

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

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

Студент гр. 8303

Данилов А.В.

Преподаватель

Филатов А.Ю.

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

Дополнительные требования

- 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 Ubuntu 20.04 в среде разработки Clion.

Реализованные классы.

Классы, добавленные в программу в данной лабораторной работе и их функционал представлены в табл. 1. В ней приведено общее описание классов, раскрывающее мотивацию их создания.

Таблица 1 – Основные добавленные классы

Класс	Назначение
Base	<p>Местоположение: ./Base/</p> <p>Класс отвечает за создание и удаление юнитов, их учёт и передачу урона определённому юниту.</p> <p>Согласно требованиям данный класс также обладает собственным здоровьем и не обладает указателем на игровое поле.</p>
Composite (реализует паттерн Компоновщик)	<p>Местоположение: ./Base</p> <p>Класс отвечает со создание комплексных подразделений из разных видов юнитов, в силу наследования от класса Unit, работа с ним осуществляется на уровне интерфейса обычного юнита.</p>
Observer + Subject (реализуют паттерн Наблюдатель)	<p>Observer – класс, реализующий интерфейс подписчика, а Subject – абстрактный класс издателя.</p> <p>Таким образом, класс Unit, реализованный в 1-ой лабораторной работе наследуется от класса Subject и сообщает игровому полю об изменениях своего состояния. Класс Field в свою очередь наследуется от класса Observer и реализует метод реакции на смерть юнита – удаление его с поля.</p>
Proxy (реализует паттерн Прокси)	<p>Местоположение: ./Landscape</p> <p>Данный класс используется в виде заместителя при обращении к элементу ландшафта с целью проверки доступа и его логгирования.</p>
Context (реализует паттерн Стратегия)	<p>Местоположение: ./NeutralObject</p> <p>Позволяет на этапе выполнения программы выбирать в зависимости от контекста необходимы вид нейтрального объекта для взаимодействия с юнитом.</p> <p>Также в данном классе осуществляется перегрузка бинарного оператора += посредством которого юнит терпит изменения после получения того или иного объекта.</p>

Выводы.

В ходе лабораторной работы были разработаны и реализованы набор классов для создания ландшафта, нейтральных объектов и базы.