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

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Объектно-ориентированное программирование»

Тема: Создание классов

Студент гр. 3343 _____ Жучков О.Д.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы

Изучить парадигму программирования ООП и на основе полученных знаний реализовать систему классов на языке C++ для симуляции размещения кораблей и взаимодействия на игровом поле наподобие игры “Морской бой”.

Задание

a) Создать класс корабля, который будет размещаться на игровом поле. Корабль может иметь длину от 1 до 4, а также может быть расположен вертикально или горизонтально. Каждый сегмент корабля может иметь три различных состояния: целый, поврежден, уничтожен. Изначально у корабля все сегменты целые. При нанесении 1 урона по сегменту, он становится поврежденным, а при нанесении 2 урона по сегменту, уничтоженным. Также добавить методы для взаимодействия с кораблем.

b) Создать класс менеджера кораблей, хранящий информацию о кораблях. Данный класс в конструкторе принимает количество кораблей и их размеры, которые нужно расставить на поле.

c) Создать класс игрового поля, которое в конструкторе принимает размеры. У поля должен быть метод, принимающий корабль, координаты, на которые нужно поставить, и его ориентацию на поле. Корабли на поле не могут соприкасаться или пересекаться. Для игрового поля добавить методы для указания того, какая клетка атакуется. При попадании в сегмент корабля изменения должны отображаться в менеджере кораблей.

Каждая клетка игрового поля имеет три статуса:

i)неизвестно (изначально вражеское поле полностью неизвестно),

ii)пустая (если на клетке ничего нет)

iii)корабль (если в клетке находится один из сегментов корабля).

Для класса игрового поля также необходимо реализовать конструкторы копирования и перемещения, а также соответствующие им операторы присваивания.

Описание архитектурных решений и классов.

