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**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Стек на списке»**

**Выполнил:** студент группы 381706-2

Паузин Леонид Павлович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018

Содержание

[1.Введение 3](#_Toc8279413)

[2. Постановка задачи 4](#_Toc8279414)

[3.Руководство пользователя 5](#_Toc8279415)

[4.Руководство программиста 6](#_Toc8279416)

[4.1 Описание структуры программы 6](#_Toc8279417)

[4.2 Описание структур данных 6](#_Toc8279418)

[4.3 Описание алгоритмов 6](#_Toc8279419)

[5.Заключение 7](#_Toc8279420)

[6.Литература 8](#_Toc8279421)

# 1.Введение

Стек ‒ абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in ‒ first out, «последним пришёл ‒ первым вышел»). Это значит, что мы будем иметь доступ только к последнему добавленному элементу.

Структура данных, представляющая собой конечное множество упорядоченных элементов (узлов), связанных друг с другом посредством указателей, называется связным списком. Каждый элемент связного списка содержит поле с данными, а также указатель (ссылку) на следующий и/или предыдущий элемент. Эта структура позволяет эффективно выполнять операции добавления и удаления элементов для любой позиции в последовательности

# 2. Постановка задачи

Выполнение лабораторной работы предполагает решение следующих задач:

* Разработка и реализация класса стека на списках
* Разработка программы, демонстрирующей работу классa TListStack.
* Реализация набор автоматических тестов с использованием Google C++ Testing Framework.

# 3.Руководство пользователя

При запуске у пользователя спрашивается размер стека. Затем создается стек и заполняется случайными числами от 1 до 100. После этого производится изъятие элементов. Когда стек полностью освободится, появится сообщение, что стек пуст

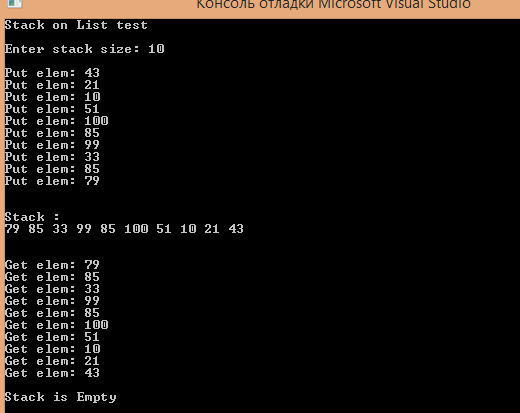


Рисунок 1. Демонстрация работы класса стек на списках

# 4.Руководство программиста

## 4.1 Описание структуры программы

liststacklib (TListStack.h) ‒ модуль, реализующий класс стек на списке

liststack(main.cpp) ‒ модуль, реализующий тестирование структуры стек на списке

liststacktest ‒ модуль тестирования при помощи Google C++ Testing Framework

## 4.2 Описание структур данных

Поля:

TDatLink<ValType>\* pFirst ‒ указатель на первое звено списка

Методы:

int IsEmpty() const ‒ проверка переполнения

void Put(ValType Val) ‒ добавить элемент в стек

ValType Get() ‒ извлечь элемент

## 4.3 Описание алгоритмов

При добавлении элемента в стек, он размещается на вершине стека. При изъятии элемента из стека берется элемент, находящийся на вершине стека. Поскольку в данной работе стеки хранится на списках, то целесообразно вершиной стека считать первый элемент списка. То есть добавление и изъятие элементов происходит только из начала списка.

# 5.Заключение

В ходе выполнения лабораторной было установлено понятие стека на списке. Была разработана библиотека, реализующая шаблонный класс TListStack. Она позволяет при работе со списком выполнять базовые операции извлечение и добавление элементов в стек. Реализованы тесты для проверки работы класса и программы для тестирования.

# 6.Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Статья «Связный список» на сайте: <https://tproger.ru/translations/linked-list-for-beginners/>
3. Статья «Стек» на сайте: <https://prog-cpp.ru/data-stack/>