

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

Сосновский Д.Н.

Жангиров Т. Р.

Санкт-Петербург

2020

Задание.

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

Выполнение работы

Для выполнения лабораторной работы были созданы следующие классы:

Таблица 1 – Основные добавленные классы

Имя класса	Назначение
Base (класс базы)	<p>Класс базы расы. Размещается на игровом поле в месте, зависящем от расы – т.к. расы 2, то одна база в левом верхнем углу, другая – в правом нижнем.</p> <p>Все юниты на поле создаются через базу своей расы. База является наблюдателем за своими юнитами (доп. требование).</p> <p>Базу можно уничтожить (что и является целью игры). Атаковать база не может.</p>
IUnitObserver, IUnitObservableByBase	Реализация паттерна « Наблюдатель » для базы, которая наблюдает за юнитами.
INeutralObject	Абстрактный класс, общий для всех нейтральных объектов
ICreatureObserver, ICreatureObservableByFieldComponent	<p>Реализация паттерна «Наблюдатель» для клетки и сущности, которая в ней находится.</p> <p>Относится к доп.баллам в конце всех лабораторных.</p>
ILandscape	Общий интерфейс для всех видов ландшафта.

LandscapeProxy	Класс, реализующий паттерн « Прокси » для взаимодействия юнита с ландшафтом.
Mountains, Ground, Water	Конкретные классы ландшафта. Определяют, может ли юнит располагаться на них. В Mountains и Ground – да, в Water – нет. По умолчанию везде генерируется Ground.
Banner, Tower, Temple, Forge	Конкретные классы нейтральных объектов. Каждый нейтральный объект имеет свою стратегию (см. далее) по взаимодействию с каждым типом юнитов
INeutralObjectsEffectStrategy	Интерфейс для всех стратегий взаимодействия юнитов с нейтральными объектами.
BannerInfantryStrategy, BannerCavalryStrategy, BannerArcherStrategy, TowerInfantryStrategy, TowerCavalryStrategy, TowerArcherStrategy, TempleInfantryStrategy, TempleCavalryStrategy, TempleArcherStrategy, ForgeInfantryStrategy, ForgeCavalryStrategy, ForgeArcherStrategy	Классы, реализующие паттерн « Стратегия » для взаимодействия юнитов с нейтральными объектами. С каждым типом юнитов каждый нейтральный объект имеет свою стратегию взаимодействия.

Создание UML-диграммы.

UML-диаграмма была создана при помощи сервиса Lucidchart. Диаграмма идёт в приложении к лабораторной работе.

Тестирование программы.

Для демонстрации работы программы были разработаны 3 примера.

Пример 1.

Цель примера – проверить работоспособность создания баз на поле и добавление юнитов на поле при помощи баз.

Создаётся поле размером 3x3 с макс. Числом юнитов на нём 3.

Создаются две базы – база гномов и эльфов (автоматически располагаются по углам поля).

Далее, при помощи базы гномов создаётся юнит CRIT_ARCHER на позиции [1][1], а при помощи базы эльфов создаётся юнит ABSORB_INFANTRY на позиции [2][0]. Далее поле выводится на экран. Результат приведён на рисунке 1.

```
Example 1 - just showing that adding units with help of base is working successfully.
Generating field:
Base          no_creature      no_creature
no_object     no_object        no_object
Ground        Ground          Ground

no_creature    CritArcher      no_creature
no_object     no_object        no_object
Ground        Ground          Ground

AbsorbInfantry no_creature      Base
no_object     no_object        no_object
Ground        Ground          Ground
```

Рисунок 1 - пример 1

Пример 2.

В этом примере создаётся поле размером 3x3, макс. число юнитов – 3, а также по базе каждой расы. Как и в примере 1, создаются те же юниты на тех же позициях (CRIT_ARCHER на позиции [1][1], и ABSORB_INFANTRY на позиции [2][0]). Выводится вид поля. Далее CRIT_ARCHER присваивается урон в 1000 (для наглядности примера) и сначала он атакует ABSORB_INFANTRY, а затем атакует базу на позиции [2][2]. Потом выводится поле.

Пример изображён на рисунке 2.

```

Example 2
Generating field:
Base      no_creature  no_creature
no_object no_object    no_object
Ground    Ground      Ground

no_creature CritArcher no_creature
no_object  no_object  no_object
Ground    Ground      Ground

AbsorbInfantry no_creature Base
no_object      no_object no_object
Ground        Ground      Ground

Now archer at position [1][1] has 1000 damage set by administrator of the game and attacks the infantry at position [2][0] and base at [2][2]

Base      no_creature  no_creature
no_object no_object    no_object
Ground    Ground      Ground

no_creature CritArcher no_creature
no_object  no_object  no_object
Ground    Ground      Ground

no_creature no_creature no_creature
no_object  no_object  no_object
Ground    Ground      Ground

```

Рисунок 2 - пример 2

Пример 3.

В этом примере создаётся поле размером 3x3, макс. число юнитов – 3, а также по базе каждой расы. Далее добавляется Tower на позицию [1][1] и изменяются ландшафты клеток [1][0] и [1][1] на Water и Mountains соответственно. Затем добавляется кавалерия на позицию [1][1] и попадает под эффект нейтрального объекта Tower. Выводится поле, а после этого выводится бонус к броне у кавалерии, что показывает работоспособность влияния нейтральных объектов на юнитов.

Пример изображён на рисунке 3.

```

Example 3
Generating field:
Base      no_creature  no_creature
no_object no_object    no_object
Ground    Ground      Ground

no_creature no_creature no_creature
no_object  no_object  no_object
Ground    Ground      Ground

no_creature no_creature Base
no_object  no_object  no_object
Ground    Ground      Ground

Now add tower to [1][1] and change landscapes and [1][0] and [1][1]. Then add a cavalry to [1][1] and then the bonus armor from tower to cavalry will be printed.

Base      no_creature  no_creature
no_object no_object    no_object
Ground    Ground      Ground

no_creature AbsorbCavalry no_creature
no_object  Tower          no_object
Water      Mountains      Ground

no_creature no_creature Base
no_object  no_object  no_object
Ground    Ground      Ground

Bonus armor from tower:2
-----

```

Рисунок 3 - пример 3

Вывод.

В ходе выполнения лабораторной работы была написана программа, в которой реализованы классы для функционала программы и взаимодействия пользователя с программой. Был использован объектно-ориентированный стиль программирования, были изучены и применены его основные положения, а также реализованы некоторые паттерны проектирования.