

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. 8381

Преподаватель

Муковский Д.В.

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы.

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.

Все реализованные в данной работе классы и их назначение представлены в табл.1.

Таблица 1 – Основные добавленные классы

<i>Command, MoveCommand, AttackCommand, HealCommand, AddUnitCommand, StartNewGameCommand</i>	Данные классы реализуют паттерн «Команда». В абстрактном классе <i>Command</i> объявлен виртуальный метод <i>execute()</i> , который переопределен у каждой конкретной команды, данный метод перенаправляет вызов получателю классу <i>GameFacade</i> .
<i>Handler, AbstractHandler, StartGame, Attack, Heal, AddUnit, MoveUnit, GetInfo</i>	Данные классы реализуют паттерн «Цепочка обязанностей». Обработчик <i>Handler</i> определяет: методы установки следующего конкретного обработчика, а также метод <i>handle</i> , который обрабатывает команды пользователя и перенаправляет их. В абстрактном обработчике команда <i>handle</i> переопределена таким образом, чтобы передавать работу следующему конкретному обработчику.
<i>ClientInput</i>	Вспомогательный класс для представления ввода пользователя.

<i>GameFacade</i>	Данный класс реализует паттерн «Фасад». Он представляет собой упрощенный интерфейс к классам игры, во многом ограниченный, чтобы облегчить работу пользователю.
<i>Game</i>	Дополнительный фасад, чтобы код был проще и читабельнее. В нем хранятся поле и базы игры и он напрямую оперирует с их методами.
<i>UnitInteractionMediator</i>	Класс, реализующий паттерн «Посредника». Сам является посредником, который реализует взаимодействие юнитов между собой.

Также в программе был реализован функционал просмотра базы, а именно: какому игроку она принадлежит, количество здоровья, количество созданных и живых на данный момент юнитов, а также максимальное количество юнитов, которых можно создать.

Также в *main.cpp* были написаны три примера, один из которых является бесконечным циклом, в котором считываются команды пользователя с консоли, а в случае уничтожения одной из двух баз цикл обрывается. Так что формально это можно назвать игрой.

Пример добавления юнита представлен на рис.1.

```

1 Player turn:
Enter command:
add
Enter coords:
3 4
Enter unit type:
w_t
Enter unit color:
blue
FlyweightFactory: Reusing existing flyweight.
Field:Object added X:3 Y:4
Base: Unit added correctly
Field:
[ _B_ ] [ ] [ ] [ ] [ h_b ] [ ] [ ] [ w_t ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ w_t ] [ ] [ ] [ ] [ a_d ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ a_b ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ w_t ] [ ] [ ]
[ ] [ ] [ ] [ ] [ h_b ] [ ] [ ] [ ] [ ] [ _B_ ]

Landscape:
[ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _M_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]

2 Player turn:
Enter command:

```

Рисунок 1 – Добавление юнита на поле

Пример передвижения юнита представлен на рис.2.

```

1 Player turn:
Enter command:
move
Enter first coords:
3 4
Enter second coords:
5 5
Field:Unit mooved from X:3 Y:4 to X:5 Y:5
Field:
[ _B_ ] [ ] [ ] [ ] [ h_b ] [ ] [ ] [ w_t ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ a_d ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ w_t ] [ ] [ ] [ ]
[ ] [ a_b ] [ ] [ ] [ ] [ ] [ ] [ w_t ] [ ] [ ]
[ ] [ ] [ ] [ ] [ h_b ] [ ] [ ] [ ] [ ] [ _B_ ]

Landscape:
[ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _M_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _M_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _M_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _W_ ] [ _W_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]
[ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ] [ _G_ ]

```

Рисунок 2 – Передвижение юнита

