**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

ОТЧЁТ

**по лабораторной работе №1**

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

**Тема: «Программирование двунаправленных связных списков»**

**Вариант 10**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9302 |  | Кнауб К.В. |
| Преподаватель |  | Тутуева А.В. |

Санкт-Петербург

2020

## Постановка задачи. Описание реализуемого класса и методов.

Задание: Реализовать класс двусвязного списка с набором методов.

Для этого мне понадобились классы «Node» и «List».

class Node

|  |  |
| --- | --- |
| **Элемент** | **Описание** |
| Node \*next; | Указатель на следующий элемент списка |
| Node \*prev; | Указатель на предыдущий элемент списка |
| int data; | Целочисленные данные, хранящиеся в списке |

**class List**

|  |  |
| --- | --- |
| **Элемент** | **Описание** |
| Node \*head; | Указатель на начало списка |
| Node \*tail; | Указатель на конец списка |
| size\_t size; | Переменная, хранящая размер списка |

|  |  |  |
| --- | --- | --- |
| **Метод** | **Описание** | **Оценка временной сложности** |
| void push\_back(int) | Добавление в конец списка | O(1) |
| void push\_front(int) | Добавление в начало списка | O(1) |
| void pop\_back() | Удаление последнего элемента | O(1) |
| void pop\_front() | Удаление первого элемента | O(1) |
| void insert(int, size\_t) | Добавление элемента по индексу | O(n) |
| int at(const int) | Получение элемента по индексу | O(n) |
| void remove(size\_t) | Удаление элемента по индексу | O(n) |
| size\_t get\_size() | Получение размера списка | O(1) |
| void print\_to\_console() | Вывод элементов в консоль через разделитель | O(n) |
| void clear() | Удаление всех элементов списка | O(n) |
| void set(size\_t, int) | Замена элемента по индексу на передаваемый элемент | O(n) |
| bool isEmpty() | Проверка на пустоту списка | O(1) |
| bool contains(List) | Проверка на содержание другого списка в списке | O(n) |

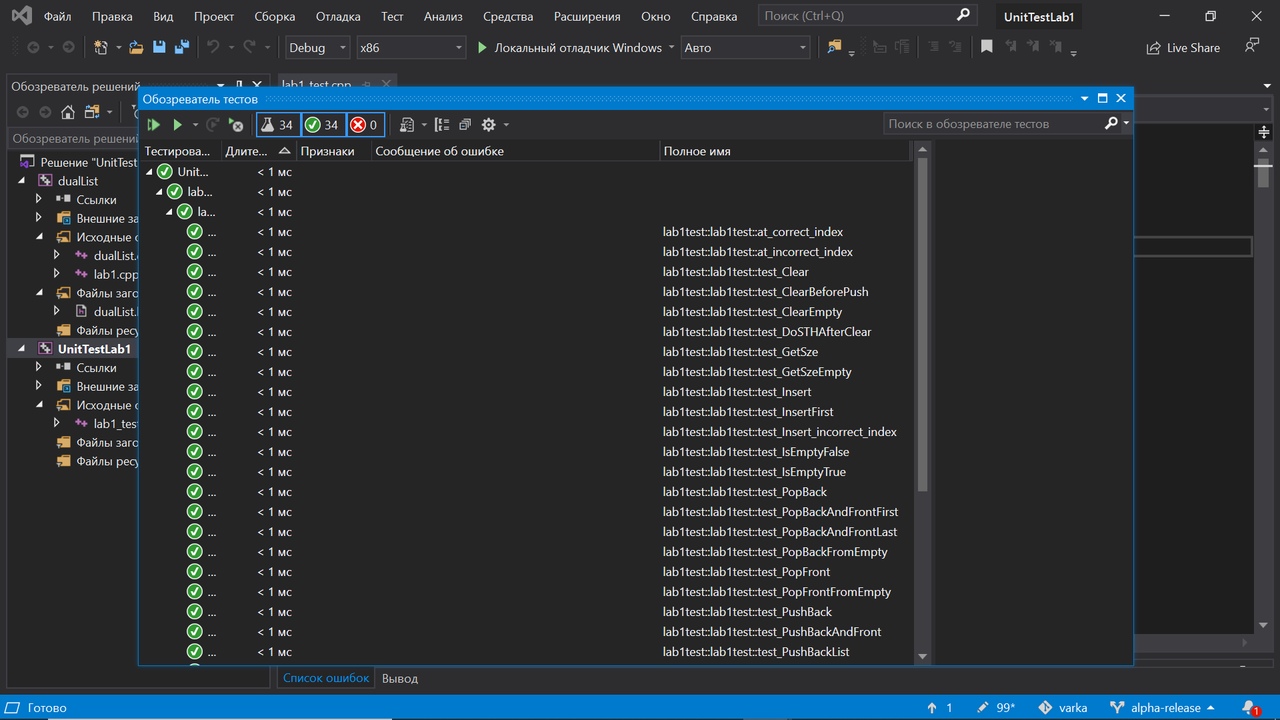
## Описание реализованных unit-тестов.

