МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант №51229

Выполнил:

Студент группы P3107

Шишкин Артём Владимирович

Проверил:

Данилов Павел Юрьевич

Санкт-Петербург, 2024

Содержание

Текст задания

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. Все разработанные классы, не имеющие наследников, должны быть реализованы таким образом, чтобы от них нельзя было наследоваться.

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

Комментарии

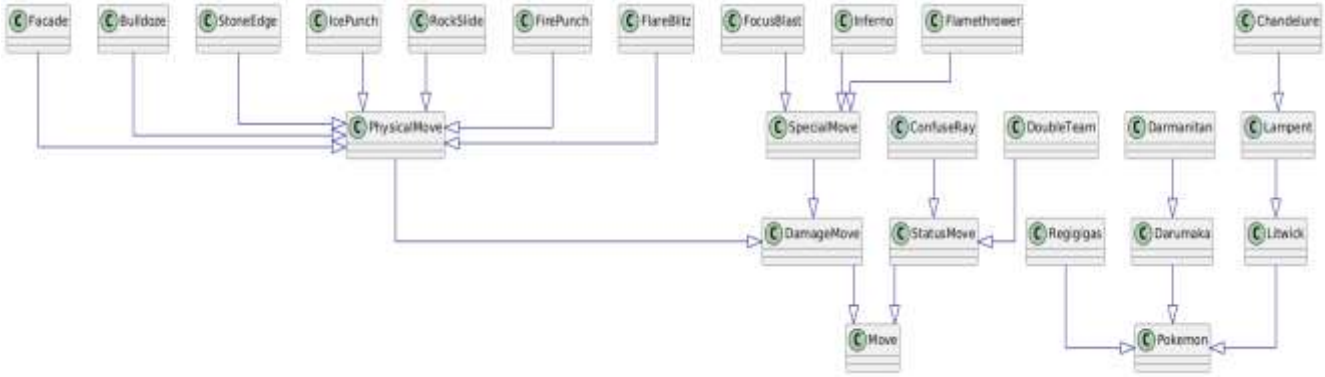
Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

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.
4. Battle b = new Battle();

5. Pokemon p1 = new Pokemon("Чужой", 1);
6. Pokemon p2 = new Pokemon("Хищник", 1);
7. b.addAlly(p1);
8. b.addFoe(p2);
9. b.go();
10. 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.
11. 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.
12. 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.
13. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Диаграмма классов



Исходный код

https://github.com/ShobakaBarobaka/ITMO/tree/main/prog_labs/lab2/src

Результат работы программы

Regigigas Дюдя из команды желтых вступает в бой!
Litwick Джефф из команды белых вступает в бой!
Regigigas Дюдя использует Confuse Ray.

Litwick Джефф использует Facade.
Regigigas Дюдя теряет 6 здоровья.

Regigigas Дюдя использует Confuse Ray.

Litwick Джефф растерянно попадает по себе.
Litwick Джефф теряет 4 здоровья.

Regigigas Дюдя использует Stone Edge.
Litwick Джефф теряет 16 здоровья.
Litwick Джефф теряет сознание.
Lampent Банни из команды белых вступает в бой!
Lampent Банни использует Inferno.
Regigigas Дюдя теряет 13 здоровья.
Regigigas Дюдя воспламеняется
Regigigas Дюдя теряет сознание.
Darumaka Уолтер из команды желтых вступает в бой!
Lampent Банни промахивается

Darumaka Уолтер использует Rock Slide.
Lampent Банни теряет 11 здоровья.

Darumaka Уолтер использует Fire Punch.
Lampent Банни теряет 5 здоровья.

Lampent Банни использует Inferno.
Darumaka Уолтер теряет 8 здоровья.

Darumaka Уолтер использует Flare Blaze.
Критический удар!
Lampent Банни теряет 12 здоровья.
Lampent Банни теряет сознание.
Chandelure Хэсус из команды белых вступает в бой!
Darumaka Уолтер использует Flare Blaze.
Chandelure Хэсус теряет 7 здоровья.

Chandelure Хэсус промахивается

Darumaka Уолтер использует Flare Blaze.
Chandelure Хэсус теряет 4 здоровья.

Chandelure Хэсус использует Inferno.
Darumaka Уолтер теряет 3 здоровья.

Darumaka Уолтер использует Rock Slide.
Chandelure Хэсус теряет 14 здоровья.
Chandelure Хэсус теряет сознание.
В команде белых не осталось покемонов.
Команда желтых побеждает в этом бою!

Заключение

В ходе выполнения лабораторной работы я узнал, как работает код Хэмминга, научился определять ошибки в сообщениях с помощью классического и неклассического кода, вычислять синдромы, контрольные суммы, коэффициент избыточности.