

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

РТУ МИРЭА

Отчет по выполнению практического задания № 6

**Тема:**

«Двунаправленные динамические списки»

Дисциплина: «Структуры и алгоритмы обработки данных»

Выполнил студент: Боргачев Т.М.

Группа: ИНБО-10-23

Москва – 2024

# СОДЕРЖАНИЕ

[1 ФОРМУЛИРОВКА ЗАДАЧИ 2](#_Toc162310243)

[1.1 Задание 2](#_Toc162310244)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ 4](#_Toc162310245)

[2.1 Задание – преобразование списка 4](#_Toc162310246)

[2.1.1 Структура узла 4](#_Toc162310247)

[2.1.2 Базовые операции 4](#_Toc162310248)

[2.1.3 Структура списка 7](#_Toc162310249)

[2.1.4 Описание дополнительных методов для списка 7](#_Toc162310250)

[2.1.5 Ожидаемые тесты программы 10](#_Toc162310251)

[2.2 Реализация алгоритма на языке C++ 11](#_Toc162310252)

[2.3 Тестирование программы 15](#_Toc162310253)

[3 ВЫВОДЫ 18](#_Toc162310254)

[4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ 19](#_Toc162310255)

Цель: получение знаний и практических навыков управления двунаправленным списком в программах на языке С++.

# 1 ФОРМУЛИРОВКА ЗАДАЧИ

## Задание

Разработать многомодульную программу, которая демонстрирует выполнение всех операций, определенных вариантом 5, над линейным двунаправленным динамическим списком.

Требования к разработке:

* 1. Разработать структуру узла списка, структура информационной части узла определена вариантом. Для определения структуры узла списка, используйте тип struct или class. Сохраните определение структуры узла и прототипы функций в заголовочном файле.
  2. Разработайте функции для выполнения операции над линейным двунаправленным динамическим списком:
* создание списка;
* вставку узла;
* удаление узла;
* вывод списка в двух направлениях (слева направо и справа налево);
* поиск узла с заданным значением (операция должна возвращать указатель на узел с заданным значением).
  1. Дополнительные операции над списком, указанные вариантом, оформите в виде функций и включите в отдельный файл с расширением cpp. Подключите к этому файлу заголовочный файл с определением структуры узла.
  2. Разработайте программу, управляемую текстовым меню, и включите в меню демонстрацию выполнения всех операций задания и варианта.
  3. Проведите тестирование операций.
* Оцените сложность алгоритма первой дополнительной операции.

Вариант 5:

Тип информационной части:

* Номер счета в банке (20-значное число), дата, вид операции (приход или расход), сумма вклада.

Дополнительные операции:

* Вставка нового узла перед первым узлом.
* Удаление сведений по счету (всех узлов), у которого общая сумма вклада равна нулю (сумма по приходу, минус сумма по расходу).
* Создать новый список из исходного, которого будет содержать остаток по всем видам операций одного счета, указав вид операции – приход, и текущую дату.

# 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

## 2.1 Задание – преобразование списка

### 2.1.1 Структура узла

В соответствии с вариантом, каждый узел представляет собой банковскую ячейку с номером счета, датой операции, типом операции, полученными или изъятыми средствами. Тогда структура узла примет вид в соответствии с рис. 1.

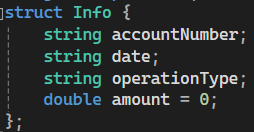


Рисунок 1 – Структура узла

### 2.1.2 Базовые операции

Блок-схемы базовых операций для списка: проверка на пустоту, добавление элемента, поиск элемента, вывод слева направо, удаление элемента, вывод справа налево – соответственно представлены на рис. 2, 3, 4, 5, 6 и 7.

