

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

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

Студент гр. 8381

Преподаватель

Муковский Д.В.

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы.

Написание игры жанра стратегия в объектно-ориентированном стиле.
Реализация классов базы, ландшафта и нейтральных объектов.

Постановка задачи.

Разработать и реализовать набор классов:

- Класс базы
- Набор классов ландшафта карты
- Набор классов нейтральных объектов поля

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 взаимодействие юнитов. 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

Выполнены основные требования к классу база	2 балла
Выполнены основные требования к набору классов ландшафта	2 балла
Выполнены основные требования к набору классов нейтр. объектов	2 балла
Добавлено взаимодействие юнитов	1 балла
Имеется 3+ демонстрационных примера	1 балл
Взаимодействие через перегрузку операторов	2 балла
<i>*Для хранения информации о юнитах в классе базы используется паттерн “Компоновщик”/ Использование “Легковеса” для хранения общих характеристик юнитов</i>	2 балла
<i>*Для наблюдения над юнитами в классе база используется паттерн “Наблюдатель”</i>	2 балла
<i>*Для взаимодействия ландшафта с юнитам используется паттерн “Прокси”</i>	3 балла
<i>*Для взаимодействия одного типа нейтрального объекта с разными типами юнитов используется паттерн “Стратегия”</i>	3 балла
Кол-во баллов за основные требования	10 баллов
Максимальное кол-во баллов за лаб. работу	20 баллов

Выполнение работы.

Был реализован класс База, объекты которого размещаются на поле. Данный класс имеет метод создания юнитов *createUnit()*, который по заданному типу юнита и координате создает юнита и отправляет его на поле.

Так же при хранении юнитов используется паттерн “Легковес”, который хранит внутреннее состояние объекта, в моем случае это цвет юнита. При вызове конструктора базы создается фабрика легковесов, в которой уже заранее созданы и занесены несколько легковесов. При создании юнита его цвет передается через параметры и с помощью метода *getFlyweight()* либо возвращается уже созданный легковес, либо в случае, если его не существовало, создается новый. После чего конкретный юнит связывается с легковесом.

Также для наблюдения за юнитами используется паттерн “Наблюдатель”. В случае уничтожения юнита, вызывается метод, который оповещает базу об этом, а она в свою очередь удаляет конкретного юнита.

Также база обладает здоровьем и учитывает количество юнитов. При создании базы устанавливается лимит юнитов и при создании нового счетчик юнитов увеличивается.

Далее были реализованы ландшафты, а именно ландшафты земли, воды и гор. Все ландшафты хранятся в поле отдельным массивом. Взаимодействие юнитов с ландшафтами происходит с помощью паттерна “Заместитель”, объект заместителя, как и конкретный объект ландшафта наследуется от одного интерфейса и в случае взаимодействия юнита с ландшафтом, объект заместитель передает работу конкретному ландшафту.

Также были реализованы нейтральные объекты, а именно аптечки и другие бусты. Взаимодействие одного типа нейтрального объекта с разными типами юнитов использует паттерн “Стратегия”. У каждого нейтрального объекта есть две стратегии взаимодействия с двумя типами юнитов, а именно с юнитами танками и юнитами дамагерами. Например юнитам танкам восстанавливается больше здоровья, чем юнитам дамагерам.

Тестирование программы.

В первом тесте была продемонстрирована работа легковеса. Что при создании юнита с уже существующим для него легковесом, новый легковес не создается. Результат представлен на рис.1.

```
Create unit with existing flyweight
FlyweightFactory: Reusing existing flyweight.
Field:Object added X:1 Y:1
Base: Unit added correctly
Field 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][w_t][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][B_ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Create unit with new flyweight
FlyweightFactory: Can't find a flyweight, creating new one.
Field:Object added X:2 Y:2
Base: Unit added correctly
Field 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][w_t][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][B_ ][B_d][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Create unit with previous flyweight
FlyweightFactory: Reusing existing flyweight.
Field:Object added X:2 Y:5
Base: Unit added correctly
Field 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
```

Рисунок 1 – Пример работы программы №1

Во втором примере была продемонстрировано взаимодействие нейтральных объектов с конкретными типами юнитов через перегрузку оператора «+=». Также было показано, что база контролирует максимальное количество созданных юнитов. Результат представлен на рис.2.


```

Base: Unit added correctly
Field 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Landscape 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Interaction of the unit with the landscape
Unit at X:3 Y:6
Unit can't swim
Field:Unit deleted X:3 Y:6
Base: Unit deleted
Field 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Landscape 1:
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

```

Рисунок 3 – Пример работы программы №3

Выводы.

В ходе выполнения лабораторной работы были изучены и применены такие паттерны проектирования, как «Стратегия», «Наблюдатель», «Заместитель» и «Легковес». Также была создана UML диаграмма для данной программы.