Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

им. В.И. Ульянова (Ленина)»

кафедра систем автоматизированного проектирования

**ОТЧЕТ**

**по лабораторной работе**

**по дисциплине «Программирование»**

**Тема: «Реализация связных списков»**

Выполнил : Ахметзянов Дамир Альбертович

Группа № 1301

Преподаватель: Родионова Е. А.

Санкт-Петербург

2022

Оглавление

[Постановка задачи](#_heading=h.1fob9te) **3**

[Описание реализуемого класса и методов и оценка временной сложности](#_heading=h.2et92p0) **3**

[Пример работы программы](#_heading=h.tyjcwt) **6**

[Листинг](#_heading=h.3dy6vkm) **6**

# Постановка задачи

Двусвязный список методов/функций, которые реализует каждый вариант (приведено для целочисленного типа данных):

1. добавление в конец списка
2. добавление в начало списка
3. удаление последнего элемента
4. удаление первого элемента
5. добавление элемента по индексу (вставка перед элементом, который был ранее доступен по этому индексу)
6. получение элемента по индексу
7. удаление элемента по индексу
8. получение размера списка
9. удаление всех элементов списка
10. замена элемента по индексу на передаваемый элемент
11. проверка на пустоту списка
12. вставка другого списка в список, начиная с индекса

# Описание реализуемого класса и методов и оценка временной сложности

Структуры:

| Название и тип | Описание |
| --- | --- |
| class ListItem | Элемент списка, в нем хранятся данные и ссылки на следующий и предыдущий элементы и 4 метода: получение предыдущего/следующего элемента, изменение предыдущего/следующего элемента |
| class List | Сам список, хранит ссылки на первый и последний элемент. В нем содержатся следующие методы. |

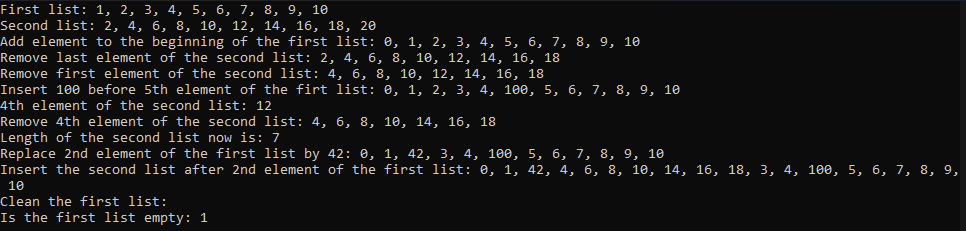
Методы

| Название метода | описание | Оценка временной сложности |
| --- | --- | --- |
| push | Добавляет в конец списка еще один элемент перезаписывая ссылки прошлого последнего элемента, вставляемого элемента и ссылку на последний элемент списка. | O(1) |
| isEmpty | Проверка на наличие узлов. Проверяется с помощью ссылки на первый элемент. | O(1) |
| unshift | Добавляет в начало списка еще один узел перезаписывая ссылки следующего первого элемента, вставляемого элемента и ссылку на первый элемент списка. | O(1) |
| shift | Удаляет первый элемент удаляя сам узел с его значением. И меняет ссылку на первый элемент списка и следующий в новом узле. | O(1) |
| pop | Удаляет последний элемент удаляет все его связи и возвращает его. При этом меняет ссылку на последний элемент списка и удаляет ссылку на следующий элемент у нового последнего элемента | O(1) |
| insert | Смещает узлы перезаписывая ссылки двух соседних узлов, прикрепляя их к новому узлу. В новом узле так же записываются ссылка на прошлый и следующий элемент. | O(n) |
| remove | Удаляет узел, перезаписывая ссылки друг на друга соседних узлов. Удаляет все связи удаляемого узла и возвращает его. | O(n) |
| clean | Удаляет все узлы, перебирая их с начала до конца и удаляя. | O(n) |
| toString | Преобразует список в строку, содержащую значения всех элементов через запятую | O(n) |
| at | Получает узел с заданным индексом | O(n) |
| replace | Заменяет значение конкретного узла, заменяя связи соседних узлов и нового узла. Возвращает замененный узел | O(n) |
| length | Возвращает размер списка, перебирая все его элементы. | O(n) |
| insertList | Вставляет в список другой список после элемента с заданным индексом. | O(n) |

# Пример работы программы

В файле [main.cpp](https://github.com/DamirAhm/AADS/blob/master/Bilinked%20list/main.cpp) написан ряд проверок на каждую из написанных функций.

Вывод программы:



# Листинг

<https://github.com/DamirAhm/AADS/blob/master/Bilinked%20list/List.cpp>