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»**

**Выполнил:** студент группы 381706-1

Денисов Владислав Львович

\_\_\_\_\_\_\_\_\_\_\_ 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

Лабораторная работа направлена на практическое освоение динамической структуры данных Стек.

Стек (англ. stack — стопка) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. 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 стек называется магазином — по аналогии с магазином в огнестрельном оружии (стрельба начнётся с патрона, заряженного последним).

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

# Постановка задачи

В рамках лабораторной работы ставится задача эффективной реализации структуры данных – стек.

Хранить данную структуру в этой лабораторной работе будем с помощью массива. Такая организация стека удобна, если элемент информации занимает в памяти фиксированное количество длины, например, какую-то одну условную единицу. При этом отпадает необходимость хранения в элементе стека явного указателя на следующий элемент стека, что экономит память.

Для работы со стеком будут реализованы операции:

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

Программное решение будет выглядеть следующим образом:

1. Класс стека – TStack.
2. Класс для обработки исключений – TException, которые могут возникнуть при выполнении различных операций.
3. Программа, демонстрирующая работу классов TStack.
4. Набор автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

В качестве примера использования стека предлагается следующее. После запуска тестовой программы пользователь вводит максимальный размер стека. Создается стек заданного размера для хранения значений целого типа. Затем пользователю предлагается заполнить все ячейки стека. После этого происходит печать стека на консоль. Происходит удаление одного элемента. Снова выполняется печать того, что осталось в стеке. Создается точно такой же стек с помощью конструктора копирования.

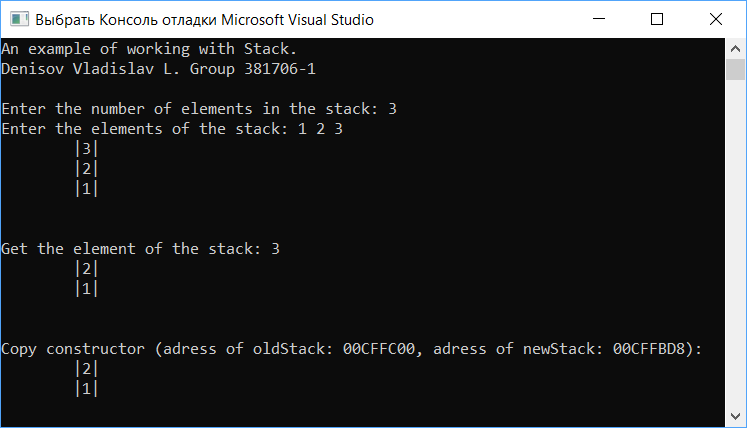


Рисунок 1 Пример работы демонстрационной программы

# Руководство программиста

## 4.1 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 8 тестов, описанных в файле *StackTest.cpp* и разработанных с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, позволяющая создавать собственные исключения.

## Описание структур данных

#### Класс TException – класс исключений.

Класс содержит 1 **private** поле *std::string msg* – переменная, хранящая сообщение об ошибке в виде строки.

И содержит 2 **public** элемента:

*TException(std::string \_str) : msg(\_msg)* – конструктор с одним параметром.

*void Print()* – метод отображения ошибки на консоль.

#### Класс TStack – вектор.

Рассмотрим класс *TStack* подробно.

template <class T> class TStack {…} – класс стека является шаблонным, что позволяет использовать его для хранения данных любого типа.

**Элементы класса, объявленные со спецификатором protected:**

int size – максимальный размер стека.

int top – позиция вершины стека.

T\* mas – указатель на область памяти, выделяемую под хранение стека.

**Элементы класса, объявленные со спецификатором public:**

TStack(int n = 0) – конструктор с одним параметром.

TStack(TStack<T> &S) – конструктор копирования.

virtual ~TStack() – деструктор.

void Put(T A) – метод, позволяющий добавить новый элемент в стек.

T Get() – метод изъятия элемента из вершины стека с удалением.

void Print() – метод отображения текущих элементов стека.

bool IsFull() – метод проверки стека на полноту.

bool IsEmpty() – метод проверки стека на пустоту.

## Описание алгоритмов

Рассмотрим некоторые алгоритмы, работа которых не очевидна на первый взгляд.

**Добавление элемента в стек.**

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

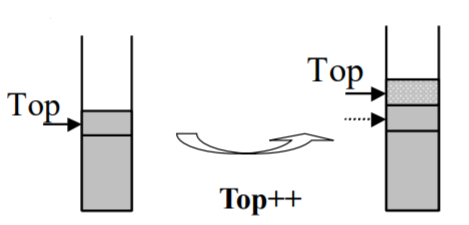


Рисунок 1 Добавление элемента в стек.

**Удаление элемента из стека.**

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

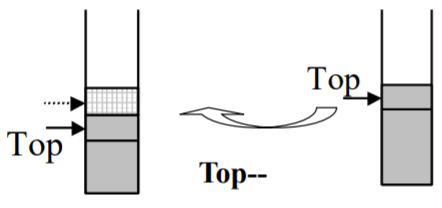


Рисунок 2 Удаление элемента из стека

# Заключение

В результате лабораторной работы была разработана библиотека, реализующая шаблонный класс стека. Она позволяет создать объект класса стека и выполнить с ним простейшие операции, задача реализации которых была поставлена в начале данной лабораторной работы.

Были разработаны и доведены до успешного выполнения тесты, разработанные для данного программного проекта с использованием Google C++ Testing Framework.

Программное решение было продемонстрировано с помощью простейшего набора операций над стеком. Описание примера работы со стеком было представлено в разделе «Руководство пользователя».

# Литература

1. Википедия: свободная электронная энциклопедия: на русском языке [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/Стек (дата обращения: 11.12.2018)
2. Методы программирования [Электронный ресурс] // URL: <http://www.itmm.unn.ru/files/2018/10/Primer-1.3.-Struktury-hraneniya-steka.pdf> (дата обращения: 11.12.2018)