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**

оставь надежды всяк сие читающий (карин т.а.)

**Отчет по учебной практике**

**«Название лабораторной»**

**Выполнил:**

студент группы 381706­1

Карин Тимофей Андреевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Проверил:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc536219868)

[3. Руководство пользователя 5](#_Toc536219869)

[4. Руководство программиста 6](#_Toc536219870)

[4.1. Описание структуры программы 6](#_Toc536219871)

[4.2 Описание структур данных 6](#_Toc536219872)

[4.3 Описание алгоритмов 6](#_Toc536219873)

[5. Заключение 7](#_Toc536219874)

[6. Список литературы 8](#_Toc536219875)

1. Введение

Список является структурой данных, в которой каждый элемент (кроме последнего) хранит указатель на следующий элемент. В основном, при этом вспоминается класс или структура, состоящая из данных и указателей (на следующий элемент, на предыдущий, на первый, на последний, на любой другой). Такое представление списка наиболее распространённое, но, как оказывается, не единственное.

В данной лабораторной работе будет представлена реализация списка немного непривычным методом. Основываться этот список будет на двух (в случае с односвязным списком) или на трёх (в случае с двусвязным списком) массивах.

Как это будет реализовано? Все массивы имеют одинаковый размер, равный количеству элементов в списке (Да, в данном случае размер списка ограничен). Один из массивов хранит непосредственно данные пользователя. Легко понять, что не хватает указателей. Соответственно второй массив хранит указатели на следующие элементы, а если есть и третий массив, то в нём хранятся указатели на предыдущие элементы. При этом конкретный элемент имеет один и тот же номер (индекс) во всех массивах.

Как заполняются массивы? Массив данных заполняет пользователь. Рассмотрим, как заполняется массив следующих элементов. Если значение указателя какого-либо элемента равно -2, то это означает, что элемент свободен и в списке никак не задействован. Если элемент имеет значение -1, то это означает, что за этим элементом ничего не следует, то есть, что этот элемент последний в списке. Если же значение равно какому-то значению в диапазоне от 0 до n-1, где n – количество элементов в списке, то это означает, что следующий элемент списка находится в массивах под этим номером.

2. Постановка задачи

Целью данной лабораторной работы является создание библиотеки для работы с двусвязными спискам на трёх массивах. Для этого необходимо:

* Разработать класс для хранения списков на массивах;
* Написать пример использования библиотеки;
* Проверить работоспособность библиотеки;

3. Руководство пользователя

Для пользователей была специально написана небольшая программа, демонстрирующая основные функции библиотеки.

В самом начале работы пользователю необходимо ввести размер создаваемого списка. Затем перед ним появится меню основных действий:

1. Добавить элемент в начало списка;
2. Добавить элемент в конец списка;
3. Получить первый элемент (с удалением из списка);
4. Получить последний элемент (с удалением из списка);
5. Проверка на пустоту;
6. Проверка на полноту;
7. Выход;

Для выбора действия необходимо ввести номер соответствующей команды. Если будет введён некорректный номер, пользователю будет предложено ввести номер снова.

Если пользователь попробует создать список отрицательного размера, или добавить в полный список ещё один элемент, или получить элемент из пустого списка, то работа программы прекратится, а пользователь на экране увидит сообщение об ошибке.

4. Руководство программиста

4.1. Описание структуры программы

Программа состоит из трёх основных модулей:

* *MasListLib* – библиотека для работы с двусвязными списками на массивах;
* *MasList* – пример использования библиотеки;
* *MasListTest* – набор тестов, проверяющих работоспособность библиотеки.

Также в программе используется класс исключений *TExeption* и фреймворк Google Test.

4.2 Описание структур данных

Рассмотрим шаблонный класс *TMasList*:

Поля со спецификатором доступа private:

* *T\* mas –* массив пользовательских данных;
* *int \*nextInd –* массив указателей на следующий элемент;
* *int \*prevInd –* массив указателей на предыдущий элемент;
* *int size –* размер списка;
* *int count –* количество элементов в списке;
* *int start –* номер (индекс) первого элемента;
* *int finish –* номер (индекс) последнего элемента;
* *TQueue<int> freeElem –* очередь пустых элементов.

