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**

**ЛАБОРАТОРНАЯ РАБОТА №5**

**«Структура хранения данных: мультистек»**

**Выполнил:** студент группы 381908-4

Яшин Егор Олегович

**Проверил:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

Нижний Новгород

2020.

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

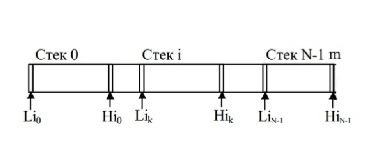
Мультистек — структура данных, представляющая собой упорядоченный набор N стеков, фиксированного размера. Каждый отдельный стек организован по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Стеки хранятся в памяти друг за другом единым блоком, размер которого m:

Рисунок 1. Структура памяти для хранения мультистека

*Где*  – индекс начала j-го стека, – индекс последнего элемент j-го стека.

Свойства такой структуры памяти:

1. – условие неподвижности первого стека;
2. – условие пустоты;
3. – условие неперекрытия;
4. – условие переполнения;

Цель данной лабораторной работы – реализация структуры данных мультистек на основе массива.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

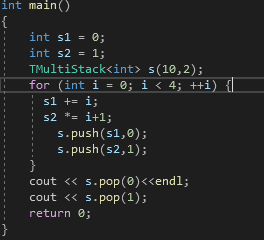
1. Разработка и реализация вспомогательного класса стека – TStack
2. Разработка и реализация класса мультистека - TNewStack
3. Реализация класса для обработки исключений – EexceptionLib
4. Реализация тестов на базе Google Test
5. Пример использования класса MStack

# Руководство пользователя

При запуске программы пользователю представляется простой пример демонстрации

программы. Создается объект класса MultiStack, мультистек. Затем в него кладутся

значения и на экран выводятся элементы из стеков.

 Руководство программиста

## Описание структуры программы

В программе содержатся следующие модули:

* Модуль MultiStackLib – содержит файл Stack.h, в котором описан интерфейс и реализация шаблонного класса *TStack* , файл MStack.h, в котором реализованы класс TMultiStack.
* Модуль MultiStack – содержит в себе файл реализации примера использования класса MultiStack.
* Модуль test\_multistacklib – содержит в себе файл test\_multistacklib.cpp, в котором находится набор тестов, для проверки работоспособности класса MultiStack.

## Описание структур данных

* Класс TStack

TStack(int size=1, bool data=true) - конструктор

TStack(const TStack<T> &stack) - конструктор копирования

~TStack() – деструктор

TStack<T> &operator=(const TStack<T> &stack) – оператор присваивания

void push(T item) - положить элемент

T pop() – взять элемент

bool is\_empty() – проверка на пустоту

bool is\_full() – проверка на заполненность

void set\_data(T\* data, int size, int front) – установить данные

T get\_max() - взять максимальный элемент

T get\_least() – взять наименьший элемент

int get\_item\_count() – получить количество элементов

int get\_size() – получить размер

template<class T1>

friend istream &operator>>(istream &istream, TStack<T1> &stack)

template<class T1>

friend ostream &operator<<(ostream &ostream, const TStack<T1> &stack) –

операторы ввода и вывода

static TStack<T> &load(const char \*name);

void save(const char \*name);

Сохранение и загрузка в файл

* Класс TMultiStack

int size - размер

T\* arr - массив элементов

int stack\_count - кол-во стеков

TStack<T> \*stacks; - cтеки

T\*\* old\_data;

TMultiStack(int size = 1, int stack\_count = 1); -конструктор

TMultiStack(const TMultiStack<T> &stack); - конструктор копирования

~TMultiStack(); -деструктор

TMultiStack<T> &operator=(const TMultiStack<T> &stack); - оператор

присваивания

void push(T item, int index);- положить элемент

T pop(int index); - взять элемент

bool is\_empty(int i); - проверка на пустоту

bool is\_full(int i); -проверка на полноту

int get\_stack\_with\_least\_elements();

template<class T1>

friend istream &operator>>(istream &istream, TMultiStack<T1> &stack);

template<class T1>

friend ostream &operator<<(ostream &ostream, const TMultiStack<T1> &stack);

static TMultiStack<T> &load(const char \*name);

void save(const char \*name);

операторы ввода, вывода, загрузки и сохранения

# Заключение

В результате выполнения лабораторной работы была разработана библиотека, позволяющая работать с шаблонным классом мультистека. С помощью этой библиотеки можно создать объект класса мультистека и некоторые операции операции над ним.

Были реализованы тесты для проверки работоспособности класса мультистека 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.