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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Объектно-ориентированное программирование»

Тема: Контейнеры данных

Студент гр. 7303 _____ Шаломов А.Д.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург

2019

Цель работы.

Реализация контейнеров данных, таких как вектор и список, а также их базовых методов, аналогичных по поведению методам для контейнеров std::vector и std::list.

Постановка задачи.

Для класса vector необходимо реализовать: конструкторы, деструктор, функции assign, resize, erase, insert, push_back и оператор присваивания.

Для класса list необходимо реализовать: конструкторы, деструктор, функции front, push_front, pop_front, back, push_back, pop_back, clear, empty, insert, erase и оператор присваивания, а также итератор для списка с операторами =, ==, !=, ++ (постфиксный и префиксный), *, ->.

Выполнение работы.

Класс vector содержит два приватных поля: m_first и m_last – указатели на первый и последний элемент соответственно. Ниже приведен порядок реализации функций.

1) Реализованы следующие конструкторы: принимающий размер вектора (по умолчанию 0); принимающий указатели на начало и конец копируемой области массива; принимающий список инициализации; копирования и перемещения, а также деструктор.
2) Реализованы операторы присваивания (копирующий и перемещающий), а также метод assign.
3) Реализованы методы resize, изменяющий размер вектора на заданный и перегруженный erase, удаляющий либо элемент в заданной позиции, либо все элементы в заданном диапазоне.
4) Реализованы методы push_back, добавляющий один элемент в конец вектора, и перегруженный метод insert, добавляющий элемент в массив справа от заданной позиции, либо набор элементов.

Класс list представляет собой двусвязный список и содержит два приватных поля: m_head, m_tail – указатели на первый и последний элемент списка типа node, структуры данных, хранящей указатели на предыдущий и следующий элемент списка, а также собственно значение элемента. Ниже приведен порядок реализации функций.

1) Реализованы методы push_back, push_front, pop_back, pop_front, front, back, clear, size, empty.
2) Реализованы конструкторы копирования, перемещения, оператор присваивания и деструктор.
3) Реализован итератор списка с операторами: =, ==, !=, ++(постфиксный и префиксный), *, ->
4) С использованием итераторов реализованы методы insert, вставляющий элемент в список справа от заданного, и erase, удаляющий заданный элемент.

Выводы.

В ходе работы на языке C++ были реализованы такие структуры данных, как вектор и список, а также основные функции для работы с ними: конструкторы, деструкторы, методы для вставки и удаления элементов и другие.