Федеральное государственное автономное
образовательное учреждение высшего образования

Университет ИТМО

Кафедра Вычислительной Техники

Лабораторная работа №2

Курса “Программирование”

Вариант 262

Выполнил:
Кривоносов Егор Дмитриевич
Группа: P3111

Преподаватель:
Гаврилов А.В.

2019 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.

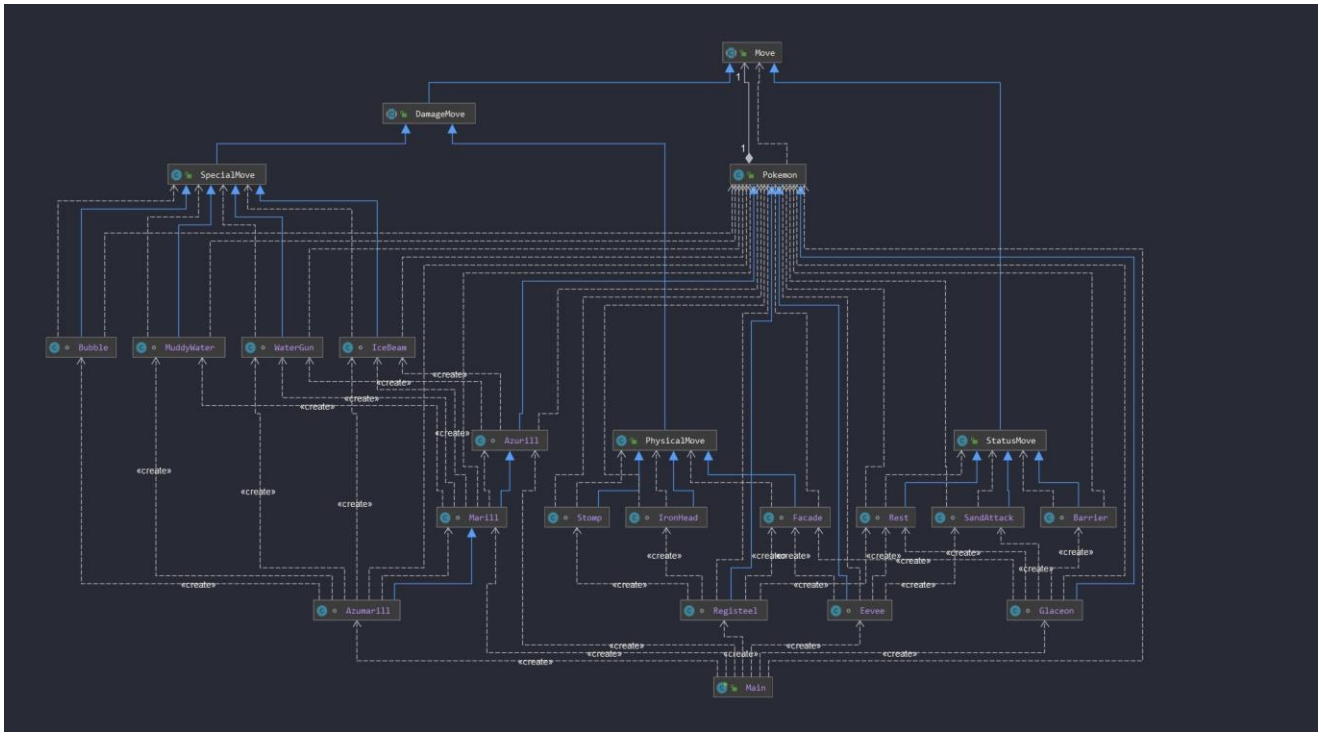
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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

Диаграмма классов:



Исходный код:

Main.java:

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


// Second lab //
public class Main {


   public static void main(String[] args) {
       Battle b = new Battle();


       Pokemon p1 = new Registeel("", 1);
       Pokemon p2 = new Eevee("", 1);
       Pokemon p3 = new Glaceon("", 1);
       Pokemon p4 = new Azurill("", 1);
       Pokemon p5 = new Marill("", 1);
```

```
        Pokemon p6 = new Azumarill("", 1);


        b.addAlly(p1);
        b.addAlly(p2);
        b.addAlly(p3);
        b.addFoe(p4);
        b.addFoe(p5);
        b.addFoe(p6);


        b.go();}}
```

Pokemons.java:

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


class Registeel extends Pokemon {
    Registeel (String name, int level){
        super(name, level);
        setStats(80,75,150,75,150,50);
        setType(Type.STEEL);
        setMove(new IronHead(), new Stomp(), new Facade(), new Rest());
    }
}
class Eevee extends Pokemon {
    Eevee (String name, int level){
        super(name, level);
        setStats(55,55,50,45,65,55);
        setType(Type.NORMAL);
        setMove(new SandAttack(), new Facade(), new Rest());
    }
```

```
}
class Glaceon extends Pokemon {
   Glaceon (String name, int level){
      super(name, level);
      setStats(65,60,110,130,95,65);
      setType(Type.ICE);
      setMove(new SandAttack(), new Facade(), new Rest(), new Barrier());
   }
}
class Azurill extends Pokemon {
   Azurill (String name, int level){
      super(name, level);
      setStats(50,20,40,20,40,20);
      setType(Type.NORMAL,Type.FAIRY);
      setMove(new WaterGun(), new IceBeam());
   }
}
class Marill extends Azurill {
   Marill (String name, int level){
      super(name, level);
      setStats(70,20,50,20,50,40);
      setType(Type.WATER,Type.FAIRY);
      setMove(new WaterGun(), new IceBeam(), new MuddyWater());
   }
}



class Azumarill extends Marill {
```

```
    Azumarill (String name, int level){
        super(name, level);
        setStats(100,50,80,60,80,50);
        setType(Type.WATER,Type.FAIRY);
        setMove(new WaterGun(), new IceBeam(), new MuddyWater(), new Bubble());
    }
}
```

