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

ОТЧЕТ

по лабораторной работе № 2

по дисциплине «Объектно-ориентированное программирование»

Тема: «Полиморфизм»

Студент гр. 3343 _____ Жучков О.Д.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2024

Цель работы

Изучить принцип полиморфизма в ООП, создать класс-интерфейс способности и три способности на его основе, класс менеджера способностей, реализовать набор классов-исключений для различных исключительных ситуаций.

Задание

1. Создать класс-интерфейс способности, которую игрок может применять. Через наследование создать 3 разные способности:
    a. Двойной урон - следующая атак при попадании по кораблю нанесет сразу 2 урона (уничтожит сегмент).
    b. Сканер - позволяет проверить участок поля 2x2 клетки и узнать, есть ли там сегмент корабля. Клетки не меняют свой статус.
    c. Обстрел - наносит 1 урон случайному сегменту случайного корабля. Клетки не меняют свой статус.
2. Создать класс менеджер-способностей. Который хранит очередь способностей, изначально игроку доступно по 1 способности в случайном порядке. Реализовать метод применения способности.
3. Реализовать функционал получения одной случайной способности при уничтожении вражеского корабля.
4. Реализуйте набор классов-исключений и их обработку для следующих ситуаций (можно добавить собственные):
    a. Попытка применить способность, когда их нет
    b. Размещение корабля вплотную или на пересечении с другим кораблем
    c. Атака за границы поля

Примечания:

- Интерфейс события должен быть унифицирован, чтобы их можно было единообразно использовать через интерфейс
- Не должно быть явных проверок на тип данных

Выполнение работы

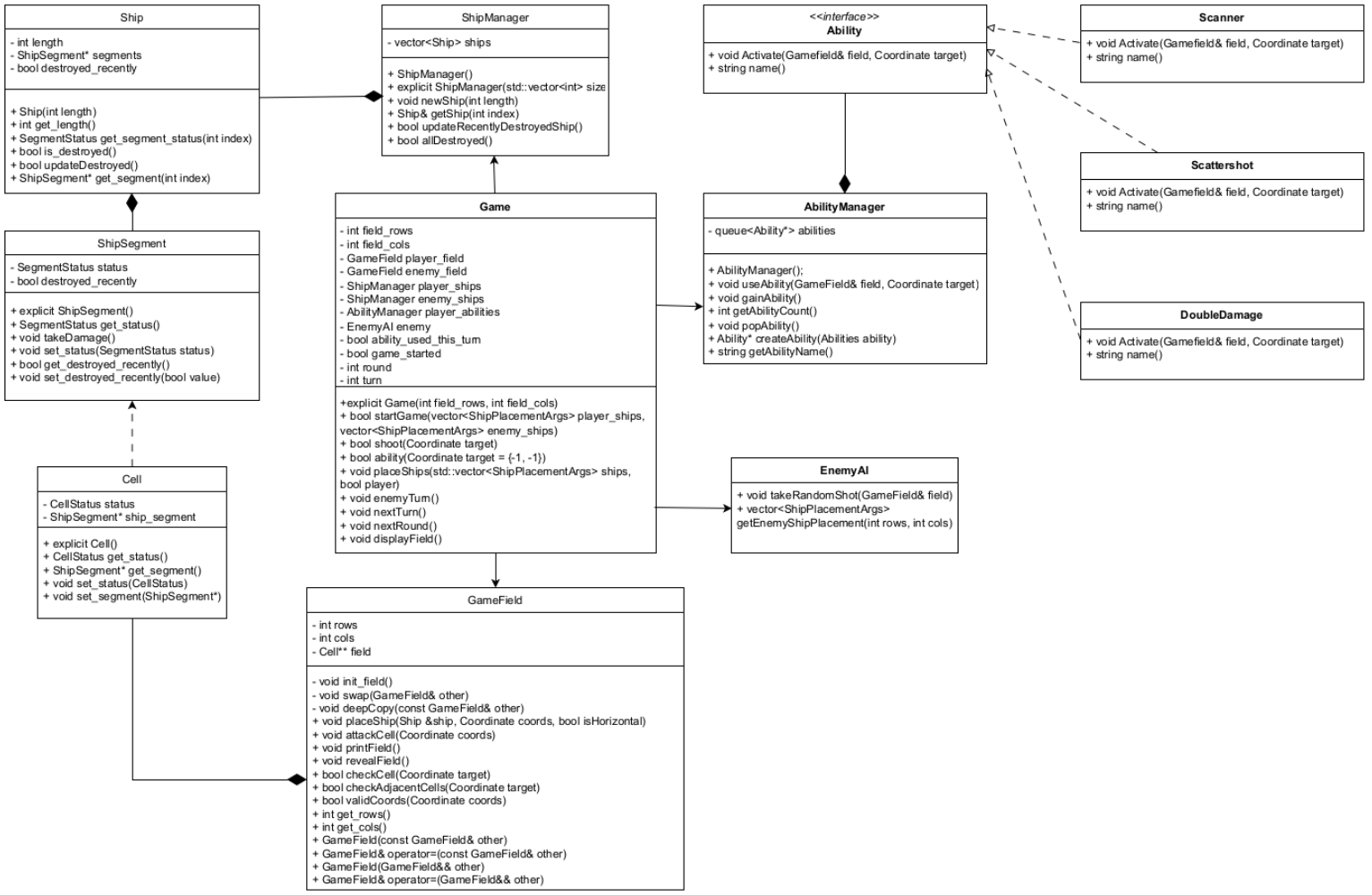

Рисунок 1 – UML-диаграмма классов

Код программы содержит реализацию классов: Ability, DoubleAttack, Scanner, Scattershot, AbilityManager.

В код добавлены следующие классы-исключения:

- Атака за пределы поля
- Попытка применить способность, когда её их нет
- Размещение корабля на уже существующем корабле или вплотную, возле него.

Ability - класс-интерфей для способностей игрока. Он имеет виртуальные методы:

- virtual void Activate(Gamefield& field, Coordinate target) – виртуальный метод активации способности.

- virtual string name() - возврат имени способности в виде строки.

Класс DoubleDamage является реализацией способности двойного урона.

- void Activate(Gamefield& field, Coordinate target) override – производит двойную атаку по заданной клетке поля.

Класс Scanner является реализацией способности сканера.

- void Activate(Gamefield& field, Coordinate target) override – пишет информацию о наличии вражеских кораблей в области 2x2 (данная координата — верхний левый угол области).

Класс Scattershot является реализацией способности случайного выстрела.

- void Activate(Gamefield& field, Coordinate _target) override – ищет случайный сегмент корабля на поле и выполняет выстрел.

Класс AbilityManager контроллирует способностями, хранит в очереди объекты способностей.

- queue<Ability*> abilities – очередь указателей на способности.
- AbilityManager() – конструктор класса.
- int getAbilityCount() – возвращает размер очереди.
- String getAbilityName() – возвращает название текущей способности.
- Ability* createAbility(Abilities ability) – создает новую способность для добавления в очередь.
- void gainAbility() – добавляет случайную способность в очередь.
- void useAbility(GameField& field, Coordinate target = {-1,-1}) – применяет способность из очереди. После выполнения способность убирается из очереди.
- void popAbility() – удаляет способность из очереди.

Выводы

Во время выполнения лабораторной работы были изучены основы полиморфизма классов в ООП C++, а также методы обработки ошибок. Реализован интерфейс способностей, три различные способности, менеджер способностей, классы-исключения для определенных ситуаций.