

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Объектно-ориентированное программирование»
Тема: Интерфейсы классов; взаимодействие классов; перегрузка
операций

Студентка гр. 8381

Преподаватель

Звегинцева Е.Н.

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать класс базы, набор классов ландшафта, набор классов нейтральных объектов поля.

Задание.

Основные требования.

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.

- Для хранения информации о юнитах в классе базы используется паттерн “Компоновщик” / Использование “Легковеса” для хранения общих характеристик юнитов.
 - Для наблюдения над юнитами в классе база используется паттерн “Наблюдатель”.
 - Для взаимодействия ландшафта с юнитами используется паттерн “Прокси”.
- Для взаимодействия одного типа нейтрального объекта с разными типами юнитов используется паттерн “Стратегия”.

Выполнение работы.

Написание работы производилось на базе операционной системы Windows10 в среде разработки Qt Creator.

В данной лабораторной работе были реализованы следующие классы:

- Base – класс, отвечающий за создание, удаление и нанесение урона юнитам. Содержит в себе счетчик(учет юнитов) и имеющий собственное «здоровье»
- CompositeUnit – наследник от Unit. Класс, отвечающий за создание комплекса юнитов из разных видов. Реализована работа, схожая с интерфейсом Unit
- Observer+Subject – классы подписчика и изделия, реализующие паттерн Наблюдатель. В случае смерти юнита, они передают эту информацию базе и происходит удаление с поля. Таким образом, класс Unit, наследуемый от класса Subject, сообщает классу PlayinField, о изменениях в юните, который в свою очередь наследуется от класса Observer и реализует метод реакции на смерть юнита – удаление его с поля.
- Proxy – класс паттерна Прокси. Работает как заместитель, для того, чтобы проверить информацию по ландшафту и корректность нахождения на ней юнита.
- Context – позволяет на этапе выполнения программы выбирать в зависимости от контекста необходимы вид нейтрального объекта для взаимодействия с юнитом. Также в данном классе осуществляется перегрузка бинарного оператора += посредством которого юнит терпит изменения после

получения того или иного объекта.

- NeutralObj – класс, реализующий нейтральные объекты поля и их взаимодействие с юнитами.

Выводы.

В ходе выполнения лабораторной работы были написаны требуемые классы поля с сопутствующими классами для хранения данных, а также абстрактный класс юнита с соответствующими производными классами.