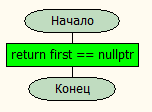


Рисунок 2 – Метод проверки на пустоту

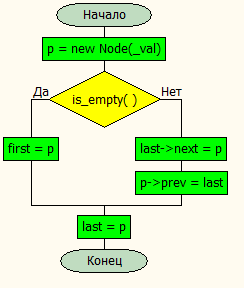


Рисунок 3 – Метод добавления элемента в конец списка

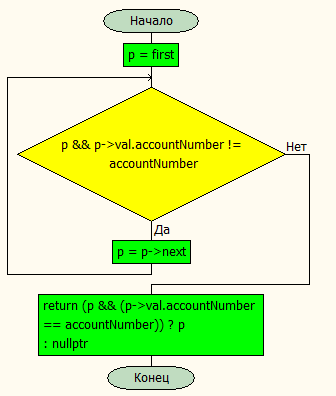


Рисунок 4 – Метод поиска объекта в списке

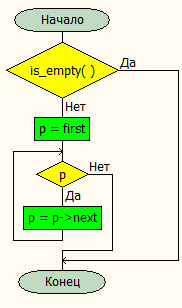


Рисунок 5 – Метод вывода списка слева направо

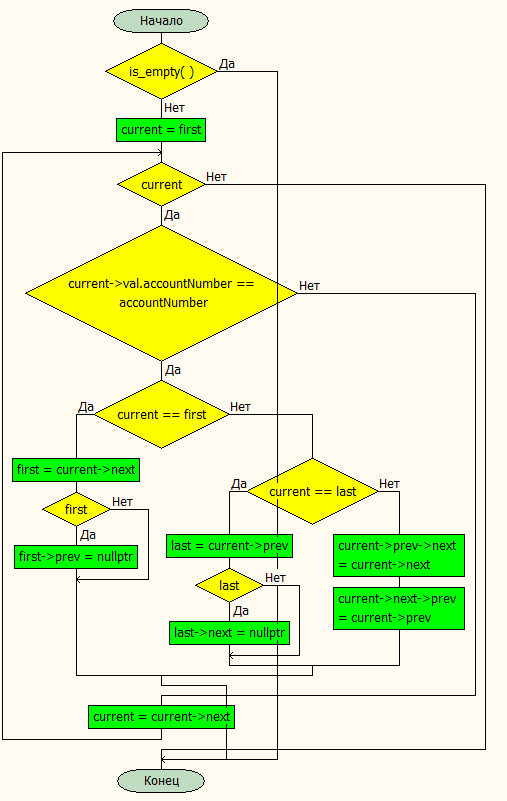


Рисунок 6 – Метод удаления объекта из списка

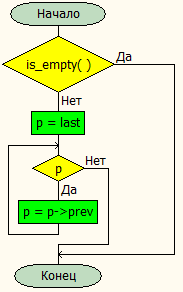


Рисунок 7 – Метод вывода списка справа налево

### 2.1.3 Структура списка

Двунаправленный список состоит из значения узла, указателя на следующий объект и указателя на предыдущий объект. Тогда структура списка будет выглядеть как представлено на рис. 8.

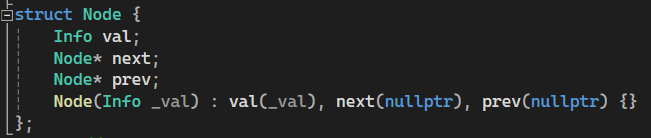


Рисунок 8 – Структура списка

### 2.1.4 Описание дополнительных методов для списка

Блок-схемы дополнительных методов для списка представлены на рис. 9, 10 и 11 соответственно.

