**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра CАПР**

отчет

**по лабораторной работе №1**

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

**Вариант 17(1)**

Тема: Создание списка. Работа с unit-тестированием

|  |  |  |
| --- | --- | --- |
| Студент гр. 9302 |  | Михайлов А.А |
| Преподаватель |  | Тутуева.А.В |

Санкт-Петербург

2020

**Постановка задачи**:

Написание списка на языке С++. Ознакомление с фреймворком юнит-тестирования Googletest framework.

**Описание списка и его методов**:

Обычный односвязный список. Реализованы все функции, которые были изучены в ходе предыдущего семестра. Метод reverse использует функции at() и set(). Таким образом, можем работать со списком как с обычным массивом. Используем стандартный цикл для переворачивания массива.

**Таблица методов класса LinkedList**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название метода** | **Описание** | **Время работы** | **Выходные данные** |
| push\_back(int) | Вставка в конец списка | O(n) | void |
| push\_front(int) | Вставка в начало списка | O(1) | void |
| pop\_front() | Удаление последнего элемента списка | O(1) | void |
| pop\_back() | Удаление первого элемента списка | O(n) | void |
| clear() | Удаление всего списка | О(n) | void |
| insert(int, size\_t) | Вставка элемента по определенному индексу | O(n) | void |
| print\_to\_console() | Вывод всего списка в консоль, используя разделитель. | O(n) | void |
| get\_size() | Длина списка | О(1) | size\_t |
| isEmpty() | Проверяет список на пустоту | O(1) | bool |
| set(size\_t, int) | Заменяет элемент списка по указанному индексу на указанный элемент. | O(n) | void |
| at(size\_t) | Получает элемент списка по индексу | O(n) | int |
| remove(size\_t); | Удаляет элемент списка по индексу | O(n) | void |
| reverse(LinkedList) | Переворачивает список | O(n^2) | void |

**Описание uint-тестов:**

Всего написано 9 unit-тестов  
1. push-метод front.

2. push-метод back.

3. insert + push методы.

4. insert\_front. Проверка вставки в начало списка.

5. remove. Проверка очищения списка.

6. insert\_repeatly. Проверка последовательной вставки методами insert и push.

7. is\_empty. Проверка на пустой список.

8. reverse. Создаём два противоположных списка, инвертируем один. Сравниваем списки.

9. set\_front. Проверка функции set.

