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 на списках»**

**Выполнил:** студент группы 381706-1

Остапович Денис Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc536523907)

[2. Постановка задачи 4](#_Toc536523908)

[3. Руководство пользователя 5](#_Toc536523909)

[4. Руководство программиста 6](#_Toc536523910)

[4.1. Описание структуры программы 6](#_Toc536523911)

[4.2. Описание структур данных 6](#_Toc536523912)

[4.3. Описание алгоритмов 7](#_Toc536523913)

[5. Заключение 8](#_Toc536523914)

[6. Литература 9](#_Toc536523915)

# Введение

**Стек на списках** — структура данных, представляющая собой упорядоченный набор элементов одного типа, фиксированного количества, связанных между собой последовательно посредством указателей, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Каждый элемент стека на списках имеет указатель на следующий элемент. Последний элемент стека на списках указывает на [NULL](https://ru.wikipedia.org/wiki/NULL_(%D0%A1%D0%B8)). Элемент, на который нет указателя, является первым (головным) элементом списка. В стек на списках все новые элементы добавляются в начало.  Вершиной стека на списках является первый элемент в списке.

**Цель лабораторной работы** – разработка структуры хранения: стек на списках.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация класса стека на списках – TStackList.
2. Разработка программы, демонстрирующей работу классa TStackList.
3. Реализация набора автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

После запуска программы пользователю предлагается ввести n – размер стека. Далее стек заполняется числами от 0 до (n - 1). После чего выводится на экран. Далее пока стек не пуст, все элементы извлекаются и выводятся на экран. Программа завершается.

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль StackOnLists. Содержит пример использования стека. Реализация в файле *main.cpp.*
* Модуль StackOnListsLib– статическая библиотека. Содержит файл stackonlists.h, в котором описан интерфейс и реализация шаблонного класса *TStackList.*
* Модуль StackOnListsTest. Содержит 14 тестов, описанных в файле *StackOnListsTest.cpp* и разработанных с помощью использования Google C++ Testing Framework.

## Описание структур данных

#### Класс TStackList

Класс *TStackList* является шаблонным классом. Наследуется от класса *TList* со спецификатором public. В классе *TStackList* определено одно поле, со спецификатором доступа protected, int size – максимальный размер стека.

**Далее со спецификатором доступа public**:

1. Прописаны 2 конструктора и деструктор:

* *TStackList(int \_size = 10)* – конструктор с одним параметром.
* *TStackList(TStackList<T> &A)* – конструктор копирования.
* *~TStackList()* – деструктор.

1. Прописаны методы для работы со стеком на списках:

* *void Push(T A)* – добавить новый элемент A в начало списка.
* *T Pop()* – изъять с удалением элемент находящийся в начале списка.
* *void Print()* – вывод элементов стека на экран.
* *bool IsFull()* – проверка стека на списках на полноту.
* *bool IsEmpty()* – проверка стека на списках на пустоту.
* *int GetSize()* – получить текущее количество элементов в стеке на списках.
* *int GetMaxSize()* – получить размер стека.

## Описание алгоритмов

**Добавление и изъятие элемента в стеке на списках.**

При добавлении элемента в стек, он размещается на вершине. При изъятии элемента из стека берется элемент, находящийся на вершине стека. В данной лабораторной работе целесообразно вершиной стека считать первый элемент списка. То есть добавление и изъятие элементов происходит только из начала списка. В этом случае сложность этих операций составит О(1). Иначе, сложность операции составила бы O(n).

# Заключение

В ходе выполнения лабораторной был произведен анализ задачи - установлено понятие стека на списках. Была разработана библиотека StackOnListsLib, реализующая шаблонный класс стека TStackList. Разработаны тесты для данного программного проекта с использованием Google C++ Testing Framework.

# Литература

* Книги
  + - 1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.

1. Bjarne Stroustrup The C++ programming language

* Ссылки в Internet
  + - 1. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [http://www.itmm.unn.ru/files/2018/11/Primer-1.6.-Struktura-hraneniya-neskolkih-stekov-s-ispolzovaniem-spiskov.pdf], 2015.
      2. Википедия: <https://ru.wikipedia.org/wiki/Стек>
      3. Википедия: <https://ru.wikipedia.org/wiki/Связный> список