

СМИНОБРНАУКИ 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

Костарев К.В.

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.

Для этого из проекта был полностью удален класс `GameBoardStd` и соответственно файлы `GameBoardStd.cpp`, `GameBoardStd.h`, удалено множество функций из класса `Board` и созданы новые классы `Game`, `Features` и `Command`, реализованные в `Game.h` и `Game.cpp`.

class Game, class Features

Реализуют паттерн «Фасад», где пользователь управляет программой.

`Features`: Метод `void printFeatures()` печатает на экране те команды, которые может совершить пользователь, метод `Commands getCommand()` возвращает обозначение команды, которую пользователь ввел для ее последующего выполнения. Все наименования команд представлены в `enum Commands` в файле `Naming.h`. Поля класса:

- `bool board`, `cityHall`, `unit` – булевы флаги, показывающие, есть ли в игре хотя бы один представитель того или иного объекта, необходимо для составления списка возможных команд.

Game: инициализирует команды интерфейса Command, введенные пользователем с помощью вспомогательного фасада Features. Полями класса являются:

- игровой мир board,
- интерфейс команд command,
- название команды, которую надо проинициализировать, nameCommand,
- булевый флаг, показывающий состояние игры, gameOn,
- объект вспомогательного фасада Features.

class Command

Интерфейс команд, реализует паттерн «Команда».

Наследниками являются команды NewBoard, NewCityHall, NewBuild, DelBuild, AboutBoard, AboutBuild, AboutCityHall, AboutTreasury.

Содержит функционал управления юнитами и базой.

class Mediator

Реализует паттерн «Посредник», описан в Mediator.h, Mediator.cpp.

Так, теперь некоторые отдельные здания действуют друг на друга: если рядом с жилым зданием построить электростанцию или водоканал, то это негативно повлияет на жилое здание (доход не более 10 и число жильцов не более 2).

Все созданные классы, и уже имеющиеся, реализуют паттерн «Цепочка обязанностей», на примере: чтобы создать базу, Game обращается к функции execute() класса NewCityHall, на экране отображается интерфейс команды, функция execute() собирает необходимые данные и обращается к функции addCityHall() класса Board, та изменяет параметры Board и обращается к функции с необходимыми для нее параметрами addCityHall(int w, int p) класса Cell, которая уже ставит на клетку базу.

Демонстрация работы.

1.

При запуске игры теперь существует простейший интерфейс с командами.

```
You wan't start a game? (1 - yes, 0 - no)      1
You want see a command panel?    1
You can:
'0' - end game
'10' - create a board
Your command:
```

Создадим игровое поле. Видно, что после создания поля доступен больший объем команд, теперь можно посмотреть информацию о поле или создать базу.

```
Your command:  10
Enter width and height of gameboard (5-49):    6 6
Well
You want see a command panel?    1
You can:
'0' - end game
'11' - get info about board
'12' - get info about treasury
'20' - create first City Hall
Your command:  _
```

Попросим вывести информацию.

```
Your command:  11
Size: 6x6
^ ^  _ _  ~ ~
_ _  ~ ~  _ ~
~ ^  _ ^  ~ ^
_ _  _   ~ ~ ^
_ _  _   ~ ~
^ _  ~ ^  ^ ^
Coins: 2000
Objects: 0
Well
```

Создадим теперь базу и заодно хрущевку.

```

Your command: 20
Enter x, y coordinates for building City Hall: 2 1
Well
You want see a command panel? 1
You can:
'0' - end game
'11' - get info about board
'12' - get info about treasury
'20' - create new City Hall
'21' - get info about City Hall
'30' - create Building
Your command: 30
Select a base (City Hall), enter x and y: 2 1
Enter x, y coordinates for building: 1 2
Apartments: 0 - Dacha, 1 - Hrushchovka
Production: 2 - Ferma, 3 - Office
Services: 4 - Powerhouse, 5 - School, 6 - Vodokanal
Enter type of build: 1
Well
You want see a command panel? 0
Your command: 11
Size: 6x6
^ ^ _ _ ~ ~
_ _ ~ ~ _ ~
~ ^ _ ^ ~ ^
_ _ _ ~ ~ ^
H _ _ _ ~ ~
^ C ~ ^ ^ ^
Coins: 2000
Objects: 2
Well

