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**

**«Разработка структуры хранения данных: стек на списках»**

**Выполнил:** студент группы 381706-1

Нечаева Екатерина Владимировна

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

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc560926)

[2. Постановка задачи 4](#_Toc560927)

[3. Руководство пользователя 5](#_Toc560928)

[4. Руководство программиста 6](#_Toc560929)

[4.1 Описание структуры программы 6](#_Toc560930)

[4.2 Описание структур данных 6](#_Toc560931)

[4.3 Описание алгоритмов 6](#_Toc560932)

[5. Заключение 7](#_Toc560933)

[6. Литература 8](#_Toc560934)

# Введение

Стек – упорядоченный набор данных, в котором размещение новых элементов и удаление существующих производится только с одного его конца, который называют вершиной стека, т.е. стек – это список с одной точкой доступа к его элементам.

Стек – структура типа ***LIFO***(*Last In, First Out*) – последним вошел, первым выйдет.

Подходы к реализации стека

1. На основе связных списков (linked lists) - длина стека ограничена объемом доступной памяти
2. На основе статических массивов - длина стека фиксирована (задана его максимальная длина – количество элементов в массиве)

Реализация стека на основе односвязного списка заключается в том, что каждый новый элемент создаётся заново. Для этого нужно определить узел, который хранит данные и указатель на следующий элемент списка.

# Постановка задачи

В рамках лабораторной работы ставится задача реализации структуры данных, представляющей собой стек на основе списка.

Для этого требуется реализовать два класса: TList, отвечающий за структуру данных список, и класс TStackList- стек на основе списка.

Для этого были реализованы методы:

• добавления элемента в начало списка, в нашем случае стека с помощью методов списка.

• извлечения элемента из списка-стека с последующим удалением

• проверка стека на полноту/пустоту

• получение текущего размера стека

# Руководство пользователя

В начале запуска программы пользователю предлагается ввести размер стека. После этого есть два варианта: либо заполнить вручную, либо случайными числами. Выбрав один из вариантов, на консоль выводится полученный стек. Затем программой извлекается первый элемент и снова выводится уже новый стек.

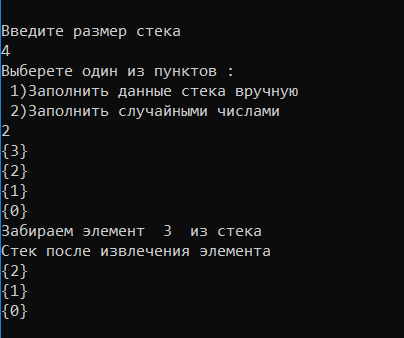


Рисунок 1. Пример использования стека и его методов.

# Руководство программиста

## 4.1 Описание структуры программы

Программа состоит из следующих модулей:

* Модуль stacklist. Содержит пример использования программы с помощью класса TStackList (mainstacklist.cpp)
* Модуль libstacklist – статическая библиотека. Содержит заголовочный файл stacklist.h, в котором реализован шаблонный класс TStackList.
* Модуль stacklistTect. Содержит тесты, проверяющие правильность работы методов (stacklistTest.cpp)

## Описание структур данных

#### Класс TStackList.

Так как данный класс наследуется со спецификатором public, полей у нго есть, но содержит методы:

* TStackList(); //конструктор по умолчанию
* TStackList(TStackList <T> &A); //конструктор копирования
* void PrintTSL(); //печать стека
* int GetSize(); //получить размер стека
* void Put(const T &a); //положить в стек
* T Get(); //забрать из стека
* bool IsEmpty(); //проверка на пустоту
* bool IsFull(); //проверка на полноту

## Описание алгоритмов

**Добавление и изъятие элемента из стека с помощью метода класса TList.**

Так как класс TStackList наследуется от класса TList, то чтобы реализовать методы вставки и изъятия элемента из стека, мы используем метод добавления в начало и изъятие из начала списка, который был реализован ранее.

В данном случае вершиной стека является начало списка.

# Заключение

В ходе выполнения данной лабораторной была реализована структура хранения данных с помощью стека на списках. Рассмотрели отличия в реализации стека на массиве и на списках. Таким образом, можно сказать, что стек, реализованный на списках кажется более оптимальным, так как размер в данном случае ограничен объемом памяти, а не пользователем, на основе массива.

Были разработаны тесты, проверяющие корректную работу методов данных классов, с помощью Google C++ Testing Framework.

# Литература

1. Применение матрицы [Электронный ресурс]

<http://topuch.ru/interesnie-primeneniya-matrici-v-matematike-fizike-ekonomike-b/index.html>

1. Матрицы и их виды [Электронный ресурс]

<https://math1.ru/education/matrix/matrix.html>

1. Учебно-методическое пособие [Электронный ресурс]

<http://www.unn.ru/books/met_files/Pract_ADS.pdf>

1. Умножение матриц [Электронный ресурс]

<http://ru.solverbook.com/spravochnik/matricy/umnozhenie-matric/>