

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

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

Студент гр. 8383

Мололкин К.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Продолжить работу над игрой, изучить паттерны проектирования, необходимые для логического разделения классов.

Постановка задачи.

Разработать и реализовать набора классов для взаимодействия пользователя с юнитами и базой. Основные требования:

- Должен быть реализован функционал управления юнитами
- Должен быть реализован функционал управления базой

Выполнение работы.

На протяжении двух предыдущих лабораторных было уже реализован некоторый функционал, необходимый для данной лабораторной работы, поэтому необходимо разделить классы, которые реализуют интерфейс работы, и классы, непосредственно выполняющие те или иные функции.

Управление юнитами и базой

Class Game реализуют паттерн «Фасад», где пользователь управляет программой.

Поля:

1. private GameBoard* board – указатель на поле;

Методы:

1. public Game() – конструктор класса;
2. public void startGame() – в данном методе происходит запуск игры, то есть запускается цикл считывания команд пользователя.

Список команд пользователя:

1. Перезагрузка игры
2. Вывод игрового поля
3. Передвижение юнита
4. Вывод информации о базе

5. Создание юнита на базе
6. Вывод информации о юните
7. Атака юнита
8. Завершение игры

class Command

Интерфейс команд, реализует паттерн «Команда».

Поля наследников:

1. GameBoard** board – указатель на указатель на поле(указатель на указатель используется для того чтобы исполнить команду перезапуска поля).

Методы:

1. bool execute() – метод исполнения команды, если команда не выполнена, то возвращается false.

Классы наследники:

1. UnitAttackCommand (исполняет команду атаки юнитом);
2. RestartCommand (исполняет команду перезагрузки игры);
3. PrintBoardCommand (исполняет команду печати поля);
4. MoveUnitCommand (исполняет команду передвижения юнита);
5. AboutBaseCommand (исполняет команду вывода информации о базе);
6. CreateUnitFromBaseCommand (исполняет команду генерации юнита на базе);
7. AboutUnitCommand (исполняет команду вывода информации о юните)

class Mediator

Реализует паттерн «Посредник».

Используется для исполнения команды атаки.

Методы:

1. attackBase(int attackPower, Base* base) – атака базы;
2. attackUnit(int attackPower, IUnit* unit) – атака юнита;

Все созданные классы, и уже имеющиеся, реализуют паттерн «Цепочка обязанностей». Для создания юнита класс `CreateUnitFromBaseCommand` обращается к методу `createUnit` класса `GameBoard`, из поля в свою очередь обращается к базе, база к фабрике юнитов, затем созданный юнит ставится на клетку. Класс `AboutUnitCommand` обращаются к полю, а поле к клетке, клетка к юниту для вывода информации о нем. Класс `AboutBaseCommand` обращается к полю, а поле к базе. `MoveUnitCommand` так же обращается к полю, а поле к клетке. Таким образом реализуется данный паттерн

Демонстрация работы.

1. При запуске игры считываются размеры поля, а затем оно создается и выводится список команд простейший интерфейс с командами.

```
Game starts!
Enter board length and width
12
12
1 - Restart game
2 - Print board
3 - Move unit
4 - Print information about base
5 - Create unit from base
6 - Print information about unit
7 - Attack unit or base
All other numbs and symbols - Stop game
```

