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 класса стек на списке»»**

**Выполнил:** студент группы 381706-2

Банденков Даниил Викторович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2019.

# Содержание

[Содержание 2](#_Toc534333789)

[1 Введение 3](#_Toc534333790)

[2 Постановка задачи 4](#_Toc534333791)

[3 Руководство пользователя 5](#_Toc534333792)

[4 Руководство программиста 6](#_Toc534333793)

[4.1 Описание структуры программы 6](#_Toc534333794)

[4.2 Описание структур данных 6](#_Toc534333795)

[4.3 Описание алгоритмов 6](#_Toc534333796)

[5 Заключение 7](#_Toc534333797)

[6 Литература 8](#_Toc534333798)

# Введение

Линейные списки являются чрезвычайно гибкой структурой, так как их легко сделать большими или меньшими, и их элементы доступны для вставки или удаления в любой позиции списка. Списки также можно объединять или разбивать на меньшие списки.

Списки регулярно используются в приложениях, например, в программах информационного поиска, трансляторах программных языков или при моделировании различных процессов.

В математике список определяется как последовательность элементов определенного типа: *X1, X2, … Xn ,*где *n>*=0. Количество элементов *n* называется длиной списка, *X1 –*первый элемент списка*, Xn*– последний элемент списка. В случае *n*=0, список пустой. Важное свойство списка заключается в том, что его элементы можно линейно упорядочить в соответствии с их позицией в списке, т.е., *Xi*предшествует*Xi+1*и следует за *Xi-1.*Элемент*Xi*имеет позицию *i*.

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»). Принцип работы стека сравнивают со стопкой листов бумаги: чтобы взять второй сверху, нужно снять верхний.

Структура хранения стека – линейный список (начало списка – вершина стека).

# Постановка задачи

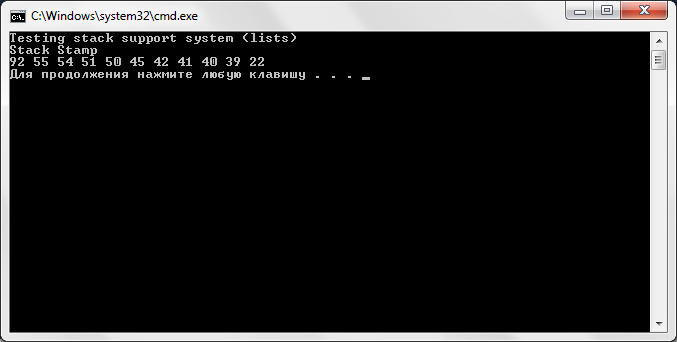
Цель данной лабораторной работы — разработать на языке программирования С++ статическую библиотеку, реализующую структуру данных для хранения стека на списке.

Разработка класса список:

1. Реализация структуры звена
2. Реализация структура памяти
3. Организация списка свободных звеньев
4. Реализация структура хранения стека

# Руководство пользователя

Данная программа тестирует динамическую структуру стек на списке. На каждой итерации цикла на 100 элементов элемент либо загружается в стек либо выгружается из него, результат выводятся на экран. (см Рис 1).

Рис 1.Пример использования программы.

# Руководство программиста

## Описание структуры программы

*В решении содержатся следующие модули:*

1. liststacklib (TListStack.h, TListStack.cpp)– модуль реализующий класс стек на списке.
2. liststacktest(liststack\_test.cpp) - модуль тестирования класса стек на списке при помощи Google C++ Testing Framework.
3. liststack(main.cpp) – реализация программы для тестирования динамической структуры стек на списке.

## Описание структур данных

**Структура:**

*Поля:*

pFirst - первое звено.

*Методы:*

ValType Get()-извлечь значение из стека.

Put(ValType Val)- добавить значение в стек.

IsEmpty() const - контроль переполнения СД.

## Описание алгоритмов

*Добавление в стек элемента a (псевдокод)*

Создать новое звено pTemp;

Положить в звено pTemp значение a;

Положить следующим звеном для pTemp pFirst;

pFirst = pTemp;

# Заключение

В ходе работы реализован класс стек на списке, в нем реализованы отношения следования при помощи сцепления (адресных указателей), функции вставки и удаления элементов.

Реализованы тесты для проверки работы класса и программа для тестирования.

# Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. <http://mvblog.ru/archives/427/> (Дата обращения 04.01. 2019)
3. <http://www.codenet.ru/progr/cpp/dlist.php> (Дата обращения 04.01. 2019)