

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Объектно-ориентированное программирование»
Тема: Контейнеры

Студентка гр. 7303

Аплачкина Е.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2019

Цель работы:

Необходимо реализовать конструкторы и деструктор для контейнера вектор и список. Поведение реализованных функций должно быть таким же, как у классов `std::vector` и `std::list`.

Ход работы:

1. Были дописаны следующие методы для класса `vector`:

- 1) Конструктор `explicit vector(size_t count = 0);`
- 2) Конструктор `vector(InputIterator first, InputIterator last);`
- 3) Конструктор `vector(std::initializer_list <Type> init);`
- 4) Конструктор копирования `vector(const vector& other);`
- 5) Конструктор перемещения `vector(vector&& other);`
- 6) Деструктор `~vector();`
- 7) Оператор присваивания копирования `vector& operator= (const vector& other);`
- 8) Оператор присваивания перемещения `vector& operator=(vector&& other);`
- 9) Метод `void assign(InputIterator first, InputIterator last);`
- 10) Метод `void resize(size_t count);`
- 11) Метод `iterator erase(const_iterator pos);`
- 12) Метод `iterator erase(const_iterator first, const_iterator last);`
- 13) Метод `void resize(size_t count);`
- 14) Метод `iterator insert(const_iterator pos, const Type& value);`
- 15) Метод `iterator insert(const_iterator pos, InputIterator first, InputIterator last);`
- 16) Метод `void push_back(const value_type& value);`

2. Были дописаны следующие методы для класса `list`:

- 1) `void push_back(const value_type& value);`
- 2) `void push_front(const value_type& value);`
- 3) `reference front();`
- 4) `const_reference back();`

- 5) void pop_front();
- 6) void pop_back();
- 7) void clear();
- 8) bool empty() const;
- 9) size_t size() const;
- 10) Деструктор ~list();
- 11) Конструктор копирования list(const list& other);
- 12) Конструктор перемещения list(list&& other);
- 13) Оператор присваивания list& operator= (const list& other);
- 14) reference front();
- 15) reference back();
- 16) bool empty() const;
- 17) void clear();
- 18) iterator insert(iterator pos, const Type& value);
- 19) iterator erase(iterator pos);

3. Также были дописаны методы для класса list_iterator:

- 1) Оператор присваивания list_iterator& operator = (const list_iterator& other);
- 2) Оператор равенства bool operator == (const list_iterator& other) const;
- 3) Оператор неравенства bool operator != (const list_iterator& other) const;
- 4) Оператор разыменования reference operator * ();
- 5) Оператор получения адреса элемента pointer operator -> ();
- 6) Постфиксный оператор перехода к следующему итератору list_iterator& operator ++ ();
- 7) префиксный оператор перехода к следующему итератору list_iterator operator ++ (int);

Выводы

В ходе выполнения лабораторной работы были изучены и реализованы контейнеры вектор и список. Были реализованы основные функции для работы с ними. Поведение реализованных функций соответствует классам `std::vector` и `std::list`.