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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Обьектно-ориентированное программирование»

Тема: Контейнеры

Студентка гр. 7304 _____ Нгуен Т.Т. Зуен

Преподаватель _____ Расмочаева Н.В.

Санкт-Петербург

2019

1. Цель работы:

Изучить реализацию контейнеры vector и list в языке программирования C++ .

2. Задание:

Vector:

Необходимо реализовать конструкторы, деструктор, операторы присваивания, функцию assign, функции resize, функцию erase, функцию insert и функцию push_back для контейнера вектор. Поведение реализованных функций должно быть таким же, как у класса std::vector.

List:

Необходимо реализовать список со функциями: вставка элементов в голову и в хвост, получение элемента из головы и из хвоста, удаление из головы, хвоста, очистка ипроверка размера, деструктор, конструктор копирования, конструктор перемещения, оператор присваивания. И реализовать итератор для списка, erase и insert. Для краткости реализации можно ограничиться однонаправленным изменяемым (неконстантным) итератором. Необходимо реализовать операторы: =, ==, !=, ++ (постфиксный и префиксный), *, ->. Поведение реализованных функций должно быть таким же, как у класса std::list.

3. Ход работы:

Vector:

В ходе реализация vector были созданы следующие финкии:

- Конструкторы и деструктор: реализованние конструктуры ключают в себя – конструктур копирования, присваивания и перемещения.
- Оператор присваивания и функцию assign - заменяет содержимое с копиями тех, кто в диапазоне [first, last)
- Функции изменения размера и стирания элементов: resize и erase
    - Resize: Изменяет количество хранимых элементов
    - Erase: Удаляет указанные элементы из контейнера.
        - Удаляет элемент в позиции pos - итератор указывающий на удаляемый элемент.
        - Удаляет элементы в диапазоне [first; last) - диапазон удаляемых элементов.

- Функцию insert: Вставляет элементы:
    - Вставляет value перед элементом, на который указывает pos.
    - Вставляет элементы из диапазона [first, last) перед элементом, на который указывает pos.
- Функцию push_back: Добавляет данный элемент value до конца контейнера.

Поведение реализованных функций должно быть таким же, как у класса std::vector.

List:

В ходе реализация list были созданы следующие финкии:

- Вставка элементов в голову и в хвост:
    - push_front: вставляет элементы в начало списка.
    - push_back: добавляет элемент в конец.
- Получение элемента из головы и из хвоста: возвращает элемент из голоы или из хвоста.
- Удаление из головы, хвоста и очистка
    - pop_front: удаляет первый элемент.
    - pop_back: удаляет последний элемент.
- Проверка размера и очистки спискаю.
- Деструктор, конструктор копирования, конструктор перемещения и оператор присваивания.
- Операторы: =, ==, !=, ++ (постфиксный и префиксный), *, -> для списка.
- Вставку элементов: вставляет value перед элементом, на который указывает pos и возвращает итератор, указывающий на вставленный value.
- Удаление элементов: удаляет элемент в позиции pos и возвращает итератор, следующий за последним удаленным элементом.

Поведение реализованных функций должно быть таким же, как у класса std::list

4. Результаты:

```
int main()
{
    stepik::vector<int> vec(10);
    for (size_t i = 0; i < vec.size(); i++) {
        vec[i] = i + 1;
    }
    cout << "Vector: ";
    for (size_t i = 0; i < vec.size(); i++) {
        cout << vec[i] << " ";
    }cout << endl;

    vec.erase(vec.begin()+2);
    cout << "Erase: ";
    for (size_t i = 0; i < vec.size(); i++) {
        cout << vec[i] << " ";
    }cout << endl;

    system("pause");
    return 0;
}
```

```
Vector: 1 2 3 4 5 6 7 8 9 10
Erase: 1 2 4 5 6 7 8 9 10
Press any key to continue . . .
```

```cpp
int main()
{
    stepik::list<int> a;
    for (int i = 0; i < 10; i++) {
        a.push_back(i + 1);
    }
    cout << "List: ";
    a.print();

    a.pop_back();
    cout << "Pop back: ";
    a.print();

    a.insert(a.end(), 3);
    cout << "Insert: ";
    a.print();

    stepik::list_iterator<int> iter = a.begin()
    iter++;
    iter = a.erase(iter);
    cout << "Erase: ";
    a.print();

    system("pause");
    return 0;
}
```

```
List: 1 2 3 4 5 6 7 8 9 10
Pop back: 1 2 3 4 5 6 7 8 9
Insert: 1 2 3 4 5 6 7 8 9 3
Erase: 1 3 4 5 6 7 8 9 3
Press any key to continue . . .
```

5. Выводы:

В ходе данной лабораторной работе была изучена реализацию контейнеры vector и list. Поведение реализованных функций должно быть таким же, как у класса std::vector и std::list из стандартной библиотеки C++ и полученные результаты.