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

Суслов Егор Игоревич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

**Содержание**

[**1.** **Введение** 3](#_Toc510437279)

[**2.** **Постановка задачи** 4](#_Toc510437280)

[**3.** **Руководство пользователя** 5](#_Toc510437281)

[**4.** **Руководство программиста** 6](#_Toc510437282)

[**4.1.** **Описание структуры программы** 6](#_Toc510437283)

[**4.2.** **Описание структур данных** 6](#_Toc510437284)

[**5.** **Заключение** 8](#_Toc510437285)

[**6.** **Литература** 9](#_Toc510437286)

# **Введение**

Стек ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *stack* — стопка; читается *стэк*) — [абстрактный тип данных](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D1%8B%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), представляющий собой [список элементов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), организованных по принципу [*LIFO*](https://ru.wikipedia.org/wiki/LIFO) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *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](https://ru.wikipedia.org/wiki/%D0%A6%D0%92%D0%9A) стек называется магазином — по аналогии с [магазином в огнестрельном оружии](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD_(%D1%87%D0%B0%D1%81%D1%82%D1%8C_%D0%BE%D1%80%D1%83%D0%B6%D0%B8%D1%8F)) (стрельба начнётся с патрона, заряженного последним).

В 1946 [Алан Тьюринг](https://ru.wikipedia.org/wiki/%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3,_%D0%90%D0%BB%D0%B0%D0%BD) ввёл понятие стека. А в 1957 году немцы Клаус Самельсон и Фридрих Л. Бауэр запатентовали идею Тьюринга.

Стек является общей структурой данных для представления данных, которые должны обрабатываться в определенном порядке. Например, когда функция вызывает другую функцию, которая, в свою очередь, вызывает третью функцию, важно, чтобы третья функция вернулась на вторую функцию, а не первую.

Один из способов реализации такого порядка обработки данных — это организовать своего рода очередь вызовов функций. Последняя добавленная в стек функция, будет завершена первой и наоборот, первая добавленная в стек функция будет завершена последней. Таким образом, сама структура данных обеспечивает надлежащий порядок вызовов.

# **Постановка задачи**

Реализовать класс TStack для работы с данными различных типов в виде стека. Для реализации этого использовать шаблоны. Поля класса должны быть закрыты. Типы полей:

Int - позиция вершины стека

T \* - указатель на стек

const int - максимальное количество элементов в стеке

Обязательно должны присутствовать методы:

* поместить элемент в вершину стека;
* удалить элемент из вершины стека и вернуть его;
* вывод стека на экран;
* n-й элемент от вершины стека;
* получить размер стека;
* получить указатель на стек;
* получить номер текущего элемента в стеке.

Должны быть реализованы конструкторы: по умолчанию, копирования и инициализатор.

# **Руководство пользователя**

1. После запуска программы пользователю предлагается ввести 5 чисел
2. Затем введенные числа выводятся на экран
3. Выход

# **Руководство программиста**

## **Описание структуры программы**

Программа состоит из следующих модулей:

* Модуль Stack. Содержит пример использования стека. Реализация в файле «Stack\_main.cpp*».*
* Модуль StackLib – статическая библиотека. Содержит файл Stack.h, в котором описан интерфейс и реализация шаблонного класса TStack(2 конструктора, деструктор и 8 методов).
* Модуль StackTest. Содержит 18 тестов, описанных в файле «StackTest.cpp*»* и разработанных с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, содержащая класс исключений.
  1. **Описание структур данных**

Класс TStack

Содержит поля с идентификатором protected :

* T \*stackPtr;  - указатель на стек
* const int size;  - максимальное количество элементов в стеке
* int top;    - номер текущего элемента стека

Содержит public методы:

1. Stack(int = 0) - по умолчанию размер стека равен 0
2. Stack(const Stack<T> &); - конструктор копирования
3. ~Stack();  - деструктор
4. void Put(T A); - поместить элемент в вершину стека
5. T Get(); - удалить элемент из вершины стека и вернуть его
6. bool IsFull(); - проверка на полноту
7. bool IsEmpty();- проверка на пустоту
8. void PrintStack(); - вывод стека
9. int operator!=(const TStack<T>& stack) const; - проверка стеков на различие
10. int operator==(const TStack<T>& stack) const; - проверка стеков на идентичность
11. TStack& operator=(const TStack<T>& stack); - присваивание стека

# 

# **Заключение**

В ходе работы мне удалось изучить:

1. Специфику типа хранения данных стек
2. Методы стека
3. Применение стека
4. Написать и проверить библиотеку с использованием Google C++ Testing Framework

Написанная мною библиотека удачно выполняет следующие действия с данными любого типа:

1. Поместить элемент в вершину стека;
2. Удалить элемент из вершины стека и вернуть его;
3. Вывод стека на экран;
4. n-й элемент от вершины стека;
5. Получить размер стека;
6. Получить указатель на стек;
7. Получить номер текущего элемента в стеке.

# **Литература**

* Книги:

1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
2. Топп У., Форд У. Структуры данных в С++. - М. Бином, 1999.
3. Мейн М., Савитч У. Структуры данных и другие объекты в С++. - М.: Издательский дом “Вильямс”, 2003.
4. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.

* Ссылки в Internet:

1. Учебно-методическое пособие из электронной библиотеки ННГУ: «ЛАБОРАТОРНЫЙ ПРАКТИКУМ» : [http://www.unn.ru/books/met\_files/Pract\_ADS.pdf].