**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР.**

**отчет**

**по лабораторной работе №1**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: «Реализация связных списков».**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1301 |  | Ищенко Д.О. |
| Преподаватель |  | Родионова Е. А. |

Санкт-Петербург

2022

Оглавление

[Формулировка задания. 3](#_Toc1069)

[Описание методов и оценка временной сложности. 3](#_Toc11877)

[Пример работы программы. 4](#_Toc28194)

[Текст программы. 5](#_Toc29095)

# Формулировка задания.

Вариант 1.

Реализовать объект в виде связного списка с набором методов/функций.

Данные, хранящиеся в списке, могут быть любого типа на ваш выбор (например, целочисленный тип данных).

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 методов и оценка временной сложности.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя метода | Оценка сложности | Принимаемые аргументы | Описание работы |
| push\_back | O(n) | const T& value | Через указатель на первый элемент проходит через весь список и удаляет последнюю ноду. |
| push\_top | O(1) | const T& value | Создает новую ноду и перемещает указатель начала списка на нее. |
| pop\_top | O(1) |  | Перемещает указатель на начало списка на второй элемент и удаляет первый. |
| pop\_back | O(n) |  | Проходит через весь список и удаляет последний элемент. |
| push | O(n) | unsigned int index, const T& value | Проходит через index -1 первых элементов, создает новую ноду, меняет указатели. |
| get | O(n) | unsigned int index | Проходит через index-1 первых элементов и возвращает значение ноды на index позиции. |
| pop | O(n) | unsigned int index | Проходит через index-1 первых элементов и удаляет ноду на index позиции. |
| get\_len | O(n) |  | Проходит через весь список до null указателя, считая элементы. |
| empty | O(n) |  | Удаляет каждую ноду. |
| replace | O(n) | unsigned int index, const T& value | Проходит через index-1 первых элементов и заменяет значение на позиции index на value. |
| is\_empty | O(1) |  | Проверяет значение указателя на начало списка. |
| insert\_ll | O(n0 + nll)  n0 - размер исх. Списка  nll - размер вставляемого списка | const LinkedList<T>& ll, unsigned int index | Проходит через index элементов, начиная с index вставляет новые ноды с помощью метода pushback со значениями из ll списка. |

# Пример работы программы.

LinkedList<int> list1;  
 std::cout<<"create empty ll: "<<list1;  
  
  
 int arr[10] = {1,2,3,4,5,6,7,8,9,10};  
 LinkedList<int> list2(arr,10);  
 std::cout<<"create list based on int array[10] : "<<list2;  
  
 LinkedList<int> list3(list2);  
 std::cout<<"create copy of list2 throw copy constractor: "<<list3;  
  
 list3.replace(5,111);  
 std::cout << "replace 5 element in list3 to 111 :" << list3;  
 std::cout << "list 2 want changed: " << list2;  
  
 std::cout<<"\npush\_back tests:\n\n";  
 list1.push\_back(123);  
 std::cout<<"push\_back in empty list: "<<list1;  
 list2.push\_back(11);  
 std::cout<<"push\_back in nonempty list: "<<list2;  
  
 std::cout<<"\npop\_back tests:\n\n";  
 list1.pop\_back();  
 std::cout<<"pop\_back from ll with 1 elem: "<<list1;  
  
 std::cout<<"\n\ninsert ll into ll test\n\n";  
 std::cout<<"ll1: "<<list2;  
 std::cout<<"ll2: "<<list3;  
 list3.insert\_ll(list2, 3);  
 std::cout<<"list2.insert\_ll(list1, 3) ll2: "<<list3;  
 return 0;

Результат:

create empty ll:

create list based on int array[10] : 1 2 3 4 5 6 7 8 9 10

create copy of list2 throw copy constractor: 1 2 3 4 5 6 7 8 9 10

replace 5 element in list3 to 111 :1 2 3 4 5 111 7 8 9 10

list 2 want changed: 1 2 3 4 5 6 7 8 9 10

push\_back tests:

push\_back in empty list: 123

push\_back in nonempty list: 1 2 3 4 5 6 7 8 9 10 11

pop\_back tests:

pop\_back from ll with 1 elem:

insert ll into ll test

ll1: 1 2 3 4 5 6 7 8 9 10 11

ll2: 1 2 3 4 5 111 7 8 9 10

list2.insert\_ll(list1, 3) ll2: 1 2 3 1 2 3 4 5 6 7 8 9 10 11 4 5 111 7 8 9 10

# Текст программы.

[Ссылка на github](https://github.com/Nekttuman/Algosee)