2. Вызовем команду печати поля

```

2
0 | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ HI | _ HI | _ PL | _ PL | _ PL | _ PL |
1 | _ PL | _ HI | _ PL | _ PL | _ PL | _ HI | _ SW | _ HI | _ HI | _ SW | _ PL | _ PL |
2 | _ PL | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ SW | _ PL | _ HI | _ PL | _ PL |
3 | _ PL | _ PL | _ PL | _ HI | _ PL | _ HI | _ PL | _ HI | _ HI | _ SW | _ PL | _ PL |
4 | _ HI | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI |
5 | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ HI | _ SW | _ PL |
6 | B1 HI | _ PL | _ HI | _ PL | _ PL | _ HI | _ HI | _ PL | _ HI | _ PL | _ PL | B2 PL |
7 | _ PL | _ HI | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ PL | _ HI | _ HI | _ HI |
8 | _ PL | _ HI | _ HI | _ PL | _ HI | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL |
9 | _ PL | _ HI | _ PL | _ HI | _ HI | _ SW | _ PL | _ PL | _ PL | _ PL | _ PL | _ HI |
10| _ PL | _ HI | _ PL | _ PL | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI | _ HI |
11| _ PL | _ HI | _ HI | _ PL | _ PL | _ SW | _ HI | _ PL | _ PL | _ PL | _ PL | _ SW |
    0      1      2      3      4      5      6      7      8      9     10     11
-----
```

3. Создадим лучника и выведем поле:

```
5
Enter base number 1 or 2
1
Choose unit type
1 - Spearman
2 - Swordsman
3 - Archer
4 - Ballista
5 - Crossbowman
6 - Catapult
3
Enter unit position
7 6
The command was executed without problems
1 - Restart game
2 - Print board
3 - Move unit
4 - Print information about base
5 - Create unit from base
6 - Print information about unit
7 - Attack unit or base
All other numbs and symbols - Stop game
2
0 | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ HI | _ HI | _ PL | _ PL | _ PL | _ PL |
1 | _ PL | _ HI | _ PL | _ PL | _ PL | _ HI | _ SW | _ HI | _ HI | _ SW | _ PL | _ PL |
2 | _ PL | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ SW | _ PL | _ HI | _ PL | _ PL |
3 | _ PL | _ PL | _ PL | _ HI | _ PL | _ HI | _ PL | _ HI | _ SW | _ HI | _ PL | _ PL |
4 | _ HI | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI |
5 | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ HI | _ SW | _ PL |
6 | B1 HI | _ PL | _ HI | _ PL | _ PL | _ HI | _ HI | X1 PL | _ HI | _ PL | _ PL | B2 PL |
7 | _ PL | _ HI | _ PL | _ PL | _ PL | _ PL | _ PL | _ SW | _ PL | _ HI | _ HI | _ HI |
8 | _ PL | _ HI | _ HI | _ PL | _ HI | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL | _ PL |
9 | _ PL | _ HI | _ PL | _ HI | _ HI | _ SW | _ PL | _ PL | _ PL | _ PL | _ PL | _ HI |
10 | _ PL | _ HI | _ PL | _ PL | _ PL | _ HI | _ PL | _ PL | _ HI | _ PL | _ HI | _ HI |
11 | _ PL | _ PL | _ HI | _ PL | _ PL | _ SW | _ HI | _ PL | _ PL | _ PL | _ PL | _ SW |
    0      1      2      3      4      5      6      7      8      9      10     11
```

4. Вызовем команду атаки юнита на базу и команду вывода свойств базы.

```
The command wasn't executed
1 - Restart game
2 - Print board
3 - Move unit
4 - Print information about base
5 - Create unit from base
6 - Print information about unit
7 - Attack unit or base
All other numbs and symbols - Stop game
7
Enter coordinates attacker and attacked 7 6
11
6
The command was executed without problems
1 - Restart game
2 - Print board
3 - Move unit
4 - Print information about base
5 - Create unit from base
6 - Print information about unit
7 - Attack unit or base
All other numbs and symbols - Stop game
4
Enter base number 1 or 2
2
Base information:
Team: 2
Posittion: (6,11)
Health: 780
Number units from base: 0
The command was executed without problems
```

5. Закончим игру.

```
1 - Restart game
2 - Print board
3 - Move unit
4 - Print information about base
5 - Create unit from base
6 - Print information about unit
7 - Attack unit or base
All other numbs and symbols - Stop game
9
Game was finished
```

Выводы.

В ходе выполнения данной лабораторной работы были изучены паттерны, необходимые для логического разделения классов.