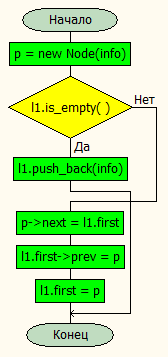


Рисунок 9 – Метод вставки объекта в начало списка

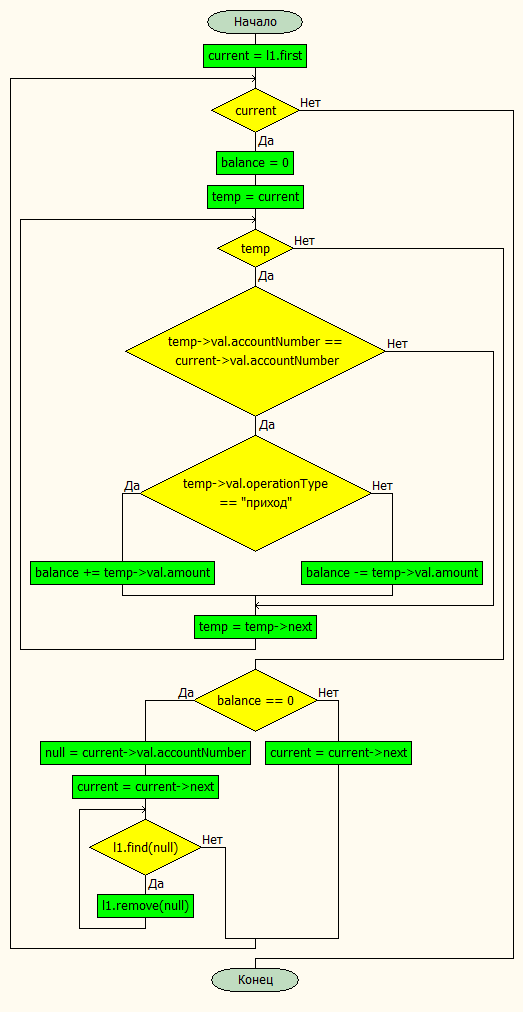


Рисунок 10 – Метод удаления счетов с нулевым балансом

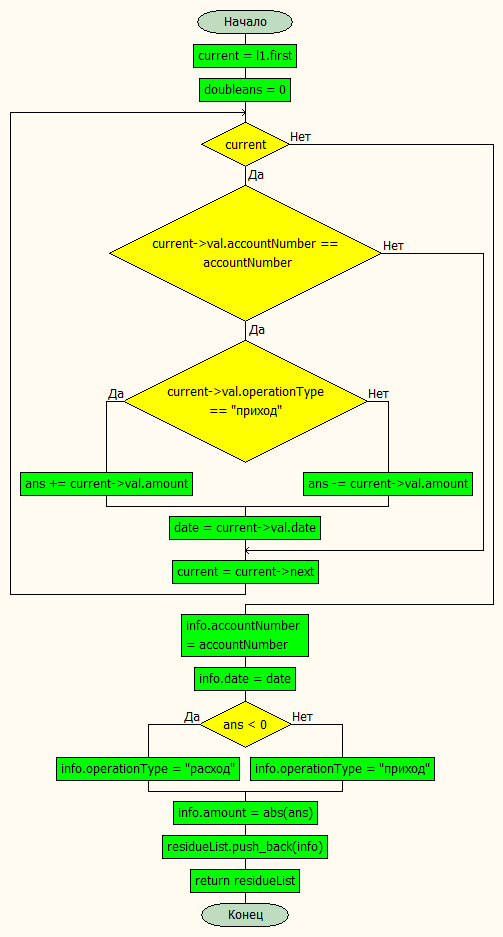


Рисунок 11 – Метод создания списка конечного состояния счета

### 2.1.5 Ожидаемые тесты программы

Алгоритм программы построим так: создадим два пустых списка l1 и l2, l2 оставим пустым, в l1 запишем некоторое количество банковских счетов.

Начальные данные для l1 представлены на рис. 12.

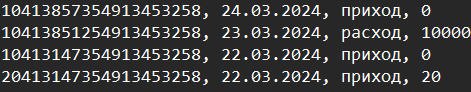


Рисунок 12 – Начальные данные

Далее проверим оба списка на пустоту и будем вызывать все методы для l1 последовательно. Ожидаемый результат проверки l2 на пустоту – правда. Таблица ожидаемых результатов для l1 представлена в табл. 1.

