федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия»

Дисциплина «программирование»

Отчет по лабораторной работе

По теме:

Перевод чисел между различными системами счисления

Вариант №1235

Амузинский Артем Андреевич P3106

Руководитель ЛР: _____ Саржевский Иван Анатольевич

Санкт-Петербург 2023

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.

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.

Покемоны:

Ваши покемоны:

Решение:

Класс Main.java

```
import ru.ifmo.se.pokemon.*;
import Pokemons.*;

public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();
        b.addAlly(new Charjabug("Конченный идиот", 1));
        b.addAlly(new Cosmog("Самый крутой мужик в мире", 2));
        b.addAlly(new Grubbin("Горячая чикса", 1));
        b.addFoe(new Nosepass("Злодей британец", 3));
        b.addFoe(new Probopass("Так себе шутник", 1));
        b.addFoe(new Vikavolt("Какой-то мужик", 2));
        b.go();
    }
}
```

Покемоны:

```
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Charjabug extends Pokemon {
    public Charjabug(String name, int level){
        super(name, level);
        setStats(57, 82, 95, 55, 75, 36);
        setType(Type.ELECTRIC, Type.BUG);
        setMove(new Thunder_Wave(), new Rest(), new Dis-
charge());
    }
}
```

```
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Cosmog extends Pokemon{
    public Cosmog(String name, int level){
        super(name, level);
        setStats(43, 29, 31, 29, 31, 37);
        setType(Type.PSYCHIC);
        setMove(new Shadow_Ball(), new Natures_Madnes(), new Fa-
cade(), new Moonblast());
    }
}
```

```java
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Grubbin extends Pokemon {
    public Grubbin(String name, int level){
        super(name, level);
        setStats(47, 62, 45, 55, 45, 46);
        setType(Type.BUG);
        setMove(new Thunder_Wave(), new Rest());
    }
}
```

```java
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Nosepass extends Pokemon {
    public Nosepass(String name, int level){
        super(name, level);
        setStats(30, 45, 135, 45, 90, 30);
        setType(Type.ROCK);
        setMove(new Power_Gem(), new Facade(), new Discharge());
    }
}
```

```java
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Probopass extends Pokemon {
    public Probopass(String name, int level){
        super(name, level);
        setStats(60, 55, 145, 75, 150, 40);
        setType(Type.ROCK, Type.STEEL);
        setMove(new Power_Gem(), new Discharge(), new Facade(),
new Magnet_Bomb());
    }
}
```

```
package Pokemons;

import Attaks.*;
import ru.ifmo.se.pokemon.*;

public class Vikavolt extends Pokemon {
    public Vikavolt(String name, int level){
        super(name, level);
        setStats(77, 70, 90, 145, 75, 43);
        setType(Type.BUG, Type.ELECTRIC);
        setMove(new Thunder_Wave(), new Rest(), new Discharge(),
new Zap_Cannon());
    }
}
```

Атаки покемонов:

```
package Attaks;

import ru.ifmo.se.pokemon.*;

public class Discharge extends SpecialMove {
    public Discharge(){
        super(Type.ELECTRIC, 80, 100);
    }
    protected void applyOppEffects(Pokemon a) {
        a.addEffect(new Ef-
fect().chance(0.3).condition(Status.PARALYZE));
    }
    @Override
    protected String describe() {
        return "Использую Discharge";
    }
}
```

```
package Attaks;

import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Type;

public class Magnet_Bomb extends PhysicalMove{
    public Magnet_Bomb(){
        super(Type.STEEL, 60, Double.POSITIVE_INFINITY);
    }
    protected String describe() {
        return "Использую Magnet_Bomb";
    }
}
```

```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Facade extends PhysicalMove {
    public Facade(){
        super(Type.NORMAL, 70, 100);
    }
    private boolean flag;
    @Override
    public void applyOppDamage(Pokemon poc, double damage){
        Status cond = poc.getCondition();
        flag = true;
        if (cond.equals(Status.POISON) ||
cond.equals(Status.BURN) || cond.equals(Status.PARALYZE)) {
            poc.setMod(Stat.HP, -2*(int)Math.round(damage));
        }
    }
    @Override
    protected String describe(){
        if(flag) return "сильно бью";
        else return "бью обыкновенно";
    }
}
```

```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Moonblast extends StatusMove {
    public Moonblast(){
        super(Type.FAIRY, 95, 100);
    }
    protected void applyOppEffects(Pokemon a) {
        a.addEffect(new Ef-
fect().chance(0.3).condition(Status.BURN));
    }
    protected String describe() {
        return "Использую Moonblast";
    }
}
```

```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Natures_Madnes extends StatusMove {
    public Natures_Madnes(){
        super(Type.FAIRY, 0, 90);
    }
    protected String describe() {
        return "использую Natures_Madnes";
    }
    public void applyOppDamage(Pokemon poc, double damage){
        poc.setMod(Stat.HP, -2*(int)Math.round(damage));
    }
}
```