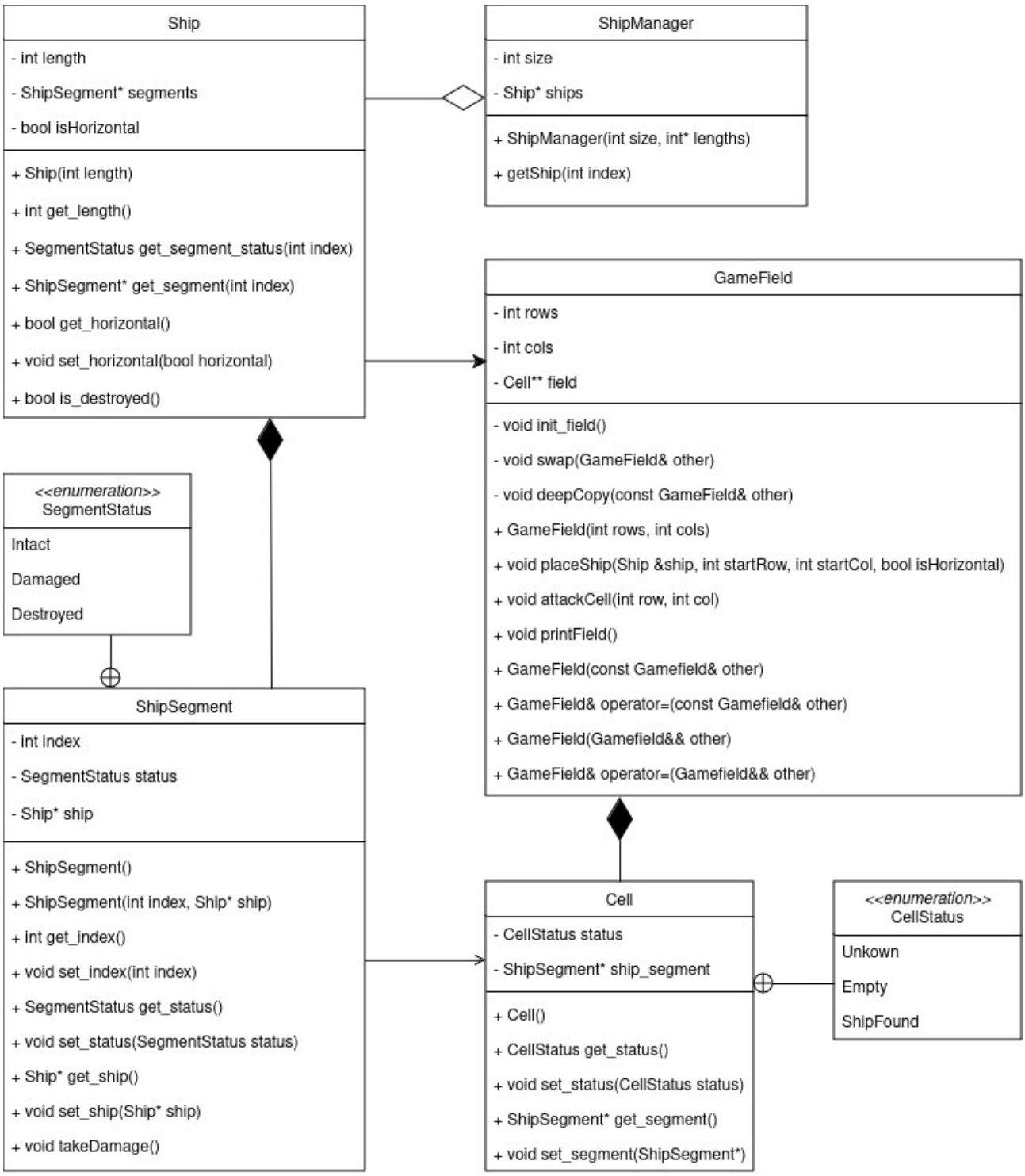
Класс Ship представляет корабль, состоящий из сегментов, которые могут быть повреждены или уничтожены.. Имеет аттрибуты: int length (длина от 1 до 4), bool isHorizontal (при размещении на поле корабль можеть принимать вертикальное или горизонтальное положение). Segments – массив объектов класса ShipSegment, представляющих собой отдельные сегменты корабля. Реализованы базовые методы для получения и изменения полей объекта, метод для проверки того, уничтожен ли корабль полностью.

ShipSegment имеет индекс, статус и связан с кораблем. Именно объекты этого класса получают нанесённый по кораблю урон через метод takeDamage. Создан данный класс для удобства взаимодействия поля с кораблём, вместо того чтобы хранить ссылки на корабли, игровое поле хранит сегменты корабля в клетках и взаимодействует только с ними.

Cell представляет отдельную клетку игрового поля. Хранит статус клетки и возможную связь с сегментом корабля. Инициализируется по умолчанию с статусом Unknown и без указателя на сегмент. Благодаря данному классу и ShipSegment все отношения между игровым полем и кораблем представлены в виде связи клетка – сегмент, поле не хранит сам объект корабля, так как для этой задачи уже существует ShipManager.

GameField представляет игровое поле с возможностью размещения кораблей, нанесения ударов и получения информации о клетках. В двумерном массиве field хранит клетки поля (Cell). Реализованы конструкторы и операторы копирования и перемещения, производится глубокое копирование, при котором также создаются копии клеток, кораблей и их сегментов.

ShipManager отвечает за создание кораблей, предоставляя методы для их доступа и управления. При инициализации принимает массив длин кораблей и инициализирует объекты Ship.



UML-диаграмма отношения классов

Выводы

В процессе выполнения работы была изучена парадигма объектно-ориентированного программирования и реализована система классов для симуляции размещения кораблей и взаимодействия на игровом поле. Созданные классы имеют четкое разделение ответственности и инкапсуляцию, что делает систему модульной, расширяемой и удобной для понимания. Была проведена работа с использованием классов при помощи языка C++, изучены принципы составления UML-диаграмм.