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 на массиве»»**

**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*.

Для представления списков можно использовать массивы вместо ссылочных структур. Они представляют интерес для определенного круга сценариев, где изначально создается список, после чего он остается в относительно стабильном положении с небольшим числом вставок и удалений. Тогда список на массивах дает преимущества, особенно если часто выполняются операции по произвольному доступу к элементам по индексу.

# Постановка задачи

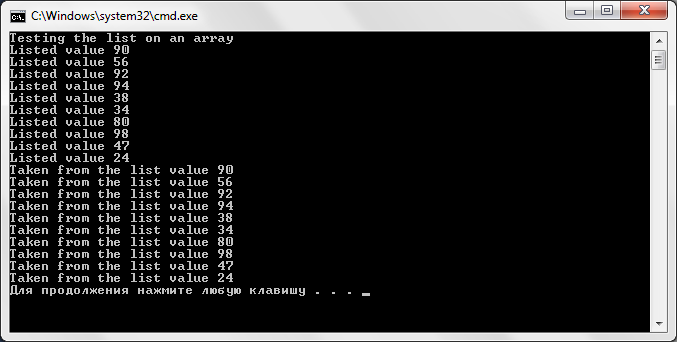
Цель данной лабораторной работы — разработать на языке программирования С++ статическую библиотеку, реализующую структуру данных для хранения списка на массиве.

Разработка класса список на массиве:

1. Реализация отношения следования при помощи массива индексов
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. arrlistlib (TarrList.h, TArrList.cpp)– модуль реализующий список на массивах.
2. arrlist(main.cpp) – модуль реализующий тестирование динамической структуры список.
3. arrlisttest(arrlist\_test.cpp) - модуль тестирования класса список на массивах при помощи Google C++ Testing Framework.

## Описание структур данных

**Структура:**

*Поля:*

mem - память под элементы списка.

index - индексы следующих элементов.

size - максимальный размер списка.

count - число элементов в списке.

firstInd - индекс начала списка.

freeDataInd - индекс первого элемента в массиве.

*Методы:*

NextFreeInd()-найти свободное место в массиве.

TArrList(int n = 10) - конструктор с параметром.

TArrList(TArrList& al) - конструктор копирования.

~TarrList() - деструктор.

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.

GetListLength()-вернуть максимальный размер списка.

IsEmpty()-пустота списка.

## Описание алгоритмов

*Вставка в начало значения val (псевдокод)*

Если список полон

Ошибка;

mem[freeDataInd] = val;

index[freeDataInd] = firstInd;

firstInd = freeDataInd;

freeDataInd = вызвать функцию NextFreeInd();

увеличить count на 1;

# Заключение

В ходе работы реализован класс список на массиве, в нем реализованы отношения следования при помощи массива индексов, 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://studopedia.ru/14_1558_spiski-realizatsiya-spiskov-na-baze-massiva.html>

(Дата обращения 03.01. 2019)

1. <https://rsdn.org/article/alg/list.xml> (Дата обращения 03.01. 2019)