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**

**«Стек»**

**Выполнил:** студент группы 381706-1

Корнев Никита Алексеевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533799739)

[2. Постановка задачи 4](#_Toc533799740)

[3. Руководство пользователя 5](#_Toc533799741)

[4. Руководство программиста 6](#_Toc533799742)

[4.1. Описание структуры программы 6](#_Toc533799743)

[4.2. Описание структур данных 7](#_Toc533799744)

[4.3 Описание алгоритмов 8](#_Toc533799745)

[5. Заключение 9](#_Toc533799746)

[6. Литература 10](#_Toc533799747)

# Введение

Структура данных (англ. data structure) — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных в вычислительной технике. Для добавления, поиска, изменения и удаления данных структура данных предоставляет некоторый набор функций, составляющих её интерфейс.

Одной из основных структур данных является стек. Стек (англ. stack — стопка; читается стэк) — 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.

В некоторых языках (например, Lisp, Python) стеком можно назвать любой список, так как для них доступны операции pop и push. В языке C++ стандартная библиотека имеет класс с реализованной структурой и методами.

Нам же в данной лабораторной работе предстоит написать собственную реализацию данной структуры.

# Постановка задачи

1. Написать класс, реализующий стек.
2. Написать тесты на основе Google Tests для проверки работы класса.

# Руководство пользователя

Чтобы использовать данный класс в своем проекте, необходимо подключить библиотеку «Stack.h». Библиотека позволяет:

* Создавать объекты типа *стек*:

Stack st1; **конструктор по умолчанию**

Stack st2(10); **конструктор с параметром**

Stack st3(st2); **конструктор копирования**

* Складывать элементы в *стек*:

Stack st(5);

st.Put(1);

* Брать элементы из *стека*:

Stack st(5);

for (int i = 0; i < 5; i++)

st.Put(i);

st.Get();

* Получать размер *стека*:

st.GetSize();

* Проверять на *пустоту*:

st.IsEmpty();

* Проверятьна *полноту*:

st.IsFull();

# Руководство программиста

## Описание структуры программы

Проект состоит из следующих модулей:

1. Stack

Модуль cодержит файл код «main.cpp», в котором продемонстрирован пример использования структуры.

1. StackLib

Модуль содержит файл заголовок «Stack.h», описывающий структуру «стек, а также файл кода «main.cpp». Так как класс шаблонный, main содержит лишь подключение.

1. StackTest

Модуль содержит файлы кода «test\_main.cpp», «test\_stack.cpp». В последнем реализованы тесты для проверки корректности работы методов данного класса.

## Описание структур данных

#### *Структура TStack*

Структура **TStack** реализована в виде класса **TStack**. Поля со спецификатором доступа «protected»:

* + **int size;** *Размер стека*
  + **T\* mas;** *Элементы стека*
  + **int top;** *Верх*

Поля со спецификатором доступа «public»:

* **TStack();** *Конструктор по умолчанию*
* **TStack(int \_size);** *Конструктор с параметром*
* **TStack(TStack &st);** *Конструктор копирования*
* **~TStack();** *Деструктор*
* **void Put(T st);** *Положить в стек*
* **T Get();** *Взять из стека*
* **int GetSize();** *Получить размер*
* **bool IsFull();** *Проверка на полноту*
* **bool IsEmpty();** *Проверка на пустоту*

## Описание алгоритмов

Описание некоторых алгоритмов.

* TStack();

Конструктор по умолчанию, создает стек нулевого размера.

* TStack(int \_size);

Конструктор с параметром, создает стек размера \_*size*, выделяет массив под хранение элементов стека. Если *\_size* меньше нуля – выбрасывате исключение.

* TStack(TStack &st);

Конструктор копирования, создает стек от уже имеющегося стека.

* Void Put(T st);

Если стек полон – выбрасывает исключение. Присваивает верху стека значение st, увеличивает параметр *top*, характеризующий положение верха на единицу.

* T Get();

Если стек пуст – выбрасывате исключение. Уменьшает значение параметра *top* на единицу, возвращает значение верхнего элемента.

* int GetSize();

Возвращает значение параметра *size*.

* bool IsFull();

Если *top* больше либо равен *size*, возращает 1, в противном случае 0.

* Bool IsEmpty();

Если *top* равен равен 0, возвращает 1, в противном случае 0.

# Заключение

В данном лабораторной работе мне удалось:

* Успешно реализовать класс для стека
* Продемонстрировать пример использования данного класса
* Написать тесты на основе Google Tests для проверки корректной работы данного класса

# Литература

Интернет-источники:

1. <https://ru.wikipedia.org/wiki/Структура_данных>
2. <https://ru.wikipedia.org/wiki/Стек>
3. <https://habr.com/post/341586/>

Книги:

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.