**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 |  | Киреев К.А. |
| Преподаватель |  | 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 типа игрока. Размещается на игровом поле в месте, зависимом от типа игрока. Все юниты на поле создаются через базу своего типа. База является наблюдателем за своими юнитами. Базу можно уничтожить. Атаковать база не может. |
| UnitObserverInterface, UnitObservableByBaseInterface | Реализация паттерна «Наблюдатель» для базы, которая наблюдает за юнитами. |
| NeutralObjectInterface | Абстрактный класс, общий для всех нейтральных объектов |
| CreatureObserverInterface, CreatureObservableByFieldComponentInterface | Реализация паттерна «Наблюдатель» для клетки и юнита, который в ней находится. |
| LandscapeInterface | Общий интерфейс для всех видов ландшафта. |
| Proxy | 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 Water – нет, в Ground – да. По умолчанию везде генерируется Ground. |
| BoostTower, ArmorHouse, GunSmith, Village | Конкретные классы нейтральных объектов. Каждый нейтральный объект имеет свою стратегию по взаимодействию с каждым типом юнитов |
| NeutralObjectsEffectStrategyInterface | Интерфейс для всех стратегий взаимодействия юнитов с нейтральными объектами. |
| BoostTowerInfantryStrategy, BoostTowerCavalryStrategy, BoostTowerArcherStrategy, VillageInfantryStrategy, VillageCavalryStrategy, VillageArcherStrategy,  ArmorHouse.InfantryStrategy, ArmorHouse.CavalryStrategy, ArmorHouse ArcherStrategy,  GunSmith InfantryStrategy,  GunSmith CavalryStrategy,  GunSmith ArcherStrategy | Классы, реализующие паттерн «Стратегия» для взаимодействия юнитов с нейтральными объектами.  С каждым типом юнитов каждый нейтральный объект имеет свою стратегию взаимодействия. |

## Тестирование программы.

Для демонстрации работы программы были разработаны 3 примера.

**Пример 1.**

Создаётся поле размером 5х5.

Создаются две базы – база людей и эльфов.

Далее, при помощи базы людей создаётся юнит CrtDmgArc на позиции [1][1], а при помощи базы эльфов создаётся юнит DmgAbsInf на позиции [0][2]. Далее поле выводится на экран.

Результат приведён на рисунке 1.

Рисунок 1 - пример 1

**Пример 2.**

В этом примере создаётся поле размером 4х4, а также по базе каждой расы. Cоздаются те же юниты на тех же позициях. Выводится вид поля. Далее CrtDmgArc присваивается урон в 500 и сначала он атакует DmgAbsInf, а затем атакует базу на позиции [2][3]. Потом выводится поле.

Пример изображён на рисунке 2.

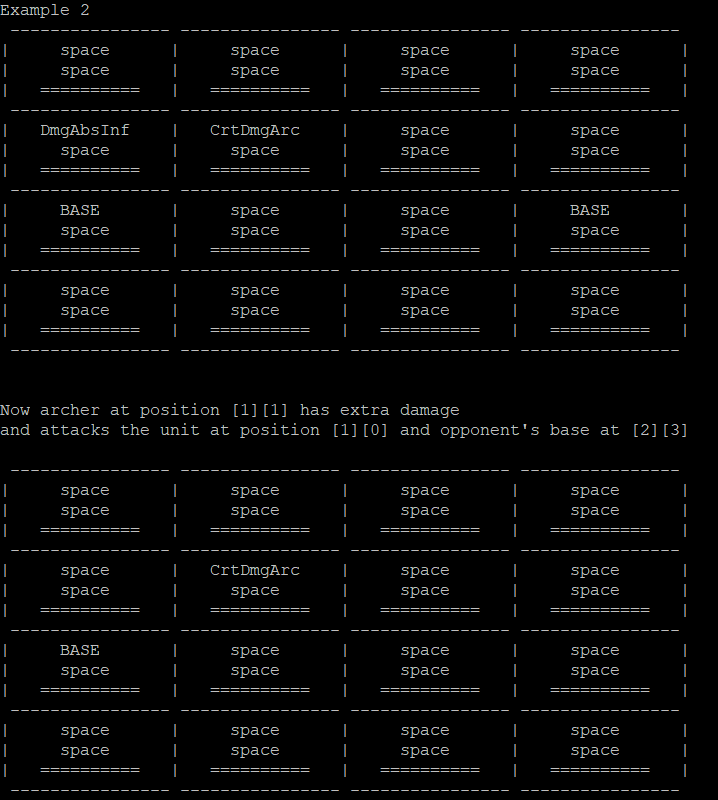


Рисунок 2 - пример 2

**Пример 3.**

В этом примере создаётся поле размером 3х3,, а также по базе каждой расы. Далее добавляется Village на позицию [0][0] и изменяются ландшафты клеток [2][2], [2][1] и [1][1]. Затем добавляется кавалерия на позицию [0][0] и попадает под эффект нейтрального объекта Village. Выводится поле, а после этого выводится бонус кавалерии, что показывает работоспособность влияния нейтральных объектов на юнитов.

Пример изображён на рисунке 3.

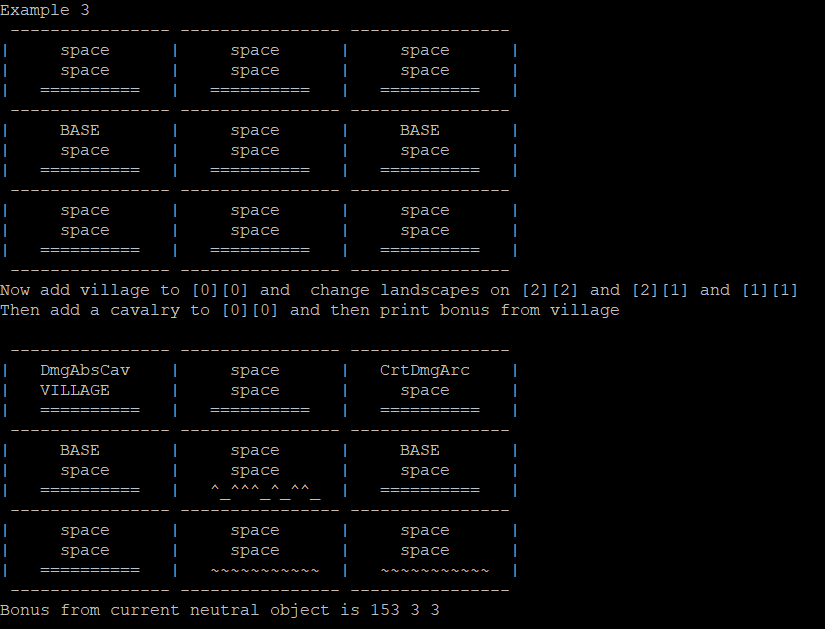


Рисунок 3 - пример 3

## 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.