МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

**«Структура данных – односвязный линейный список с использованием массивов»**

**Выполнил:** студент группы 381706-1

Голованова Елена Александровна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018.

Оглавление

[1. Введение 3](#_Toc1890508)

[2. Постановка задачи 4](#_Toc1890509)

[3. Руководство пользователя 5](#_Toc1890510)

[4. Руководство программиста 6](#_Toc1890511)

[a. Описание структуры программы 6](#_Toc1890512)

[b. Описание структур данных 6](#_Toc1890513)

[c. Описание алгоритмов 7](#_Toc1890514)

[5. Заключение 9](#_Toc1890515)

[6. Список литературы 10](#_Toc1890516)

# Введение

Нехватка памяти —одна из главных проблем программирования. Очень много программ требуется достаточно большого объема вместимости, и поэтому появляются многие разные структуры хранения данных.

**Структура данных** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *data structure*) — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных [данных](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)) в [вычислительной технике](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80).

В этой лабораторной работе мы рассмотрим такую структуру, как односвязный линейный список с использованием массивов.

**Односвязный список на массивах** – это структура данных, представляющая собой упорядоченный набор значений, в котором некоторое значение может встречаться более одного раза.

Для создания такого списка используется два массива одинакового размера: содержательный массив и массив индексов. Эти массивы связанны друг с другом по индексу. По фиксированному *i*, содержательный массив, в *i*-й ячейке, хранит значение элемента списка, а массив индексов, в *i*-й ячейке, содержит индекс следующего элемента списка в содержательном массиве.

Если список заполнен не полностью, то для пустых ячеек в содержательном массиве, в соответствующих ячейках в массиве индексов лежит значение «-2». Для последнего элемента списка, в соответствующей ячейке массива индексов лежит значение -1.

Как правило, стоит дополнительно хранить индекс первого элемента списка в содержательном массиве.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Реализация класса TArrayList.
2. Выполнение определенных функций, операторов и конструкторов в ранее упомянутом классе.
3. Создать класс TException для избежания ошибок.
4. Выполнение примера использования программы.
5. Реализация простых тестов на базе Google Test.

# Руководство пользователя

Работа пользователя происходит так:

1. В начале программы у пользователя спрашивают размер списка.
2. Далее пользователь должен ввести количество элементов, которые будут лежать в конце списка.
3. Затем спрашивается количество элементов, которое необходимо положить в конец списка.
4. Полученный список выводится на экран. Он будет выглядеть следующим образом: .
5. Следующим действием будет изъятия элементов с конца и начала списка, после чего они появятся на экране.
6. Последним пунктом программы является вывод списка после изъятия элементов.

# 4. Руководство программиста

## Описание структуры программы

Для реализации лабораторной работы создается несколько модулей:

* Модуль ArrayList.

Этот модуль содержит реализацию примера использования программы.

* Модуль ArrayListLib

Этот модуль содержит реализацию. Содержит файл ArrayList.h, в котором описан интерфейс и реализация шаблонного класса ArrayList.

* Модуль ArrayListTest

Этот модуль содержит тесты, проверяющие работу каждого методов класса ArrayList. Они разработаны с помощью использования Google C++ Testing Framework.

* Модуль Exception

Этот модуль содержит класс исключений.

## Описание структур данных

Класс TElem – класс узлов списка. Он является шаблонным. Некоторые фукции наследуются от класса Очередь.

1. В зоне protected определены восемь полей:

* T \* mas - Массив элементов списка;
* int \*nextInd - Массив индексов, указывающих на следующий элемент списка;
* int \*predInd - Массив индексов, указывающих на предыдущий элемент списка;
* int size - Размер списка;
* int num - Количество элементов в списке;
* int start - Индекс первого элемента списка;
* int end - Индекс последнего элемента списка;
* Queue <int> freeElem - Очередь пустых ячеек в массив;

2. В зоне public:

* TArrayList(int \_size) - Конструктор с параметром;
* TArrayList(TArrayList<T> &A) - Конструктор копирования;
* ~TArrayList() - Деструктор;
* void Put(int n, T elem) - Добавить промежуточный элемент;
* T Get(int n) - Извлечь промежуточный элемент;
* void PutStart(T elem) - Положить в начало списка;
* void PutEnd(T elem) - Положить в конец списка;
* T GetStart() - Забрать из начала списка с удалением;
* T GetEnd() - Забрать из конца списка с удалением;
* bool IsFull() - Проверка на полноту;
* bool IsEmpty() - Проверка на пустоту;
* void Print() - Вывод списка.

## Описание алгоритмов

* Добавление звена списка в начало и в конец.

Сначала проверяем не заполнен ли список. Если он заполнен, то бросаем исключение. Если нет, то в очереди свободных позиций freeElem, берем первую свободную ячейку i. По полученному индексу в массив mas записываем значение, которое хотим положить в список. Определяем, что следующим для этого элемента, будет элемент с текущим индексом start, то есть nextInd[i] = start. Если, перед добавлением, список не был пуст, то предыдущим для первого элемента списка делаем только что добавленный элемент, то есть: predInd[start] = i. Если же список был пуст, то определяем, что добавленный элемент является и последним элементом в списке. Затем индекс start переопределяем на только что добавленный элемент: start = i. Увеличиваем количество элементов в списке num++.

Для добавления элемента в конец списка, рассуждения аналогичны, с поправкой на то, что добавляем в конец списка.

* Удаление звена списка из начала и из конца.

Сначала проверяем не пуст ли список. Если он пуст, то бросаем исключение. Если нет, то в переменную elem записываем элемент в начале списка mas[start]. Затем переопределяем индекс первого элемента списка, он будет равен индексу элемента следующим за первым: newstart = nextInd[start]. Затем в очередь свободных позиций freeElem, добавляем освободившуюся после изъятия первого элемента свободную ячейку start. Идентифицируем пустыми соответствующие ячейки массивов nextInd и predInd: nextInd[start] = predInd[start] = -2. Если после извлечения первого элемента, список не пуст, то предыдущего элемента для нового первого элемента не существует: predInd[newstart] = -1. Затем индекс start переопределяем на newstart: start = newstart. Уменьшаем количество элементов в списке num--.

Для извлечения элемента из конца списка, рассуждения аналогичны.

# 5. Заключение

В результате выполнения данной лабораторной работы была разработана структура хранения данных – односвязные списки с использованием массивов.

Получены новые знания и навыки.

Были реализованы тесты для проверки работоспособности вышеперечисленных классов на базе GoogleTest.

# 6. Список литературы

1. Википедия. Статья «Структура данных»

[https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85]

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
3. Википедия: свободная электронная энциклопедия: на русском языке: [https://ru.wikipedia.org/wiki/Список\_(информатика)]