Таблица 1 – Ожидаемые результаты

|  |  |
| --- | --- |
| Метод | Результат для l1 |
| is\_empty() | true |
| push\_back(Info \_val) | 10413857354913453258, 24.03.2024, приход, 0  10413851254913453258, 23.03.2024, расход, 10000  10413147354913453258, 22.03.2024, приход, 0  20413147354913453258, 22.03.2024, приход, 20 |
| is\_empty() | false |
| find(“10413851254913453258”) | Указатель на 3ий узел |
| remove(“20413147354913453258”) | 10413857354913453258, 24.03.2024, приход, 0  10413851254913453258, 23.03.2024, расход, 10000  10413147354913453258, 22.03.2024, приход, 0 |
| left\_print() | Account: 1041385735491345325, Date: 24.03.2024, Operation: приход, Amount: 0  Account: 1041385125491345325, Date: 23.03.2024, Operation: расход, Amount: 10000  Account: 1041314735491345325, Date: 22.03.2024, Operation: приход, Amount: 0 |
| Метод | Результат для l1 |
| right\_print() | Account: 1041314735491345325, Date: 22.03.2024, Operation: приход, Amount: 0  Account: 1041385125491345325, Date: 23.03.2024, Operation: расход, Amount: 10000  Account: 1041385735491345325, Date: 24.03.2024, Operation: приход, Amount: 0 |
| Insert(“111111111111111111111, 26.03.2024, расход, 0”,&l1) | 111111111111111111111, 26.03.2024, расход, 0  10413857354913453258, 24.03.2024, приход, 0  10413851254913453258, 23.03.2024, расход, 10000  10413147354913453258, 22.03.2024, приход, 0 |
| removeZeroBalanceAccounts(l1) | 10413851254913453258, 23.03.2024, расход, 10000 |
| createResidueList(“1041385125491345325”, l1) | Account: 1041385125491345325, Date: 23.03.2024, Operation: расход, Amount: 10000 |

## 2.2 Реализация алгоритма на языке C++

Реализация основных методов для списка представлена на рис. 13 и 14.

