# 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: Списки

# Вариант 7

|  |  |  |
| --- | --- | --- |
| Студент гр. 8302 |  | Гришаков И.В. |
| Преподаватель |  | Тутуева А.В. |

Санкт-Петербург

2019

1. Постановка задачи. Описание реализуемого класса и методов

Требуется реализовать список на языке C++, используя классы и функции, для всех функций должны быть написаны unit-тесты. Для написания unit-тестов используется библиотека Google Tests.

Для реализации списка были написаны два класса:

1. Node – элемент списка,
2. List – сам список.

Список функций класса Node:

int get\_data(); // получение данных элемента

void set\_data(); // установка данных элемента

Node\* get\_next(); // получение ссылки на следующий элемент

void set\_next(Node\*); // установка ссылки следующий элемент

Поля:

int data – данные элемента

Node\* next – ссылка на следующий элемент

Список функций класса List:

void push\_back(int); // добавление в конец списка

void push\_front(int); // добавление в начало списка

void pop\_back(); // удаление последнего элемента

void pop\_front(); // удаление первого элемента

void insert(int, size\_t) // добавление элемента по индексу (вставка перед элементом, который был ранее доступен по этому индексу)

int at(size\_t); // получение элемента по индексу

void remove(size\_t); // удаление элемента по индексу

size\_t get\_size(); // получение размера списка

void print\_to\_console(); // вывод элементов списка в консоль через разделитель, не использовать at

void clear(); // удаление всех элементов списка

void set(size\_t, int); // замена элемента по индексу на передаваемый элемент

bool isEmpty(); // проверка на пустоту списка

void add\_first(int data); // добавление первого элемент в список

void push\_front(List); // вставка другого списка в начало

Поля:

int size – размер списка

Node \*head – ссылка на первый элемент списка

Node \*tail – ссылка на последний элемент списка

1. Оценка временной сложности методов

void push\_back(int); // O(1)

void push\_front(int); // O(1)

void pop\_back(); // O(N)

void pop\_front(); // O(1)

void insert(int, size\_t) // O(N)

int at(size\_t); // O(N)

void remove(size\_t); // O(N)

size\_t get\_size(); // O(1)

void print\_to\_console(); // O(N)

void clear(); // O(N)

void set(size\_t, int); // O(N)

bool isEmpty(); // O(1)

void add\_first(int data); // O(1)

void push\_front(List); // O(1)

1. Описание реализованных unit-тестов

Для класса Node были реализованы следующие тесты:

Тест на создание.

Тест на получение данных элемента и установки следующего.

Для класса List были реализованы следующие тесты:

Тест на создание.

Тест на добавление элемента.

Тест на удаление элемента и списка.

Тест на добавление списка в начало другого списка.

1. Пример работы

Код примера:

List list1;

list1.push\_back(4);

list1.push\_back(5);

list1.push\_back(6);

List list2;

list2.push\_front(3);

list2.push\_front(2);

list2.push\_front(1);

list1.push\_front(list2);

for (int i = list1.get\_size(); i > 0; i--) {

cout << list1.at(i - 1);

}

Вывод:

654321

1. Листинг

Листинг в IDE.