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**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Структура хранения данных: Список»**

**Выполнил:** студент группы 381706-2

Паузин Леонид Павлович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018

Содержание

[1.Введение 3](#_Toc8259886)

[2. Постановка задачи 4](#_Toc8259887)

[3.Руководство пользователя 5](#_Toc8259888)

[4.Руководство программиста 6](#_Toc8259889)

[4.1 Описание структуры программы 6](#_Toc8259890)

[4.2 Описание структур данных 6](#_Toc8259891)

[4.3 Описание алгоритмов 7](#_Toc8259892)

[5.Заключение 8](#_Toc8259893)

[6.Литература 9](#_Toc8259894)

# 1.Введение

Структура данных, представляющая собой конечное множество упорядоченных элементов (узлов), связанных друг с другом посредством указателей, называется связным списком. Каждый элемент связного списка содержит поле с данными, а также указатель (ссылку) на следующий и/или предыдущий элемент. Эта структура позволяет эффективно выполнять операции добавления и удаления элементов для любой позиции в последовательности

Причем это не потребует реорганизации структуры, которая бы потребовалась в массиве. Минусом связного списка, как и других структур типа «список», в сравнении его с массивом, является отсутствие возможности работать с данными в режиме произвольного доступа, т. е. список – структура последовательно доступа, в то время как массив – произвольного.

Каждый узел односвязного списка содержит указатель на следующий узел. Из одной точки можно попасть лишь в следующую точку, двигаясь тем самым в конец. Так получается своеобразный поток, текущий в одном направлении



Рисунок 1. Односвязный список

На изображении каждый из блоков представляет элемент (узел) списка. Здесь и далее Head list – заголовочный элемент списка (для него предполагается поле next). Он не содержит данные, а только ссылку на следующий элемент. На наличие данных указывает поле info, а ссылки – поле next. Признаком отсутствия указателя является поле NULL.

# 2. Постановка задачи

Реализация класса структуры данных ‒ список и выполнение следующих операций:

* добавления в начало узла списка
* добавления в конец узла списка
* добавление и извлечение узла в промежуточные узлы списка
* проверка полноты
* проверка пустоты
* извлечения с удалением узла из начала списка
* извлечения с удалением узла из конца списка,

# 3.Руководство пользователя

В список поочерёдно загружаются 12 элементов, а затем они извлекаются из списка.



Рисунок 2. Демонстрация работы класса список

# 4.Руководство программиста

## 4.1 Описание структуры программы

Модульная структура программы:

* datlistlib (TDatList.h, TDatList.cpp) ‒ модуль, реализующий класс список
* datlist(main.cpp) ‒ модуль, реализующий тестирование списка
* datlisttest ‒ модуль тестирования при помощи Google C++ Testing Framework
* datlink(TDatLink.h, TDatLink.cpp) ‒ модуль, реализующий звено списка

## 4.2 Описание структур данных

Поля:

int listLen – количество звеньев в списке

TDatLink<ValType>\* pFirst – первое звено в списке

TDatLink<ValType>\* pLast – последнее звено в списке

Методы:

int IsFull() ‒ проверка переполнения

int IsEmpty() ‒ проверка пустоты

void InsFirst(const ValType &Val) – установить первым элементом

void InsLast(const ValType &Val) – установить последним элементом

void InsTo(int num, const ValType& Val) – вставить на выбранную позицию

void Del(int num) – удалить элемент с заданным номером

void DelFirst() – удалить первый элемент списка

void DelList() – удалить последний элемент списка

ValType& GetDatValue(int num) const получить значение элемента с заданным номером

int GetListLength() – получить количество звеньев

## 4.3 Описание алгоритмов

**Добавление звена списка в начало**

При добавлении звена в начало списка мы создаем указатель на объект класса TDatLink. Затем выделяем память под объект этого класса и с помощью конструктора с параметрами для TDatLink, передав туда значение, которое необходимо положить в начало списка и указатель на текущее начало, создаем очередное звено списка. Указатель на начало списка переопределяем на добавленный элемент

# 5.Заключение

В ходе выполнения лабораторной было установлено понятие списка на указателях. Была разработана библиотека, реализующая шаблонный класс списка. Она позволяет при работе со списком выполнять базовые операции извлечение и добавление элементов списка. Реализованы тесты для проверки работы класса и программы для тестирования.

# 6.Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Статья «Алгоритмы и структуры данных для начинающих: связный список» <https://tproger.ru/translations/linked-list-for-beginners/>
3. Статья «Связный список» <http://kvodo.ru/linked-lists.html>
4. Википедия: свободная электронная энциклопедия ‒ связный список: <https://ru.wikipedia.org/wiki/Связный_список>