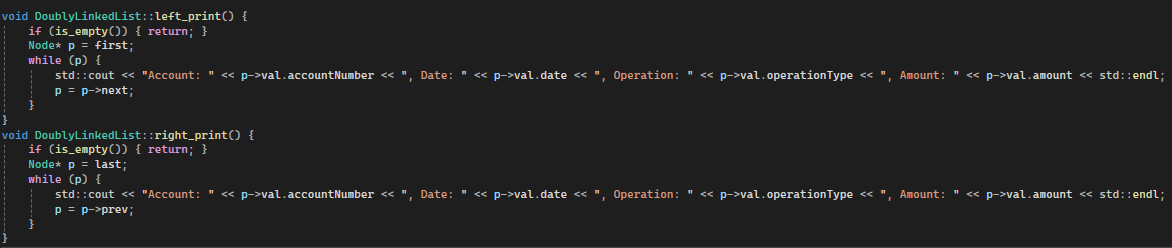


Рисунок 13 – Реализация стандартных методов

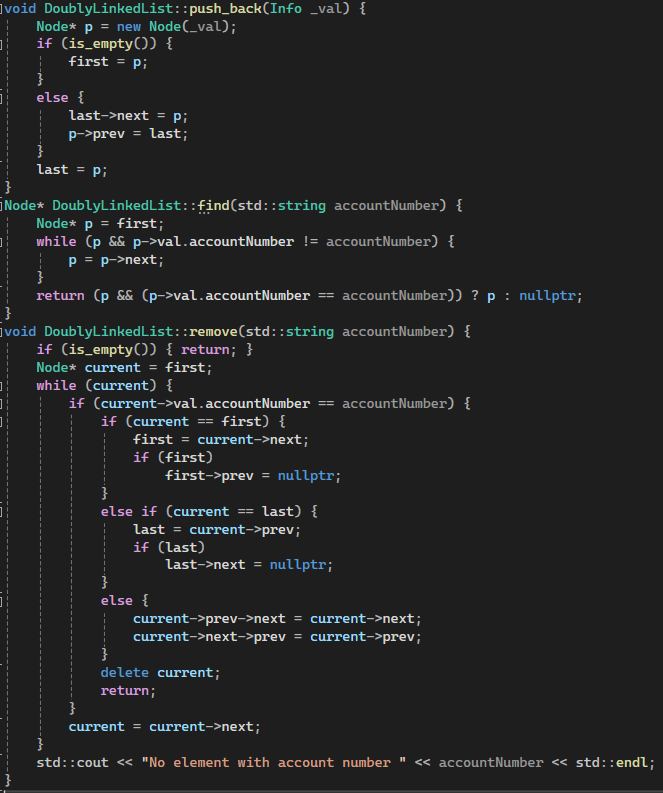


Рисунок 14 – Реализация стандартных методов

Реализация дополнительных методов представлена на рис. 15 и 16.

Рисунок 15 – Реализация дополнительных методов

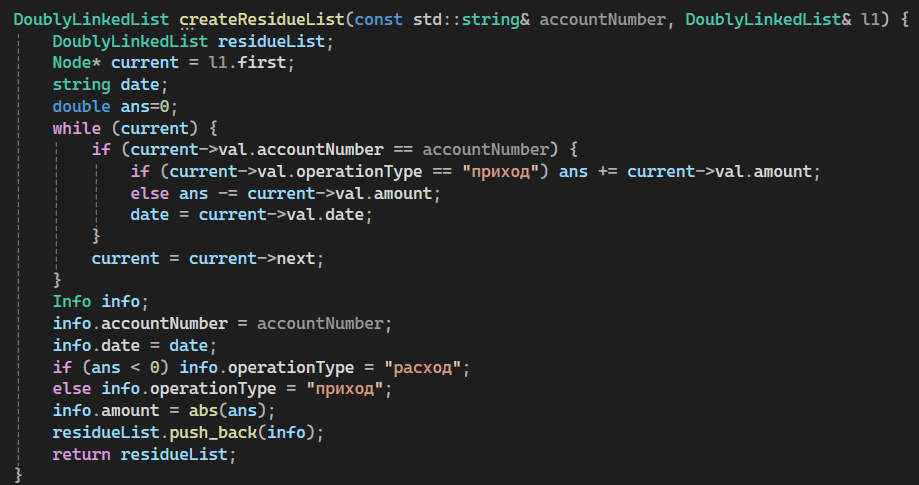


Рисунок 16 – Реализация дополнительных методов

Основной алгоритм программы для проверки корректности работы всех методов представлен на рис. 17.