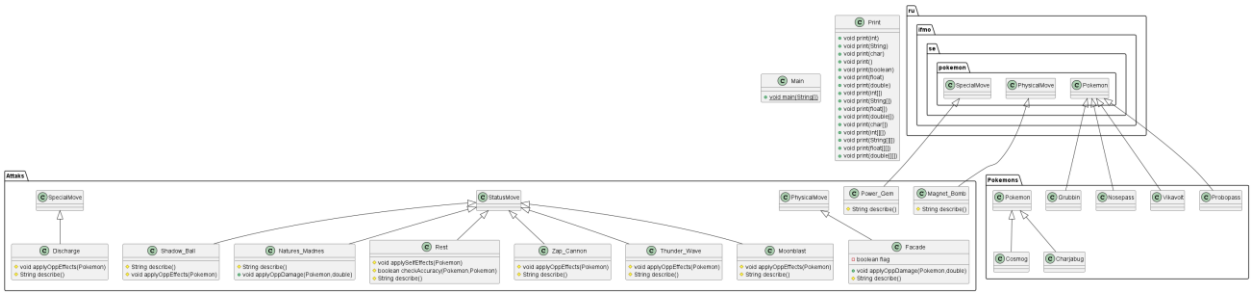
```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Power_Gem extends SpecialMove {
    public Power_Gem(){
        super(Type.ROCK, 80, 100);
    }
    protected String describe() {
        return "Использую Power_Gem";
    }
}
```

```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Rest extends StatusMove{
    public Rest(){
        super(Type.NORMAL, 0, 0);
    }
    @Override
    protected void applySelfEffects(Pokemon att){
        Effect eff = new Effect();
        eff = eff.condition(Status.SLEEP);
        eff = eff.turns(2);
        att.restore();
        att.addEffect(eff);
    }
    protected boolean checkAccuracy(Pokemon att,Pokemon def){
        return true;
    }
    @Override
    protected String describe(){
        return "Спит 2 хода";
    }
}
```

```java
package Attaks;
import ru.ifmo.se.pokemon.*;
public class Shadow_Ball extends StatusMove {
    public Shadow_Ball(){
        super(Type.GHOST, 80, 1);
    }
    @Override
    protected String describe() {
        return "использую Shadow Ball";
    }
    @Override
    protected void applyOppEffects(Pokemon p) {
        if(Math.random() <= 0.2)
            p.addEffect(new Ef-
fect().chance(0.25).stat(Stat.SPECIAL_DEFENSE, -1));
    }
}
```

```java
package Attaks;

import ru.ifmo.se.pokemon.*;

public class Thunder_Wave extends StatusMove {
    public Thunder_Wave() {
        super(Type.ELECTRIC, 0, 0.9);
    }

    protected void applyOppEffects(Pokemon a) {
        a.addEffect(new Ef-
fect().chance(0.25).condition(Status.PARALYZE));
        if (a.getLevel() <= 6 && a.getLevel() >= 1)
a.addEffect(new Effect().stat(Stat.SPEED,
(int)(a.getStat(Stat.SPEED) * 0.25)));
        else a.addEffect(new Effect().stat(Stat.SPEED,
(int)a.getStat(Stat.SPEED) / 2));
    }

    @Override
    protected String describe() {
        return "парализую противника с помощью Thunder Wave";
    }
}
```

```java
package Attaks;


import ru.ifmo.se.pokemon.*;

public class Zap_Cannon extends StatusMove {
    public Zap_Cannon() {

        super(Type.ELECTRIC, 120, 50);
    }

    @Override
    protected void applyOppEffects(Pokemon a) {
        a.addEffect(new Ef-
fect().chance(0.25).condition(Status.PARALYZE));
        if (a.getLevel() <= 6 && a.getLevel() >= 1)
a.addEffect(new Effect().stat(Stat.SPEED,
(int)(a.getStat(Stat.SPEED) * 0.25)));
        else a.addEffect(new Effect().stat(Stat.SPEED,
(int)a.getStat(Stat.SPEED) / 2));
    }

    @Override
    protected String describe() {
        return "парализую противника с помощью Zap_Cannon";
    }
}
```


Print
Main
ru
zho
se
pokemon
SpecialMove
PhysicalMove
Pokemon
Attacks
SpecialMove
StatusMove
PhysicalMove
Power_Gem
Magnet_Bomb
Discharge
Shadow_Ball
Natures_Madnes
Rest
Zap_Cannon
Thunder_Wave
Moonblast
Facade
Pokemons
Pokemon
Grubbin
Nosepass
Vikavolt
Probopass
Cosmog
Charjabug