МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Лабораторная работа №2

по дисциплине «Программирование»

Студент:

Лабин Макар Андреевич

Преподаватель:

Наумова Надежда Александровна

г. Санкт-Петербург, 2024

Текст задания

Лабораторная работа #2

На основе базового класса `Pokemon` написать свои классы для заданных видов покемонов. Каждый вид покемона должен иметь один или два типа и стандартные базовые характеристики:

- очки здоровья (HP)
- атака (attack)
- защита (defense)
- специальная атака (special attack)
- специальная защита (special defense)
- скорость (speed)

Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов `PhysicalMove`, `SpecialMove` и `StatusMove` реализовать свои классы для заданных видов атак.

Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

Используя класс симуляции боя `Battle`, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

Базовые классы и симулятор сражения находятся в jar-архиве (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - здесь.

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах http://poke-universe.ru, http://pokemondb.net, http://veekun.com/dex/pokemon

Введите вариант: 313102

Ваши покемоны:

Mesprit

Атаки:
- Shadow Ball
- Facade
- Thunderbolt
- Dream Eater

Sneasel

Атаки:
- Low Sweep
- Aerial Ace
- Swords Dance

Weavile

Атаки:
- Low Sweep
- Aerial Ace
- Swords Dance
- Nasty Plot

Poliwag

Атаки:
- Bubble Beam
- Confide

Poliwhirl

Атаки:
- Bubble Beam
- Confide
- Bubble

Politoed

Атаки:
- Bubble Beam
- Confide
- Bubble
- Swagger

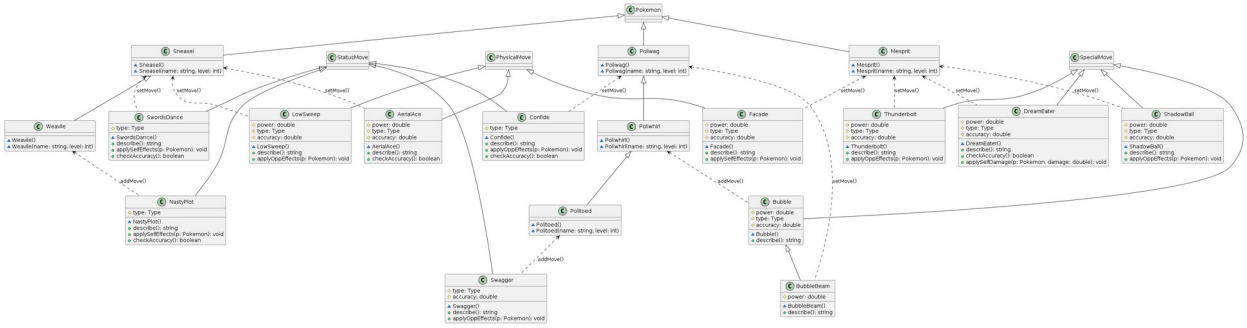
Комментарии

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с документацией, обращая особое внимание на классы `Pokemon` и `Move`. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл Pokemon.jar. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть как она работает.
```
Battle b = new Battle();
Pokemon p1 = new Pokemon("Чужой", 1);
Pokemon p2 = new Pokemon("Хищник", 1);
b.addAlly(p1);
b.addFoe(p2);
b.go();
```
4. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса `Pokemon`. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
5. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса `PhysicalMove` или `SpecialMove`. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод `describe`, чтобы выводилось нужное сообщение.
6. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники `StatusMove`), скорее всего придется разобраться с классом `Effect`. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
7. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Диаграмма классов реализованной объектной модели



Исходный код программы

С исходным кодом программы можно ознакомиться в репозитории GitHub по ссылке: https://github.com/MrDvD/itmo_labs/tree/master/lab2/src.

Результат работы программы

С результатом работы программы можно ознакомиться в репозитории GitHub по ссылке: https://github.com/MrDvD/itmo_labs/blob/master/lab2/file.out

Выводы по работе

В ходе выполнения лабораторной работы я вспомнил о базовых принципах объектно-ориентированного программирования: наследование, инкапсуляция и полиморфизм. Для работы с классами из внешнего jar-файла необходимо было изучить возможность его подключения без распаковки, прочитать описание classpath. В процессе реализации сформированной диаграммы классов объектной модели развивал навык внимательного чтения документации, прилагаемой к стороннему коду. Также столкнулся с трудностями переопределения используемых методов, которые были решены повторным прочтением документации.