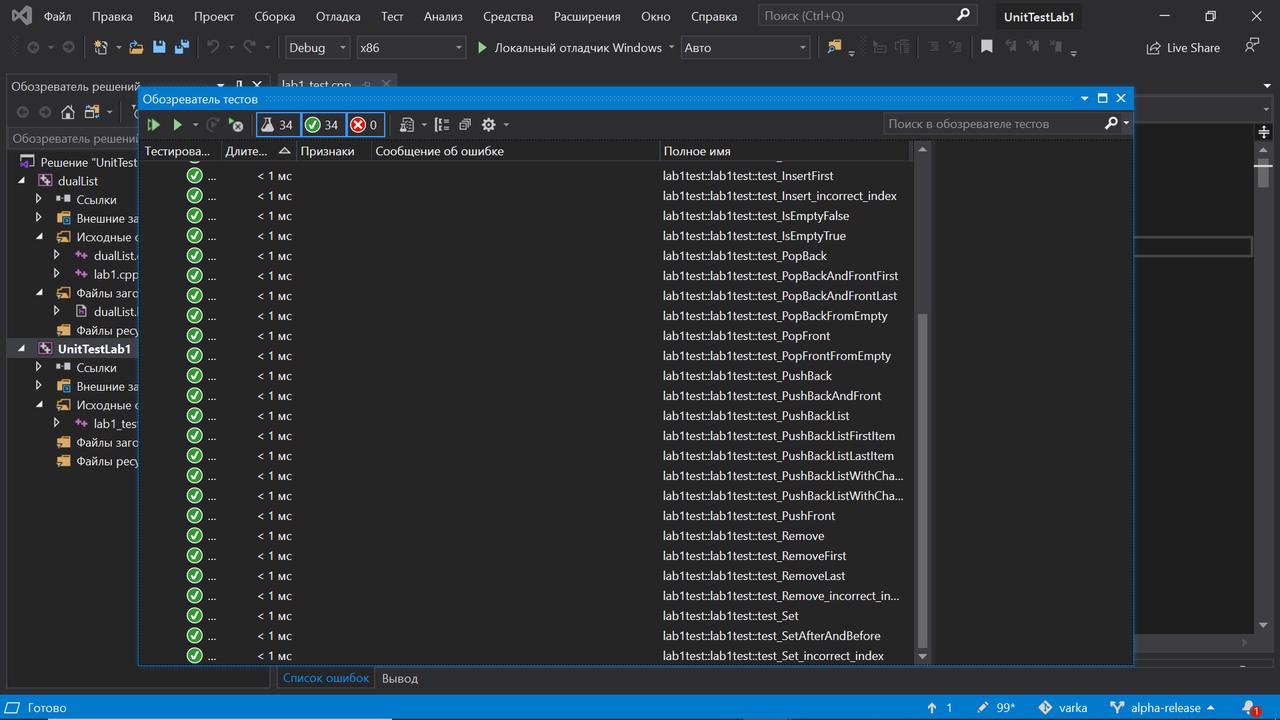
|  |  |
| --- | --- |
| Название теста | Что проверяет |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Листинг

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## Результат выполнения всех unit-тестов





## Вывод

В данной лабораторной работе мы научились реализовывать связные списки и работать с ними. Также мы научились проверять работу написанной программы с помощью unit-тестов.