Методы со спецификатором доступа public:

* *TMasList(int \_size = 10) –* конструктор;
* *TMasList(TMasList<T> &A) –* конструктор копирования;
* *~TMasList() –* деструктор;
* *void PushStart(T elem) –* добавить элемент в начало;
* *void PushFinish(T elem) –* добавить элемент в конец;
* *T PullStart() –* получить первый элемент (с удалением);
* *T PullFinish() –* получить последний элемент (с удалением);
* *bool IsFull() –* проверка на полноту;
* *bool IsEmpty() -* проверка на пустоту.

4.3 Описание алгоритмов

Добавление элемента в начало:

Прежде всего нужно убедиться, что список не полон. Затем, берём из очереди номер пустой ячейки. Заполняем соответствующие этому номеру ячейки массивов. Пользовательские данные кладём соответственно в массив данных. В массив указателей на следующий элемент кладём указатель на первый элемент. Если номер первого элемента равняется -1 (то есть изначально список был пуст), то элемент, который мы добавляем является и первым, и последним, присваиваем соответствующие значения. Если же список не был пуст, то у того элемента, который был первым появится предшествующий ему. Соответственно у него нужно сменить индекс предыдущего элемента, а начальным сделать элемент, который мы добавили. Также нужно увеличить значение переменной, хранящей количество элементов, на единицу.

Добавление элемента в конец:

Точно так же, как и в предыдущем случае, проверяем на полноту и берём пустую ячейку, добавляем пользовательские данные. Если изначально список был пуст, то теперь наш элемент является и первым, и последним, и нужно указателю на начальный элемент присвоить значение нашего элемента. Если же список не был пуст, то последний элемент становится предпоследним, и его указателю на следующий элемент нужно присвоить номер добавляемого элемента. Теперь осталось присвоить указателю на конец списка номер добавленного элемента и увеличить значение переменной, хранящей количество элементов списка, на единицу.

Удаление из начала списка:

Проверяем на пустоту. Берём указатель на начало. Сохраняем данные из соответствующей ячейки и указатель на следующий элемент. Добавляем элемент в очередь пустых. Обоим указателям присваиваем значение -2 (то есть освободили ячейку). Если элементов не осталось, то, значит, наш элемент был также и последним, и указателям на конец и на начало списка присваиваем значение -1. Если элементы ещё остались, то нужно перейти к следующему элементу, сделать его начальным, а индекс на предыдущий элемент изменить на -1. Также уменьшаем количество элементов на 1.

Удаление из конца списка:

Проверяем на пустоту, берём последний элемент, сохраняем его данные и указатель на предыдущий элемент. Добавляем элемент в очередь пустых, указателям присваиваем значение -2. Элемент, предшествующий удаляемому, становится последним. Если он существует, то его указатель на следующий элемент меняем на -1. Если его не существует, то мы удалили также и первый элемент, поэтому меняем указатель на начало на -1.

5. Заключение

В ходе проделанной работы были получены следующие результаты:

* Создана библиотека для работы с двусвязными списками на трёх массивах;
* Написана программа, демонстрирующая работу библиотеки;
* Обработано большинство исключительных ситуаций;
* Все методы библиотеки проверены с помощью фреймворка Google Test.

6. Список литературы

1. Ахо Альфред В, Хопкрофт Джон Э и Ульман Джеффри Д Структуры данных и алгоритмы [Книга]. - [б.м.] : Вильямс, 2003.
2. Лафоре Роберт Структуры данных и алгоритмы в Java [Книга]. - СПб : Питер, 2013. - 2 : стр. 704.
3. Павловская Т. А. C/C++ Программирование на языке высокого уровня [Книга]. - СПб : Питер, 2003.
4. Страуструп Бьерн Язык программирования C++ Бином, 2004.