МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**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

Колесова Кристина Юрьевна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018.

Оглавление

[1. Введение 3](#_Toc534392616)

[2. Постановка задачи 4](#_Toc534392617)

[3. Руководство пользователя 5](#_Toc534392618)

[4. Руководство программиста 6](#_Toc534392619)

[4.1. Описание структуры программы 6](#_Toc534392620)

[4.2. Описание структур данных 6](#_Toc534392621)

[4.3. Описание алгоритмов 7](#_Toc534392622)

[5. Заключение 8](#_Toc534392623)

[6. Список литературы 9](#_Toc534392624)

# Введение

Стек (англ. stack — стопка) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

Реализация структуры возможна на основе различных структур данных, например, с использованием статических или динамических массивов и линейных списков.

Цель данной лабораторной работы – реализация структуры данных стек на основе динамического одномерного массива.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Реализация класса стека - TStack
2. Реализация класса для обработки исключений – Texception
3. Реализация тестов на базе Google Test
4. Пример использования класса Tstack

В классе TSteck предполагается реализовать следующие методы:

* + 1. Put – добавление элемента
    2. Get – удаление элемента
    3. IsEmpty – проверка стека на пустоту
    4. IsFull – проверка стека на полноту

# Руководство пользователя

В качестве примера использования стека предлагается следующая тестовая программа.

1. Пользователю предлагают ввести размер стека, после чего создается стек заданного размера для хранения целочисленных значений;
2. Пользователю предлагают заполнить ячейки стека целыми значениями.;
3. Заданный стек выводится на консоль;
4. Удаляется один элемент стека;
5. Снова производится вывод стека на консоль.

На этом работа программы заканчивается.

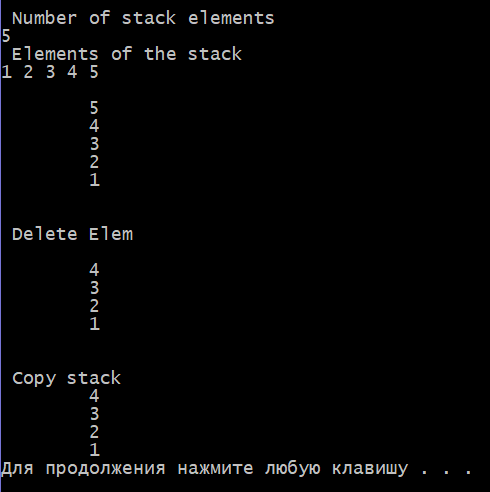


Рисунок . Пример использования стека

# Руководство программиста

## Описание структуры программы

В программе содержатся следующие модули:

* Модуль StackLib – статическая библиотека. Содержит файл Stack.h, в котором описан интерфейс и реализация шаблонного класса *TStack*.
* Модуль ExceptionLib – библиотека, позволяющая создавать собственные исключения.
* Модуль Stack. Содержит пример использования стека. Реализация в файле *main\_Stack.cpp.*
* Модуль StackTest. Содержит файл *StackTest.cpp,* в котором реализованы тесты для класса TSteck, разработанные с помощью использования фреймворка Google Test.

## Описание структур данных

* Класс TStack

Класс TStack является шаблонным.

1. 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.

1. 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.

* Класс TException

Класс содержит 1 private поле:

string msg – переменная, хранящая сообщение об ошибке в виде строки.

Содержит 2 public элемента:

msg(\_msg) – конструктор.

void Print() – метод отображения ошибки на консоль.

## Описание алгоритмов

* Добавление элемента в стек

При добавлении элемента в стек он будет размещаться в первую свободную ячейку стека. На эту ячейку указывает параметр top. Затем значение top увеличивается на 1.

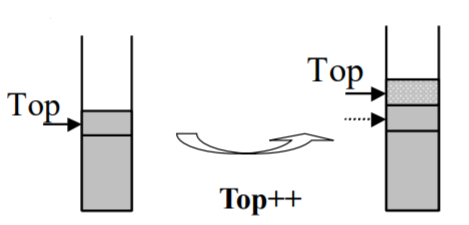


Рисунок 2. Добавление элемента в стек.

* Удаление элемента из стека

При удалении элемента из стека будет удаляться тот элемент, который последним был добавлен в стек. Для этого значение top уменьшается на 1. Затем возвращается элемент из ячейки, соответствующей новому параметру top. Само значение будет изменено при записи нового элемента.

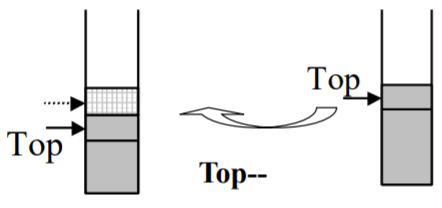


Рисунок 3. Удаление элемента из стека

# Заключение

В результате выполнения лабораторной работы была разработана библиотека, позволяющая работать с шаблонным классом стека. С помощью этой библиотеки можно создать объект класса стека и выполнить с ним операции добавления, извлечения, проверки на пустоту и полноту.

Были реализованы тесты для проверки работоспособности класса стека на базе GoogleTest. А также приведен пример работы стека.

# Список литературы

1. Википедия. Статья «Стек»: [https://ru.wikipedia.org/wiki/Стек].
2. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
3. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.