

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Объектно-ориентированное программирование»
Тема: Логическое разделение классов

Студент гр. 8304

Бутко А.М.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор классов для взаимодействия пользователя с юнитами и базой. Реализовать функционал управления юнитами и базой.

Выполнение работы

Требование к работе	Выполнение требования	
Должен быть реализован функционал управления юнитами	+	Юнит можно создать на базе (ф-ция createUnit()), выставить на игровом поле (ф-ция getUnit()), передвигать по полю (ф-ция moveObject()), атаковать противника (базу или юнита) (ф-ция makeDamage()).
Должен быть реализован функционал управления базой	+	Реализован функционал создания базы createBase(), получения состояния базы getInformation(), взятия юнита с базы (для последующего размещения на поле), функция атаки базой противника makeDamage().
Выполнены все основные требования к взаимодействию	+	Описаны выше
Добавлен функционал просмотра состояния базы	+	Функция getInformation(), которая выводит в консоль текущее здоровье базы, урон базы, количество созданных юнитов (общее количество и отдельно по каждому), так же информацию о юнитах на игровом поле и об умерших юнитах на игровом поле.
Имеются 3+ демонстрационных примера.	+	В функции exes3() можно проверить функциональность программы.
<i>* Реализован паттерн «Фасад» через который пользователь управляет программой</i>	+	Пользователь управляет программой с помощью класса Facade. Класс имеет два метода — createGame() для создания и инициализации игрового поля и цепочки обязанностей, playGame() для ввода команд пользователя. Программа заканчивается командой '/quit'.
<i>* Объекты между собой взаимодействуют через паттерн «Посредник»</i>	+	Посредником между объектами выступают команды паттерна «Команда»

<i>* Для передачи команд используется паттерн «Команда»</i>	+	Паттерн реализован с помощью класса Command. Команды содержатся в папке Commands.
<i>* Для приема команд от пользователя используется паттерн «Цепочка обязанностей»</i>	+	Паттерн реализован с помощью класса Chain(). Создаются звенья цепи, которые выполняют команду пользователя или передают ее, если не смогли обработать.

Выводы.

Были реализованы основные требования к данной лабораторной работе, а так же изучены паттерны «Фасад», «Посредник», «Команда», «Цепочка обязанностей».