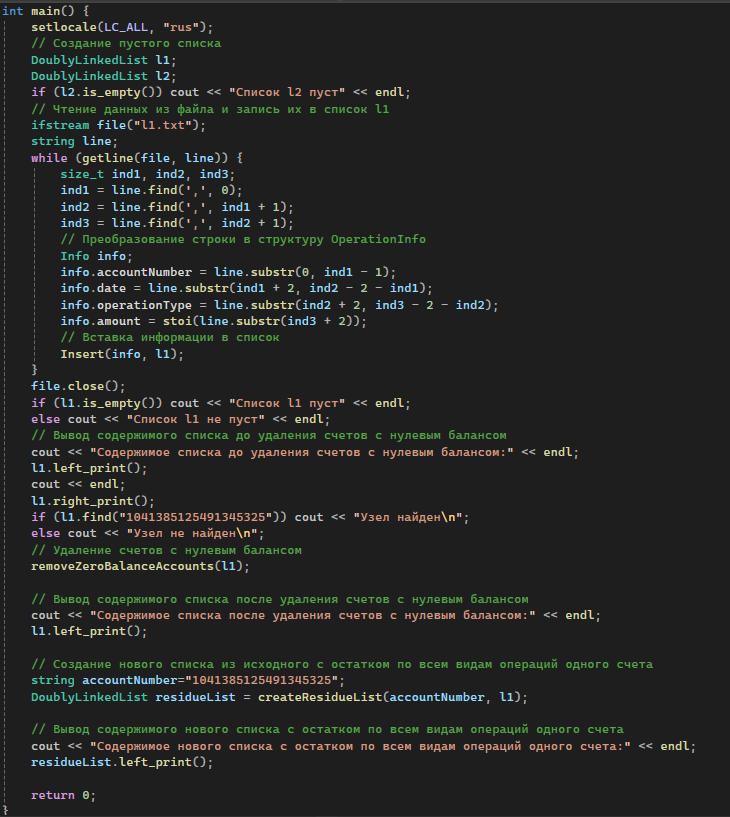


Рисунок 17 – Основной алгоритм программы

## 2.3 Тестирование программы

Тестовые данные для программы представлены на рис. 18.

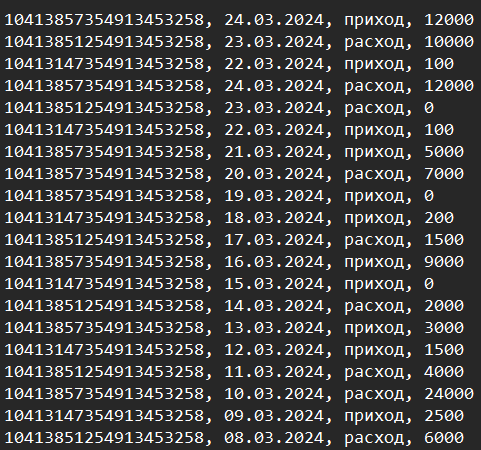


Рисунок 18 – Входные данные

Результат тестирования алгоритма на входных данных представлен на рис. 19 и 20.

