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**

**«Реализация списка на основе массивов»**

**Выполнил:** студент группы 381706-1

Соболева Ю.А.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533027637)

[2. Постановка задачи 4](#_Toc533027638)

[3. Руководство пользователя 5](#_Toc533027639)

[4. Руководство программиста 6](#_Toc533027640)

4.[1. Описание структуры программы 6](#_Toc533027641)

4.[2. Описание структур данных 6](#_Toc533027642)

4.[3. Описание алгоритмов 7](#_Toc533027643)

[5. Заключение 8](#_Toc533027644)

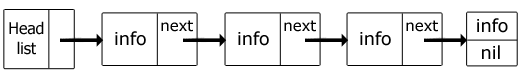
[6. Литература 9](#_Toc533027645)

# **Введение**

**Структура данных** - программная единица, которая определяет метод хранения и обработки различных логически связанных данных в вычислительной технике. Знание структур данных позволяет наиболее компактно и практично расположить данные в памяти компьютера. В данной работе мы рассмотрим такую структуру данных, как **список**.

**Список –** абстрактный тип данных, реализующий упорядоченный набор значений. Списки отличаются от массивов тем, что доступ к их элементам осуществляется последовательно, в то время как массивы – структура данных произвольного доступа. Данный абстрактный тип имеет несколько реализаций в виде структур данных.

**Связный список –** это структура данных, представляющая собой конечное множество упорядоченных элементов, связанных друг с другом посредствам указателей. Каждый элемент структуры содержит поле с какой-либо информацией, а также указатель на следующий элемент. В данной лабораторной работе указатели на следующий элемент мы будем хранить в отдельном массиве.



**Целью данной лабораторной работы** является разработка структуры данных для хранения связных списков с использованием массивов, а также освоение таких инструментов разработки программного обеспечения, как система контроля версий Git и фрэймворк для разработки автоматических тестов Google Test.

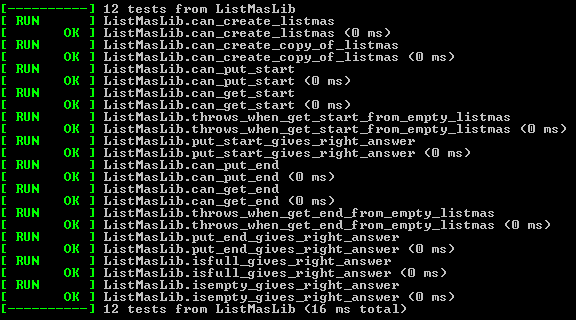
# **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Реализация класса списка TListMas.
2. Разработка интерфейса для данного класса.
3. Обеспечение работоспособности примера использования.
4. Реализация нескольких тестов на базе Google Test.

# **Руководство пользователя**

При запуске программы на экран выводится сообщение, что все тесты пройдены успешно. Пример вывода сообщений для класса TListMas показан на рис.1.



# **Руководство программиста**

## **Описание структуры программы**

**Проект “ListMas” состоит из следующих файлов:**

1. mainListMas.cpp (в нём находится main)

**Проект “ ListMasLib ” состоит из следующих файлов:**

1. ListMasLib.h (описание класса “TListMas”, реализация методов класса “TListMas” и перегрузка операций )
2. ListMasLib.cpp

**Проект “ Test” состоит из следующих файлов:**

1. test\_listmaslib.cpp (реализация тестов для класса TListMas)

## **Описание структур данных**

***Класс TListMas:***

*Поля:*

* T\* mas; - массив на котором составляется список;
* int\* index; - массив указателей на следующий элементж
* int start, size, count; - индекс начала списка, размер массива и количество заполненных элементов в списке;

*Конструкторы и деструктор:*

* TListMas<T> (int n = 0); - конструктор инициализатор;
* TListMas<T> (TListMas &A); - конструктор копирования;
* *~*TListMas<T> (); - деструктор;

*Методы:*

* void PutStart (T B); - добавляет элемент в начало списка (если он не полон);
* void PutEnd (T B); - добавляет элемент в конец списка (если он не полон);
* T GetStart(); - забирает начальный элемент из списка (если он не пуст);
* T GetEnd(); - забирает последний элемент из списка (если он не пуст);
* bool IsFull(); - проверяет полон ли список;
* bool IsEmpty(); - проверяет пуст ли список

*Перегруженные операторы:*

Нет перегруженных операторов

## **Описание алгоритмов**

**Метод PutStart:**

Если список полон, бросаем исключение. Ищем элемент массива, не принадлежащий списку. Присваиваем ему указанное значение. Указатель на начало списка переопределяем на только что добавленный элемент. Переопределяем его указатель на прошлое начало.

**Метод GetStart:**

Если список пуст, то бросаем исключение. Начало списка устанавливаем на следующий за первым элемент. Удаляем бывший первый элемент.

**Метод PutEnd:**

Если список не полон, то создаем указатель на элемент массива, в него записываем значение начала списка. В цикле ищем последний элемент, следуя по указателям*.* Как только конец списка будет найден, берем пустой элемент массива и инициализируем его. Устанавливаем для текущего последнего элемента указатель на следующий – только что созданный.

**Метод GetEnd:**

Если список пуст, то бросаем исключение. Иначе создаем указатель на первый элемент списка. В цикле ищем предпоследний элемент, следуя по указателям*.* Создаем новый элемент. В него записываем указатель на последнее звено списка. Получаем данные из этого звена. Удаляем указатель на последнее звенои тем самым освобождаем память, занимаемую последним элементом. Для предпоследнего элемента*,* устанавливаем в качестве следующего за ним -1, т.к. он теперь стал последним.

# **Заключение**

В ходе выполнения лабораторной работы я смогла реализовать такую структуру данных, как список на двух массивах. Вместе с ней, согласно заданному интерфейсу, я реализовала класс TListMas. Написание нескольких своих тестов, помогло мне разобраться с системой автоматических тестов Google Test.

В результате проделанной работы у меня получилось

1. Реализовать класс списка TListMas.

2. Разработать интерфейс для данных классов.

3. Обеспечить работоспособность примера использования.

4. Реализовать некоторые тесты на базе Google Test

Таким образом, данная лабораторная работа отвечает всем поставленным задачам. Это означает, что цель работы была достигнута.

# **Литература**

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 105с.

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

1. http://kvodo.ru/data-structures-introduction.html
2. Национальный открытый университет «Интуит». Курс «Основы программирования». Лекция 11: Структуры данных: общее понятие, реализация. Простейшие структуры данных: очередь, стек. Использование стека и обратная польская запись.

URL: https://www.intuit.ru/studies/courses/2193/67/lecture/1980?page=3