```

Посмотрим информацию о базе и сравним с информацией о здании. Увы, оказалось, что мы построили хрущевку на месте радиации.

```

You want see a command panel?  0
Your command:  21
Enter x, y coordinates of City Hall:  2 1
CITY HALL.
Sum Saldo: -100
Sum Employment: 0
Sum Education: 0
Sum Energy: 0
Sum Water: 0
Sum Satiety: 0
Well
You want see a command panel?  0
Your command:  31
Enter x, y coordinates of build:  1 2
Saldo: -100
Energy: 0
Water: 0
Eat: 0
Workmans: 0
Students: 0
Kids: 0
Well

```

Выйдем из игры.

```

You want see a command panel?  1
You can:
'0' - end game
'11' - get info about board
'12' - get info about treasury
'20' - create new City Hall
'21' - get info about City Hall
'30' - create Building
'31' - get info about building
'32' - blow up a building
Your command:  0
Ok, goodbye!

```

2.

Покажем, как объекты влияют друг на друга.

Создадим хрущевку, убедимся, что с ней все в порядке.

```

Size: 10x10
^ ~ ~ ^ _ ~ _ ~ ^ _
_ ~ ^ ^ _ ~ ~ ~ ~ ^
^ ^ ^ ^ ^ ^ ~ ~ _ ~
^ ~ ~ _ ~ _ _ ~ ^ ^
~ ~ ~ ^ _ _ ~ ^ _ _
_ ~ ^ ^ ~ ^ ^ ~ ~ ~
^ ~ ~ ~ _ ~ ^ _ _ _
^ ~ _ ~ _ ~ _ ~ _ ~
_ ^ _ ^ ^ ^ _ ~ ~ ^
^ C ^ ^ ^ ~ ~ ^ ~ ~
Coins: 2000
Objects: 1
Well

```

```

Size: 10x10
^ ~ ~ ^ _ ~ _ ~ ^ _
_ ~ ^ ^ _ ~ ~ ~ ~ ^
^ ^ ^ ^ ^ ^ ~ ~ _ ~
^ ~ ~ _ ~ _ _ ~ ^ ^
~ ~ ~ ^ _ _ ~ ^ _ _
_ ~ ^ ^ ~ ^ ^ ~ ~ ~
^ ~ ~ ~ _ ~ ^ _ _ _
^ ~ _ ~ _ ~ _ ~ _ ~
_ ^ _ ^ ^ ^ H ~ ~ ^
^ C ^ ^ ^ ~ ~ ^ ~ ~
Coins: 2000
Objects: 2
Well
You want see a command panel?  0
Your command:  31
Enter x, y coordinates of build:      7 2
Saldo: 318
Energy: -530
Water: -424
Eat: -318
Workmans: 69
Students: 24
Kids: 13
Well

```

Построим теперь рядом электростанцию и посмотрим характеристики хрущевки.

```

Enter x, y coordinates of build:      7 2
Saldo: 8
Energy: -530
Water: -424
Eat: -318
Workmans: 1
Students: 24
Kids: 13
Well
You want see a command panel?    0
Your command:    11
Size: 10x10
^ ~ ~ ^ _ ~ _ ~ ^ _
_ ~ ^ ^ _ ~ ~ ~ ~ ^
^ ^ ^ ^ ^ ^ ~ ~ _ ~
^ ~ ~ _ ~ _ ~ ^ ^
~ ~ ~ ^ _ ~ ^ _ _
_ ~ ^ ^ ~ ^ ^ ~ ~ ~
^ ~ ~ ~ _ ~ ^ _ _ _
^ ~ _ ~ _ ~ _ ~ ~
_ ^ _ ^ ^ P H ~ ~ ^
^ C ^ ^ ^ ~ ~ ^ ~ ~

```

Сальдо и количество рабочих действительно уменьшились, электростанция плохо повлияла на хрущевку.

3.

При неправильном вводе данных игра просит ввести команды заново:

```

You want see a command panel?    0
Your command:    20
Enter x, y coordinates for building City Hall:  654 665
THE COMMAND CAN'T BE EXECUTED
You want see a command panel?

```

Выводы.

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