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 список»»**

**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 7](#_Toc534333796)

[5 Заключение 8](#_Toc534333797)

[6 Литература 9](#_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*.

# Постановка задачи

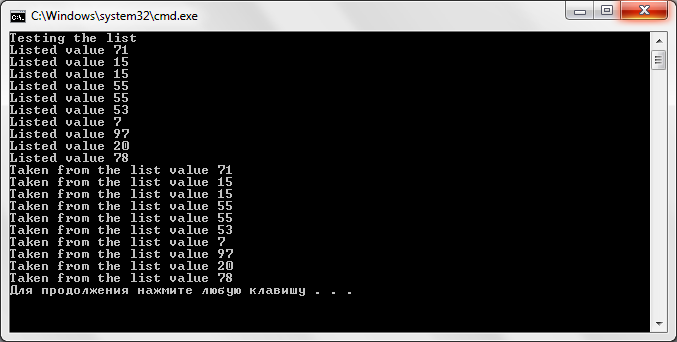
Цель данной лабораторной работы — разработать на языке программирования С++ статическую библиотеку, реализующую структуру данных для хранения списка.

Разработка класса список:

1. Реализация отношения следования при помощи сцепления (адресных указателей)
2. Линейный список как структура хранения
3. Реализация списков с использованием динамически-распределяемой области памяти
4. Разработка общего представления линейного списка для обеспечения списковой структуры хранения

# Руководство пользователя

Данная программа тестирует динамическую структуру список. В список поочередно загружаются 10 элементов, а затем они выгружаются из списка и выводятся на экран. (см Рис 1).

Рис 1.Пример использования программы.

# Руководство программиста

## Описание структуры программы

*В решении содержатся следующие модули:*

1. datlistlib (TDatList.h, TDatList.cpp)– модуль реализующий класс список.
2. datlink (TdatLink.h, TdatLink.cpp)– реализация класса звено списка.
3. datlisttest(datlist\_test.cpp) - модуль тестирования класса список при помощи Google C++ Testing Framework.
4. datlist(main.cpp) – реализация программы для тестирования динамической структуры список.

## Описание структур данных

**Структура:**

*Поля:*

pFirst - первое звено.

pLast - последнее звено.

listLen - количество звеньев в списке.

*Методы:*

InsFirst(ValType val)- установить первым элементом списка.

InsLast(ValType val) - установить последним элементом списка.

GetFirst() - получить значение первого элемента списка.

GetDatValue(int num) - получить значение n-ного элемента списка.

GetLast() - получить значение последнего элемента списка.

DelFirst()-удалить первый элемент списка.

DelLast()-удалить последний элемент списка.

Del(int num)- удалить n-ный элемент списка.

IsFull()-полнота списка.

InsTo(int num, const ValType& Val)- вставить на позицию.

IsEmpty()-пустота списка.

## Описание алгоритмов

*Вставка в начало значения val (псевдокод)*

Создать новое звено tmp со знвченим Val и указателем на pFirst;

pFirst = tmp;

Если ListLen = 0

pLast = tmp;

Инкрементировать ListLen;

# Заключение

В ходе работы реализован класс список, 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. <https://prog-cpp.ru/data-ols/> (Дата обращения 03.01. 2019)
3. <http://www.codenet.ru/progr/cpp/dlist.php> (Дата обращения 03.01. 2019)