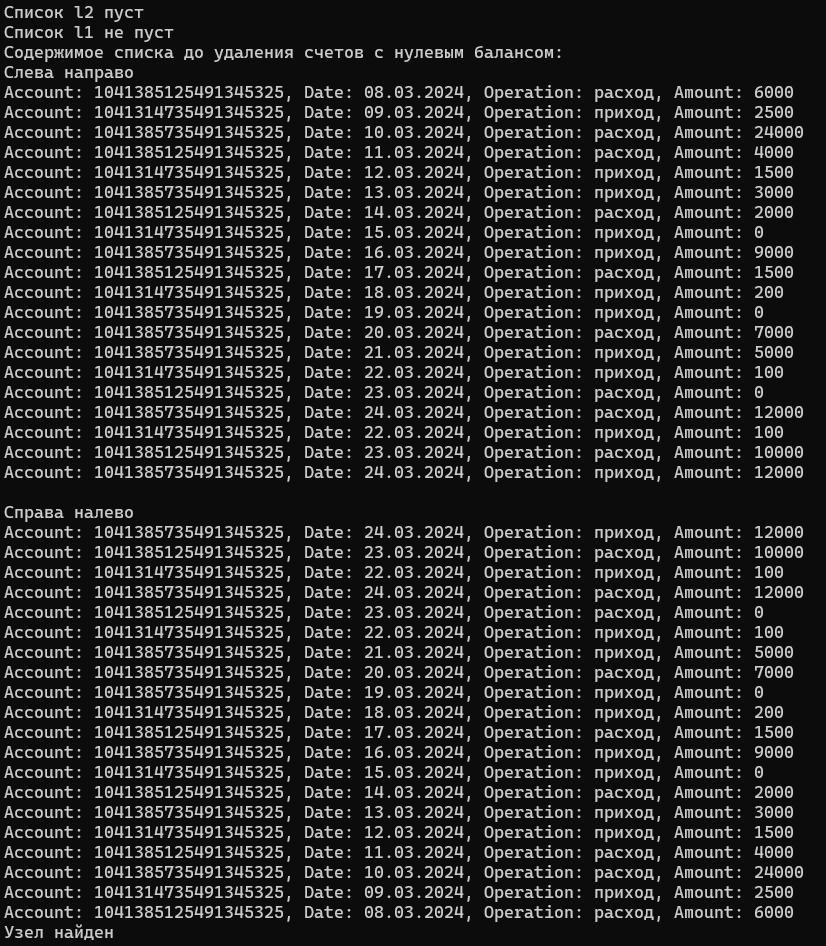


Рисунок 19 – Результат тестирования основных методов и метода Insert

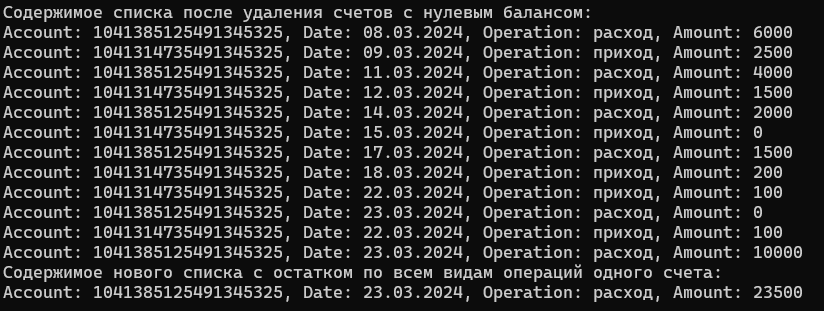


Рисунок 20 – Результат тестирования дополнительных методов и метода push\_back

Таким образом, алгоритм обрабатывает входные данные корректно.

# 3 ВЫВОДЫ

В ходе работы были реализованы структуры для работы с двунаправленными списками, написаны основные методы для работы с данными в них, а также 3 дополнительных функции для обработки данных, а именно: вставка объекта в начало списка, удаление из списка узлов с нулевыми счетами, создание нового списка для конечного результата одного из счетов.

Таким образом, были получены теоретические знания о двунаправленных списках, а также практические навыки и умения для работы с ними, и их реализации в программах на C++.

# 4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. Сартаков М.В., ПР-1.1 (Теоретическая сложность алгоритма)М., МИРЭА — Российский технологический университет – 12 с. - URL: <https://online-edu.mirea.ru/pluginfile.php?file=%2F1042738%2Fmod_assign%2Fintroattachment%2F0%2FПР1.1%20%28Теоретическая%20сложность%20алгоритма%29.pdf&amp;forcedownload=1> (дата обращения: 15.02.2024). - Режим доступа: Электронно-облачная система – Cloud MIREA РТУ МИРЭА. - Текст: электронный.
2. Рысин М.Л., Сартаков М.В., Туманова М.Б., Введение в структуры и алгоритмы обработки данных. Ч. 1 - учебное пособие, 2022, МИРЭА – Российский технологический университет. – 2022, 109с. – URL: [file:///C:/Users/borga/Downloads/Рысин%20М.Л.%20и%20др.%20Введение%20в%20структуры%20и%20алгоритмы%20обработки%20данных.%20Ч.%201%20-%20учебное%20пособие,%202022.pdf](C://Users/borga/Downloads/Рысин%20М.Л.%20и%20др.%20Введение%20в%20структуры%20и%20алгоритмы%20обработки%20данных.%20Ч.%201%20-%20учебное%20пособие,%202022.pdf) (дата обращения: 15.02.2024 ). – Режим доступа: Электронно-облачная система – Cloud MIREA РТУ МИРЭА. - Текст: электронный.