

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

Факультет: Программная инженерия

Дисциплина: «Программирование»

ОТЧЁТ

по лабораторной работе № 2

Вариант № 652381

Студент группы Р3121 : Бабушкин Александр Михайлович

Санкт-Петербург

2022 г.

Цель работы

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>

Описание работы

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

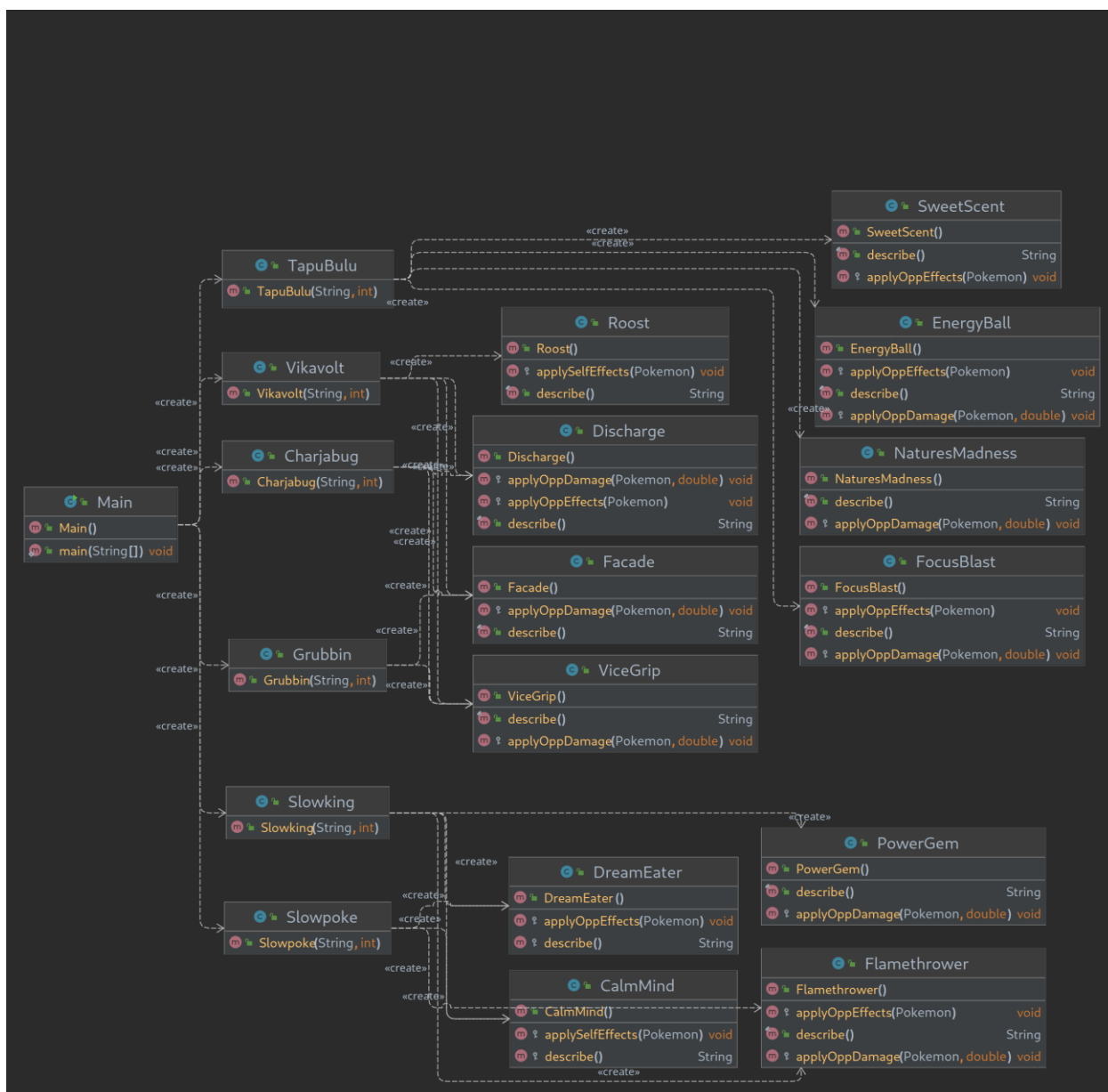
покемону и проверить ее действие в сражении. Не забудьте переопределить метод `describe`, чтобы выводилось нужное сообщение.

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.

Ваши покемоны:

<div>Tapu-Bulu</div> <div></div> <div>Атаки: ✂ Energy Ball ✂ Sweet Scent ✂ Nature's Madness ✂ Focus Blast</div>	<div>Slowpoke</div> <div></div> <div>Атаки: ✂ Flamethrower ✂ Calm Mind ✂ Dream Eater</div>	<div>Slowking</div> <div></div> <div>Атаки: ✂ Flamethrower ✂ Calm Mind ✂ Dream Eater ✂ Power Gem</div>	<div>Grubbin</div> <div></div> <div>Атаки: ✂ Vice Grip ✂ Facade</div>	<div>Charjabug</div> <div></div> <div>Атаки: ✂ Vice Grip ✂ Facade ✂ Discharge</div>	<div>Vikavolt</div> <div></div> <div>Атаки: ✂ Vice Grip ✂ Facade ✂ Discharge ✂ Roost</div>
---	--	--	--	---	--

UML диаграмма классов: ([github](#))



Исходный код

[Github](#)

Результат работы

TapuBulu Leon из команды фиолетовых вступает в бой!
Grubbin El Primo из команды синих вступает в бой!
TapuBulu Leon специальную атаку.
Grubbin El Primo теряет 4 здоровья.

Grubbin El Primo атаку.
TapuBulu Leon теряет 5 здоровья.

TapuBulu Leon специальную атаку.
Grubbin El Primo теряет 4 здоровья.

Grubbin El Primo атаку.
TapuBulu Leon теряет 3 здоровья.

TapuBulu Leon специальную атаку.
Grubbin El Primo теряет 4 здоровья.

Grubbin El Primo атаку.
TapuBulu Leon теряет 4 здоровья.

TapuBulu Leon специальную атаку.
Grubbin El Primo теряет 3 здоровья.
Grubbin El Primo теряет сознание.
Charjabug Shelly` из команды синих вступает в бой!
TapuBulu Leon специальную атаку.
Charjabug Shelly` теряет 2 здоровья.

Charjabug Shelly` специальную атаку.
TapuBulu Leon теряет 3 здоровья.
TapuBulu Leon теряет сознание.
Slowpoke Colt из команды фиолетовых вступает в бой!
Charjabug Shelly` специальную атаку.
Slowpoke Colt теряет 20 здоровья.
Slowpoke Colt теряет сознание.
Slowking Dynamike из команды фиолетовых вступает в бой!
Charjabug Shelly` специальную атаку.
Slowking Dynamike теряет 12 здоровья.
Slowking Dynamike парализован

Slowking Dynamike применяет DreamEater.
Критический удар!
Charjabug Shelly` теряет 16 здоровья.
Charjabug Shelly` теряет сознание.
Vikavolt Spike из команды синих вступает в бой!
Slowking Dynamike специальную атаку.
Критический удар!
Vikavolt Spike теряет 17 здоровья.
Vikavolt Spike теряет сознание.
В команде синих не осталось покемонов.
Команда фиолетовых побеждает в этом бою!

Вывод

Во время выполнение данной лабораторной работы мной были освоены базовые навыки объектно ориентированного программирования.