

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.

Студент гр. 7304

Моторин Е.В.

Преподаватель

Размочаева Н.В

Санкт-Петербург

2019

Цель работы:

Изучить реализация контейнеров `list` и `vector` в языке программирования C++.

Задача:

Реализовать конструктор, деструктор, операторы присваивания, функцию `assign`, функцию `resize`, функцию `erase`, функцию `insert` и функцию `push_back`. Поведение реализованных функций должно быть так же как и у `std::vector`. Реализовать список с функциями: вставка элемента в голову, вставка элемента в хвост, получение элемента из головы, получение элемента из хвоста, удаление из головы, из хвоста, очистка списка, проверка размера, деструктор, конструктор копирования, конструктор перемещения, оператор присваивания, `insert`, `erase`, а так же итераторы для списка: `=`, `==`, `!=`, `++` (постфиксный и префиксный), `*`, `->`. Поведение реализованных функций должно быть таким же, как у класса `std::list`.

Ход работы:

■ List.

- Функции вставки элемента в голову и в хвост.
- Функции получение элемента из головы и из хвоста.
- Функции удаления из головы, удаления из хвоста.
- Функции очистки списка, проверки размера.
- Деструктор, конструктор копирования, конструктор перемещения, оператор присваивания.
- Операторы для итератора списка: `=`, `==`, `!=`, `++`, `*`, `->`.
- Функции удаления элемента и вставка элемента в произвольное место.

■ Vector.

- Все функции соответствующие классу `std::vector`.
- Конструкторы копирования и перемещения.
- Оператор присваивания и функция `assign`.
- Функции вставки элемента в голову и в хвост.
- Функции получение элемента из головы и из хвоста.
- Функции удаления из головы, удаления из хвоста.
- Функции очистки списка, проверки размера.

Результат:

- LIST.

```
CREATE NEW LIST:
LIST: [ 7 9 13 18 10 12 4 18 3 9 0 5 12 2 7 3 7 9 0 12 3 9 9 17 0 13 19 18 16 15 ]

*****

PUSH FRONT '123'
LIST: [ 123 7 9 13 18 10 12 4 18 3 9 0 5 12 2 7 3 7 9 0 12 3 9 9 17 0 13 19 18 16 15 ]

*****

SIZE: 31
*****

POP BACK
LIST: [ 123 7 9 13 18 10 12 4 18 3 9 0 5 12 2 7 3 7 9 0 12 3 9 9 17 0 13 19 18 16 ]

*****

POP FRONT
LIST: [ 7 9 13 18 10 12 4 18 3 9 0 5 12 2 7 3 7 9 0 12 3 9 9 17 0 13 19 18 16 ]

*****

CLEAR
LIST:
*****

PUSH BACK -1, 10 TIMES
List: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

*****

COPY CONSTRUCTOR
LIST: [ -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ]

*****
```

- VECTOR.

```
CONSTRUCTOR
7 49 73 58 130 72 144 78 123 109
*****

COPY -7 49 73 58 130 72 144 78 123 109
*****

RESIZE
7 49 73 58 130 72 144 78 123 109 0 0 0 0
*****

RESIZE -7 49 73 58 130 72 144 78 123 109 0 0 0 0 0 0 0 0 0
-----

ERASE
7 49 73 58 130 72 144 78 123 109 0 0 0 0 0 0 0 0
*****

ERASE
7 0 0
```

INSERT

7 6 0 0

INSERT

7 6 49 73 58 130 72 144 78 123 0 0

PUSH BACK

7 6 49 73 58 130 72 144 78 123 0 0 190

Вывод:

Таким образом, в ходе данной лабораторной работы была изучена реализация таких контейнеров, как вектор и список, были реализованы основные функции для работы с ними, такие как вставка, удаление произвольных элементов, изменение размера. Разработаны необходимые конструкторы и итераторы для работы с данными контейнерами.