Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.04 – Программная инженерия,
Системное и прикладное программное обеспечение.
Дисциплина «Программирование»

Отчет
По лабораторной работе №2
Программа на языке Java

Вариант № 9911230013

Выполнил:
Молчанов Фёдор Денисович

Группа: P3113

Преподаватель:
Иманзаде Фахри Рашидович

Г. Санкт-Петербург, 2023 г.

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.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:

Sableye

Атаки:
- Dream Eater
- Calm Mind
- Recover
- Dazzling Gleam

Eevee

Атаки:
- Confide
- Baby-Doll Eyes
- Facade

Sylveon

Атаки:
- Confide
- Baby-Doll Eyes
- Facade
- Draining Kiss

Lotad

Атаки:
- Energy Ball
- Blizzard

Lombre

Атаки:
- Energy Ball
- Blizzard
- Fury Swipes

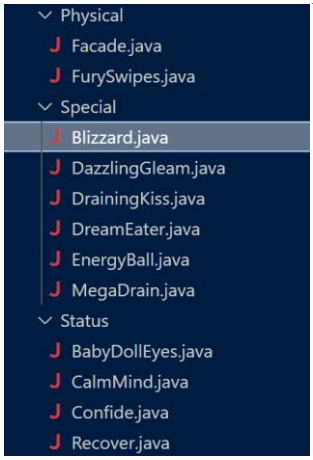
Ludicolo

Атаки:
- Energy Ball
- Blizzard
- Fury Swipes
- Mega Drain

Решение

Добавляю скаченный jar в папку Lib, а затем подключаю подключаю его к своей программе.

Создаю папку Moves, реализуя различные типы атак. Делю их на три подпапки в зависимости от типа: Special, Physical, Status.



Пример реализации SpecialMove:

```java
package moves.Special;

import ru.ifmo.se.pokemon.*;

public class Blizzard extends SpecialMove {
    public Blizzard(){
        super(Type.ICE, 110, 70);
    }
    boolean freezeFlag = false;
    @Override
    protected void applyOppEffects(Pokemon p){
        int freezeChance = (int)(Math.random()*101);
        if (freezeChance <= 10){
            Effect.freeze(p);
            freezeFlag = true;
        }
    }
    @Override
    protected String describe(){
        if (freezeFlag == true){
            return "Использует Blizzard и замораживает противника";
        }
        else{
            return "промахивается и не замораживает проитвника с помощью Blizzard";
        }
    }
}
```

Примечание: Override используется для переопределения существующих методов класса SpecialMove, от которого наследуется Blizzard.

Далее создаю папку Pokemons, в которой прописываю характеристики каждого из них, а также доступные им атаки.

Пример одного из покемонов:

```java
package pokemons;

import moves.Status.*;
import moves.Physical.*;
import ru.ifmo.se.pokemon.*;

public class Eevee extends Pokemon{
    public Eevee(String name, int level){
        super(name, level);
        setStats(55, 55, 50, 45,65, 55);
        setType(Type.NORMAL);
        setMove(new Confide(), new BabyDollEyes(), new Facade());
    }

}
```

Выводы

В ходе работы я познакомился с основами ООП(наследование, полиморфизм, инкапсуляция), поработал с существующей библиотекой, лучше узнал методы, модификаторы доступа.