



Вариант № 3109999
Лабораторная работа № 2
По дисциплине
Программирование

Выполнил студент группы Р3109:
Михальченков Александр

Преподаватель:
Наумова Надежда Александровна

1. Текст задания

Лабораторная работа #2

Пособие для подготовки к работе

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.

```
Battle b = new Battle();
Pokemon p1 = new Pokemon("Чужой", 1);
Pokemon p2 = new Pokemon("Хищник", 1);
b.addAll(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:

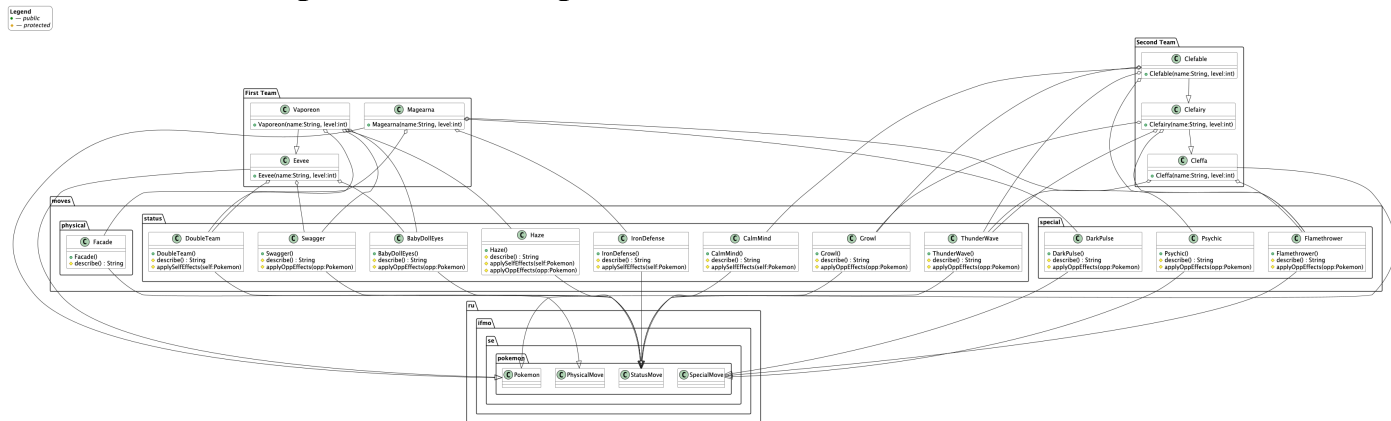
Ваши покемоны:

| | | | | | |
|--|--|---|---|--|---|
| Magearna  Атаки: ✓ Facade ✓ Dark Pulse ✓ Iron Defense ✓ Psychic | Eevee  Атаки: ✓ Swagger ✓ Baby-Doll Eyes ✓ Double Team | Vaporeon  Атаки: ✓ Swagger ✓ Baby-Doll Eyes ✓ Double Team ✓ Haze | Clefpa  Атаки: ✓ Thunder Wave ✓ Flamethrower | Clefairy  Атаки: ✓ Thunder Wave ✓ Flamethrower ✓ Growl | Clefable  Атаки: ✓ Thunder Wave ✓ Flamethrower ✓ Growl ✓ Calm Mind |
|--|--|---|---|--|---|

2. Исходный код программы

Репозиторий на GitHub: <https://github.com/mikhalexandr/itmo-cse-programming/tree/main/lab2>

3. Диаграмма классов реализованной объектной модели



4. Результат работы программы

```
task run
Магearна Магearна из команды желтых вступает в бой!
Clefа Клефа из команды фиолетовых вступает в бой!
Магearна Магearна промахивается

Clefа Клефа использует Flamethrower.
Магearна Магearна теряет 9 здоровья.

Магearна Магearна промахивается

Clefа Клефа использует Thunder Wave.
Магearна Магearна парализован

Магearна Магearна использует Facade.
Критический удар!
Clefа Клефа теряет 215 здоровья.
Clefа Клефа теряет сознание.
Clefairy Клефейри из команды фиолетовых вступает в бой!
Магearна Магearна промахивается

Clefairy Клефейри использует Growl.
Магearна Магearна уменьшает атаку.

Магearна Магearна промахивается

Clefairy Клефейри использует Thunder Wave.

Магearна Магearна использует Psychic.
Критический удар!
Clefairy Клефейри теряет 111 здоровья.
Clefairy Клефейри уменьшает специальную защиту.
Clefairy Клефейри теряет сознание.
Clefable Клефейбл из команды фиолетовых вступает в бой!
Магearна Магearна использует Facade.
Критический удар!
Clefable Клефейбл теряет 68 здоровья.

Clefable Клефейбл использует Flamethrower.
Магearна Магearна теряет 30 здоровья.

Clefable Клефейбл использует Flamethrower.
Магearна Магearна теряет 36 здоровья.

Clefable Клефейбл промахивается

Магearна Магearна использует Dark Pulse.
Критический удар!
Clefable Клефейбл теряет 28 здоровья.

Clefable Клефейбл использует Growl.
Магearна Магearна уменьшает атаку.

Магearна Магearна использует Psychic.
Критический удар!
Clefable Клефейбл теряет 82 здоровья.
Clefable Клефейбл теряет сознание.
В команде фиолетовых не осталось покемонов.
Команда желтых побеждает в этом бою!
```

5. Выводы по работе

В ходе выполнения лабораторной работы я познакомился с принципами объектно-ориентированного программирования на языке Java и получил практические навыки разработки класса приложения с использованием пользовательских классов и наследования. Я научился создавать собственные классы и реализовывать между ними отношения наследования, а также переопределять методы для задания индивидуального поведения объектов. В процессе работы освоил работу со сторонней библиотекой (Pokemon.jar), использование пакетов, работу с массивами объектов, а также организацию и запуск боёв между командами покемонов. Я считаю, что полученные знания и навыки будут полезны для дальнейшего совершенствования в программировании на Java и понимания принципов ООП.