

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет Программной инженерии и компьютерных технологий
Направление: Нейротехнологии и программная инженерия

Дисциплина: Программирование
Лабораторная работа № 2
Вариант 666111

Выполнил студент
Рязанов Демид Витальевич
Группа № Р3121

Преподаватель: Петренко Никита Алексеевич

г. Санкт-Петербург

2022

Содержание

Задание.....	3
Отчет.....	4
Вывод.....	5

Задание

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>

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.addAilly(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.

Введите вариант:

Ваши покемоны:

Bouffalant  Атаки: ✓ Head Charge ✓ Rock Climb ✓ Stone Edge ✓ Leer	Baltoy  Атаки: ✓ Facade ✓ Psychic ✓ Charge Beam	Claydol  Атаки: ✓ Facade ✓ Psychic ✓ Charge Beam ✓ Stone Edge	Flabebe  Атаки: ✓ Facade ✓ Rest	Floette  Атаки: ✓ Facade ✓ Rest ✓ Razor Leaf	Florges  Атаки: ✓ Facade ✓ Rest ✓ Razor Leaf ✓ Confide
--	---	--	--	---	---

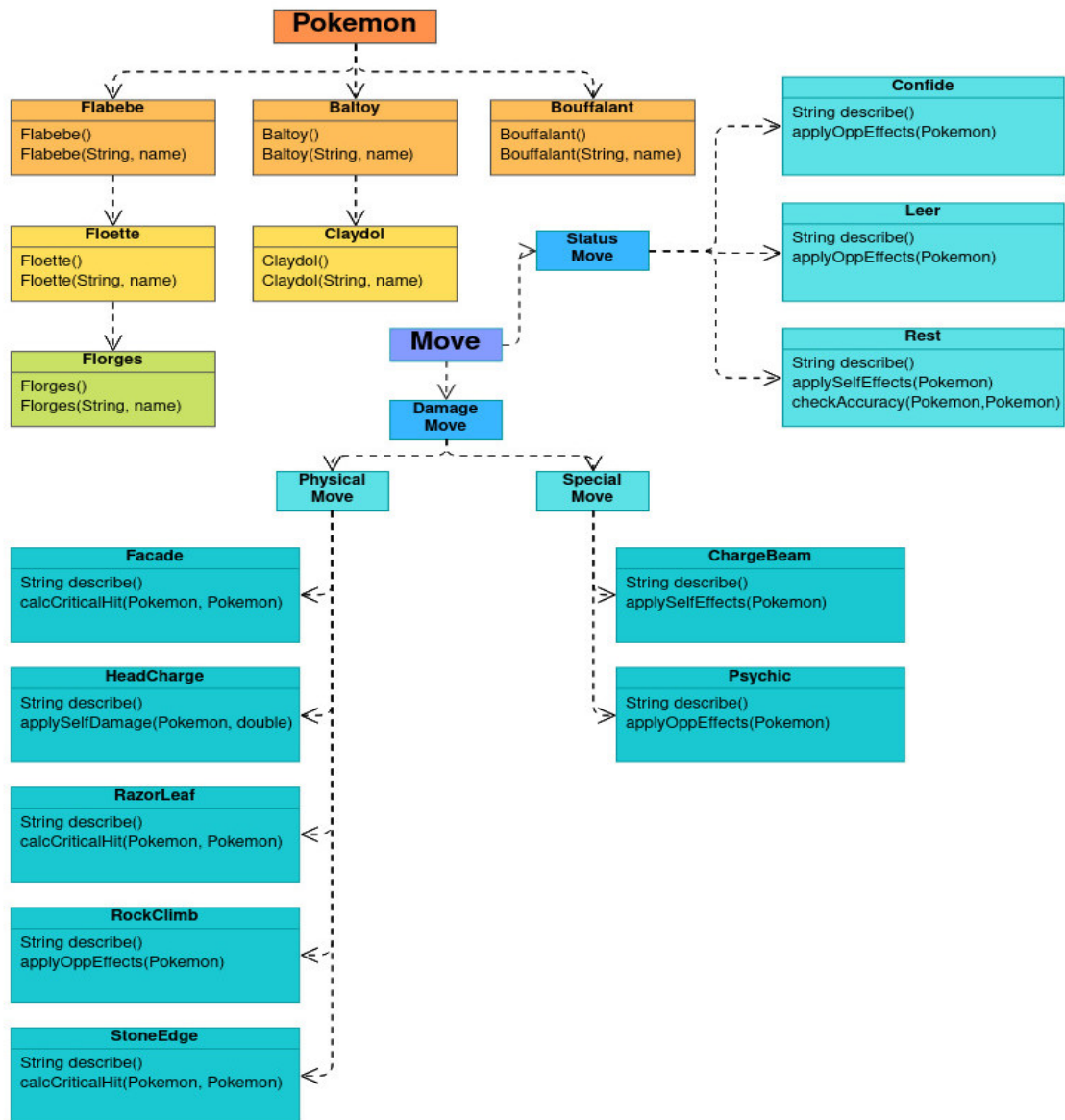
Отчет

Исходный код: <https://github.com/Demid0/Lab2>

Результат выполнения:

https://github.com/Demid0/Lab2/blob/master/out/artifacts/Lab2_jar/output.txt

Диаграмма классов:



Вывод

В ходе выполнения данной лабораторной работы я познакомился с основными понятиями ООП: объект, наследование, полиморфизм, инкапсуляция; инструкцией `import` и модификаторами `static` и `final`.