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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Объектно-ориентированное программирование»

Тема: «Контейнеры вектор и список»

Студент гр. 7304 _____ Петруненко Д.А

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург

2019

Цель работы

Изучить реализацию таких контейнеров как vector и list в языке программирования c++.

Задача

Реализовать конструкторы, деструктор, операторы присваивания, функцию assign, функцию resize, функцию erase, функцию insert и функцию push_back. Поведение реализованных функций должно быть таким же, как у класса std::vector.

Реализовать список со следующими 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 также итераторы для списка: =, ==, !=, ++ (постфиксный и префиксный), *, ->. Поведение реализованных функций должно быть таким же, как у класса std::list.

Ход работы

1. Vector: Все функции были реализованы в соответствие с поведением класса std::vector.
    1.1. Были реализованы конструкторы копирования и перемещения.
    1.2. Были реализованы операторы присвоения и функция assign.
    1.3. Были реализованы следующие функции: resize, erase, push_back, insect.
2. List: Все функции были реализованы в соответствие с поведением класса std::list.
    2.1. Были реализованы функции: вставка элемента в голову, вставка элемента в хвост, получение элемента из головы, получение элемента из хоста, удаление из головы, удаление из хвоста, очистка списка, проверка размера.

2.2. Были реализованы: деструктор, конструктор копирования, конструктор перемещения, оператор присвоения.

2.3. Были реализованы операторы для итератора списка: =, ==, !=, ++ (постфиксный и префиксный), *, ->.

2.4. Были реализованы функции удаления элемента и вставка элемента в произвольное место.

Результат работы

До поворота фигур в векторе

1. VecRec.push_back(rect);

Before turn:
Rectangle:
List of points:
(-1,-1) (-1,1) (1,1) (1,-1)
Color: (0,0,0)
A: 2 B: 2
ID = 1

2. VecRec.push_back(Paral);

Before turn:
Parallelogram:
List of points:
(-6.79904,4.25) (-4.20096,5.75) (0.799038,5.75)
(-1.79904,4.25)
Color: (0,0,0)
A: 3 B: 5
ID = 2

3. VecRec.push_back(Hexagon);

Before turn:
Regular Hexagon:
List of points:
(-2,4) (-3.5,6.59808) (-6.5,6.59808) (-8,4)
(-6.5,1.40192) (-3.5,1.40192)
Color: (0,0,0)
Radius: 3
ID = 3

После поворота фигур в векторе

1. turnVector(VecRec);	After turn: Rectangle: List of points: (1,1) (1,-1) (-1,-1) (-1,1) Color: (0,0,0) A: 2 B: 2 ID = 1
2. turnVector(VecRec);	After turn: Parallelogram: List of points: (6.79904,-4.25) (4.20096,-5.75) (-0.799038,-5.75) (1.79904,-4.25) Color: (0,0,0) A: 3 B: 5 ID = 2
3. turnVector(VecRec);	After turn: Regular Hexagon: List of points: (2,-4) (3.5,-6.59808) (6.5,-6.59808) (8,-4) (6.5,-1.40192) (3.5,-1.40192) Color: (0,0,0) Radius: 3 ID = 3

Вывод

В ходе выполнения данной лабораторной работы была изучена реализация таких контейнеров, как вектор и список, были реализованы основные функции для работы с этими контейнерами, как вставка в произвольное место, удаление произвольного элемента, изменение размера, необходимые конструкторы и итераторы для работы с этими контейнерами.