int size; //размер стека
* int top; //вершина стека
* T\* mas; //массив элементов

Содержит публичные методы:

* TStack(int \_size = 5); //конструктор-инициализатор
* TStack(TStack<T> &A); //конструктор копирования
* ~TStack(); //деструктор
* void Put(const T &A); //положить элемент в стек
* virtual T Get(); //забрать элемент из стека с его последующим удалением
* int GetSize(); //получить размер стека
* void Print(); //вывод стека
* bool IsFull(); //проверка на полноту
* bool IsEmpty(); //проверка на пустоту

## Описание алгоритмов

Добавление элемента в стек.

При добавлении элемента в стек, размещаем его в первую свободную ячейку массива. На эту ячейку указывает параметр *top* – вершина стека, которая изначальна равна 0. При добавлении элемента, в массив с индексом *top* записывается данное значение. Затем значение *top* увеличиваем на 1.

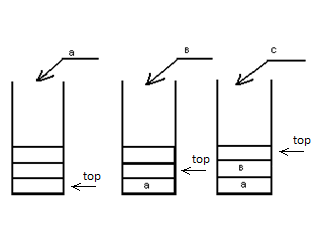


Рисунок 1. Добавление элемента в стек.

Забрать элемент с последующим удалением.

Забирается последний добавленный элемент в стеке, а значение *top* уменьшается на 1.

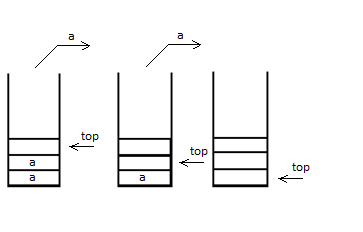


Рисунок 2 Удаление элемента из стека

# Заключение

В данной лабораторной работе была реализована библиотека TStack шаблонного типа, в которой были реализованы операции: добавление и удаление элемента, проверка стека на полноту или пустоту.

Программа прошла все созданные тесты на базе Google Test.

# Литература

1. Википедия: свободная электронная энциклопедия: на русском языке [Электронный ресурс]

https://ru.wikipedia.org/wiki/Стек

1. Структура данных стек [Электронный ресурс]

<https://studfiles.net/preview/1417256/page:50/>

1. Основные принципы программирования: стек и куча [Электронный ресурс]

<https://tproger.ru/translations/programming-concepts-stack-and-heap/>