

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «ООП»

Тема: Интерфейсы классов; взаимодействие классов; перегрузка операций

Студент(ка) гр. 0000

Ивченко А.А.

Преподаватель

Размочасева Н.В.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор классов:

- Класс базы
- Набор классов ландшафта карты
- Набор классов нейтральных объектов поля

Класс базы должен отвечать за создание юнитов, а также учитывать юнитов, относящихся к текущей базе. Основные требования к классу база:

- База должна размещаться на поле
- Методы для создания юнитов
- Учет юнитов, и реакция на их уничтожение и создание
- База должна обладать характеристиками такими, как здоровье, максимальное количество юнитов, которые могут быть одновременно созданы на базе, и.т.д.

Набор классов ландшафта определяют вид поля. Основные требования к классам ландшафта:

Должно быть создано минимум 3 типа ландшафта

- Все классы ландшафта должны иметь как минимум один интерфейс
- Ландшафт должен влиять на юнитов (например, возможно пройти по клетке с определенным ландшафтом или запрет для атаки определенного типа юнитов)
- На каждой клетке поля должен быть определенный тип ландшафта

Набор классов нейтральных объектов представляют объекты, располагаемые на поле и с которыми могут взаимодействовать юниты. Основные требования к классам нейтральных объектов поля:

- Создано не менее 4 типов нейтральных объектов
- Взаимодействие юнитов с нейтральными объектами, должно быть реализовано в виде перегрузки операций
- Классы нейтральных объектов должны иметь как минимум один общий интерфейс

Описание классов.

В ходе работы был разработан класс объекта на клетке Object, который имеет следующие поля:

`char` obj_name — имя объекта

`Unit*` unit — ссылка на юнит, в данный момент находящийся на клетке(может быть nullptr)

`Neutral*` ntr — ссылка на нейтральный объект(может быть nullptr);

`bool` availability — доступ к этой клетке для юнита

Также класс Object реализует методы:

`void` SetUnit(`Unit` *a) — устанавливает юнит

`void` RemoveUnit() - удаляет юнит

`void` SetObj(`char` a) — устанавливает объект

`Unit*` getUnit() - возвращает юнит

`char` getObj() -возвращает объект

`Neutral` *getNeutral() -возвращает ссылка на нейтр. объект

`void` setNeutral(`Neutral` *a) — устанавливает нейтр. объект

`void` SetAvailability(`bool` b) — устанавливает доступ

`bool` IsAvailable() - проверяет доступ

Класс Object является родительским классом для классов base, landscape

Также был создан класс нейтральных объектов Neutral с полями:

`char` neutral_name — имя нейтр объекта

`Context` context — контекст для выбора стратегии

и реализующий методы:

`void` setType(`char` a) — устанавливает тип нейтр. объекта

`char` getType() - возвращает тип нейтр. объекта

`void` use(`Unit` *a) — применить нейтральные объект к юниту

Вывод.

В результате лабораторной работы были получены знания по наследованию и взаимодействию классов, перегрузке операторов. Был созданы необходимые классы и реализован их интерфейс, согласно требованиям в задании.