



Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа №2
Вариант 1033567

Выполнил: студент группы
Р3110 М.А. Маракушев

Преподаватель:
Сорокин Р. Б.

г. Санкт-Петербург
2021

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.

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.

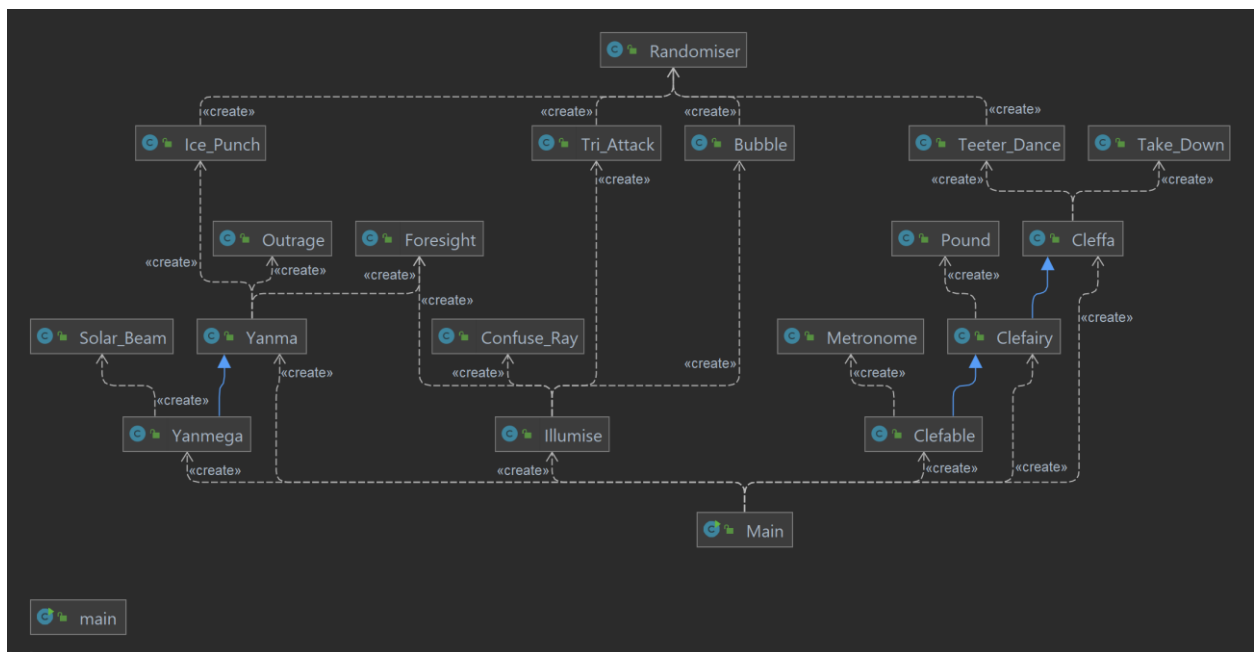
Что надо сделать (краткое описание)

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 реализованной объектной модели



Ссылка на код на GitHub:

<https://github.com/Watcher0/Laba2>

Пример вывода программы

Plumise Йода из команды желтых вступает в бой!

Yanmega Шмель из команды красных вступает в бой!

Yanmega Шмель промахивается

Plumise Йода промахивается

Yanmega Шмель атакует.

Plumise Йода теряет 16 здоровья.

Plumise Йода теряет 227 здоровья.

Plumise Йода теряет сознание.

Yanma Муха из команды желтых вступает в бой!

Yanmega Шмель растерянно попадает по себе.

Yanmega Шмель теряет 6 здоровья.

Yanma Муха использовал Ice Punch.

Yanmega Шмель теряет 10 здоровья.

Yanmega Шмель замерзает

Yanma Муха использовал Ice Punch.

Yanmega Шмель теряет 7 здоровья.

Yanma Муха использовал Ice Punch.

Yanmega Шмель теряет 8 здоровья.

Yanmega Шмель теряет сознание.

Clefairy Ева из команды красных вступает в бой!

Yanma Муха промахивается

Clefairy Ева использует Take Down.

Yanma Муха теряет 6 здоровья.

Clefairy Ева теряет 1 здоровья.

Yanma Муха использовал Ice Punch.

Clefairy Ева теряет 5 здоровья.

Clefairy Ева замерзает

Clefairy Ева использовал Teeter Dance.

Clefairy Ева оттаивает

Yanma Муха использовал Ice Punch.

Clefairy Ева теряет 4 здоровья.

Clefairy Ева замерзает

Clefairy Ева использовал Pound.

Yanma Муха теряет 3 здоровья.

Clefairy Ева оттаивает

Yanma Муха использовал Ice Punch.

Clefairy Ева теряет 7 здоровья.

Clefairy Ева замерзает

Clefairy Ева использует Take Down.

Yanma Муха теряет 5 здоровья.

Clefairy Ева теряет 1 здоровья.

Clefairy Ева теряет сознание.

Cleffa Я, пришедший в ИТМО. из команды красных вступает в бой!

Yanma Муха атакует.

Cleffa Я, пришедший в ИТМО. теряет 1 здоровья.

Cleffa Я, пришедший в ИТМО. не замечает воздействие типа DRAGON

Cleffa Я, пришедший в ИТМО. использовал Teeter Dance.

Yanma Муха промахивается

Cleffa Я, пришедший в ИТМО. использует Take Down.

Yanma Муха теряет 3 здоровья.

Yanma Муха растерянно попадает по себе.

Yanma Муха теряет 6 здоровья.

Yanma Муха теряет сознание.

Clefable Пол из команды желтых вступает в бой!

Clefable Пол промахивается

Cleffa Я, пришедший в ИТМО. использовал Teeter Dance.

Clefable Пол использовал Teeter Dance.

Cleffa Я, пришедший в ИТМО. использовал Teeter Dance.

Clefable Пол использовал Pound.

Cleffa Я, пришедший в ИТМО. теряет 10 здоровья.

Cleffa Я, пришедший в ИТМО. использовал Teeter Dance.

Clefable Пол использовал Teeter Dance.

Cleffa Я, пришедший в ИТМО. использует Take Down.

Clefable Пол теряет 5 здоровья.

Cleffa Я, пришедший в ИТМО. теряет 1 здоровья.

Clefable Пол использовал Teeter Dance.

Cleffa Я, пришедший в ИТМО. растерянно попадает по себе.

Cleffa Я, пришедший в ИТМО. теряет 4 здоровья.

Cleffa Я, пришедший в ИТМО. теряет сознание.

В команде красных не осталось покемонов.

Команда желтых побеждает в этом бою!

Выводы

Таким образом, в ходе выполнения данной лабораторной работы мною были освоены основные понятия ООП в Java. Я изучил такие принципы ООП, как абстракция, инкапсуляция, наследование и полиморфизм. Я также впервые основывал код проекта на предоставленном архиве jar, анализируя его и наследуя от его классов. Мною была получена диаграмма классов, подробно показывающая преимущества работы ООП.