

НИУ ИТМО

Факультет ПИИКТ
Программирование

Лабораторная работа №2

Вариант 311631

Выполнил Бойко Г. А., группа Р3116

г. Санкт-Петербург, 2024

Задание:

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. 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 покемона 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.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.

Введите вариант:

Ваши покемоны:

<div>Cosmog  Атаки: ✓ Moonblast ✓ Facade ✓ Swagger ✓ Water Pulse</div>	<div>Kabuto  Атаки: ✓ Aerial Ace ✓ Aqua Jet ✓ Leer ✓ Leer</div>	<div>Kabutops  Атаки: ✓ Aerial Ace ✓ Aqua Jet ✓ Leer ✓ Slash</div>	<div>Seedot  Атаки: ✓ Double Team ✓ Facade</div>
<div>Nuzleaf  Атаки: ✓ Double Team ✓ Facade ✓ Extrasensory</div>	<div>Shiftry  Атаки: ✓ Double Team ✓ Facade ✓ Extrasensory ✓ Rest</div>		

Исходный код программы:

<https://github.com/georgeboiko/proga-lab2>

Результат работы программы:

Cosmog Сигма из команды белых вступает в бой!

Nuzleaf Билли из команды фиолетовых вступает в бой!

Nuzleaf Билли attacks with Extrasensory.

Cosmog Сигма теряет 3 здоровья.

Cosmog Сигма attacks with Swagger.

Nuzleaf Билли attacks with Facade.

Cosmog Сигма теряет 3 здоровья.

Cosmog Сигма attacks with Water Pulse.

Nuzleaf Билли теряет 2 здоровья.

Nuzleaf Билли растерянно попадает по себе.

Nuzleaf Билли теряет 5 здоровья.

Cosmog Сигма attacks with Facade.

Nuzleaf Билли теряет 1 здоровья.

Nuzleaf Билли не замечает воздействие типа PSYCHIC

Nuzleaf Билли промахивается

Cosmog Сигма attacks with Facade.

Nuzleaf Билли теряет 1 здоровья.

Nuzleaf Билли не замечает воздействие типа PSYCHIC

Nuzleaf Билли растерянно попадает по себе.

Nuzleaf Билли теряет 4 здоровья.

Nuzleaf Билли теряет сознание.

Seedot Нормис из команды фиолетовых вступает в бой!

Cosmog Сигма attacks with Moonblast.

Seedot Нормис теряет 4 здоровья.

Seedot Нормис промахивается

Cosmog Сигма attacks with Facade.

Seedot Нормис теряет 6 здоровья.

Seedot Нормис промахивается

Cosmog Сигма attacks with Facade.

Seedot Нормис теряет 6 здоровья.

Seedot Нормис теряет сознание.

Shiftry Арни из команды фиолетовых вступает в бой!

Shiftry Арни attacks with Facade.

Cosmog Сигма теряет 2 здоровья.

Cosmog Сигма attacks with Swagger.

Shiftry Арни attacks with Facade.

Cosmog Сигма теряет 3 здоровья.

Cosmog Сигма attacks with Moonblast.

Shiftry Арни теряет 12 здоровья.

Shiftry Арни растерянно попадает по себе.

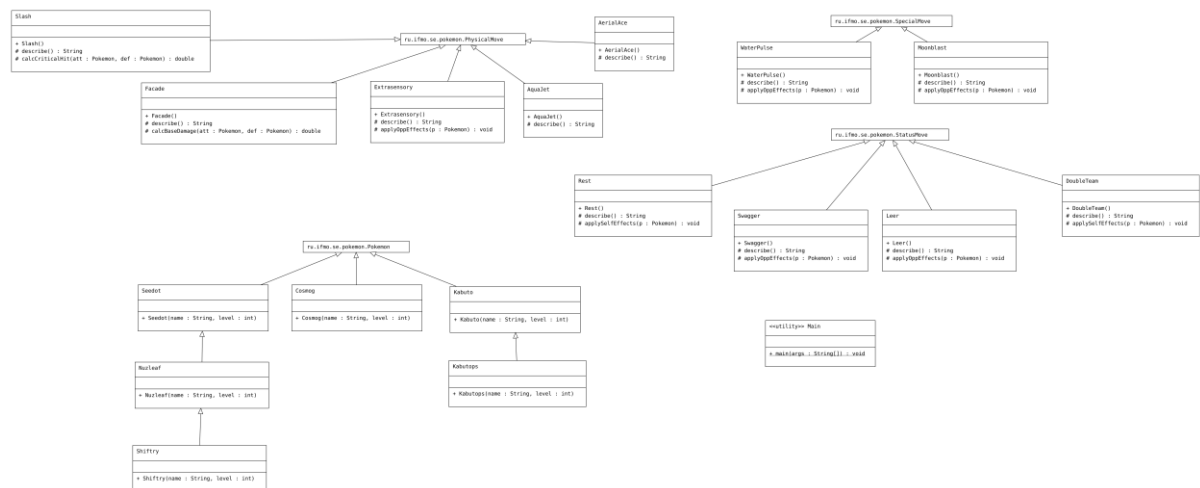
Shiftry Арни теряет 4 здоровья.

Shiftry Арни теряет сознание.

В команде фиолетовых не осталось покемонов.

Команда белых побеждает в этом бою!

Диаграмма классов:



Заключение:

В процессе выполнения лабораторной работы я познакомился с ООП, изучил его основные признаки. Научился строить диаграммы классов, подключать сторонние библиотеки к проектам.