Sposobnosti.java:

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

class IronHead extends PhysicalMove{
    protected  IronHead(){
        super(Type.STEEL,80,100);
    }
    @Override
    protected void applyOppEffects(Pokemon p){
        if (Math.random() <= 0.3) Effect.flinch(p);
        p.setMod(Stat.ACCURACY, 0);
    }
    @Override
    protected String describe(){
        return "применил Iron Head";
    }
}

class Stomp extends PhysicalMove{
    protected Stomp(){
```

```
        super(Type.NORMAL, 65,100);
    }
    @Override
    protected void applyOppEffects(Pokemon p){
        if (Math.random() <= 0.3) Effect.flinch(p);
        p.setMod(Stat.ACCURACY, 0);
    }
    @Override
    protected void applyOppDamage(Pokemon def, double damage){
        def.setMod(Stat.HP, (int) Math.round(damage) * 2);
    }
    @Override
    protected String describe(){
        return "применил Stomp";
    }
}

class Facade extends PhysicalMove{
    protected Facade(){
        super(Type.NORMAL, 70, 100);
    }
    @Override
    protected void applyOppDamage (Pokemon def, double damage){
        Status PokCon = def.getCondition();
        if (PokCon.equals(Status.BURN) || PokCon.equals(Status.POISON) ||
PokCon.equals(Status.PARALYZE)){
            def.setMod(Stat.HP, (int) Math.round(damage)*2);
        }
        def.setMod(Stat.HP, (int) Math.round(damage));
    }
```

```
    @Override
    protected String describe(){
        return "применил Facade";
    }
}

class Rest extends StatusMove{
    protected Rest () {
        super(Type.PSYCHIC,0,25);
    }
    Effect e = new Effect().turns(2);
    @Override
    protected void applySelfEffects(Pokemon p){
        e.sleep(p);
        p.restore();
    }
    @Override
    protected String describe() {
        return "хочет спать!";
    }
}

class SandAttack extends StatusMove{
    protected SandAttack(){
        super(Type.GROUND,0,100);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        p.setMod(Stat.ACCURACY, -1);
```

```
    }
    @Override
    protected String describe(){
        return "применил Sand Attack";
    }
}

class Barrier extends StatusMove{
    protected Barrier(){
        super(Type.PSYCHIC,0,0);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        p.setMod(Stat.DEFENSE, 1);
    }
    @Override
    protected String describe(){
        return "применил Barrier";
    }
}

class WaterGun extends SpecialMove{
    protected WaterGun(){
        super(Type.WATER, 40,100);
    }
    @Override
    protected void applyOppDamage(Pokemon def, double damage){
        def.setMod(Stat.HP, (int) Math.round(damage));
    }
```

```
    @Override
    protected String describe(){
        return "применил Water Gun";
    }
}

class IceBeam extends SpecialMove{
    protected IceBeam(){
        super(Type.ICE,90,100);
    }
    @Override
    protected void applyOppDamage(Pokemon def, double damage) {
        def.setMod(Stat.HP, (int) Math.round(damage));
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        if (Math.random() <= 0.1) Effect.freeze(p);
        p.setMod(Stat.ACCURACY, 0);
    }
    @Override
    protected String describe(){
        return "применил Ice Beam";
    }
}

class MuddyWater extends SpecialMove{
    protected MuddyWater(){
        super(Type.WATER,90,30);
    }
```

```
    @Override
    protected void applyOppDamage(Pokemon def, double damage){
        def.setMod(Stat.HP, (int) Math.round(damage));
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        if (Math.random() <= 0.3) p.setMod(Stat.ACCURACY, -1);
    }
    @Override
    protected String describe(){
        return "применил Muddy Water";
    }
}

class Bubble extends SpecialMove{
    protected Bubble(){
        super(Type.WATER,40,100);
    }
    @Override
    protected void applyOppDamage(Pokemon def, double damage){
        def.setMod(Stat.HP, (int) Math.round(damage));
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        if (Math.random() <= 0.1) p.setMod(Stat.SPEED, -1);
    }
    @Override
    protected String describe(){
        return "применил Bubble";
```

```
    }
}
```

Результат работы:

Registeel из команды желтых вступает в бой!
Azurill из команды синих вступает в бой!

Registeel применил Iron Head.
Azurill теряет 4 здоровья.

Azurill применил Ice Beam.
Registeel теряет 2 здоровья.
Azurill замерзает

Registeel применил Stomp.
Azurill теряет 8 здоровья.

Registeel применил Iron Head.
Azurill теряет 3 здоровья.
Azurill теряет сознание.

Marill из команды синих вступает в бой!

Registeel применил Stomp.
Marill теряет 6 здоровья.

Marill применил Water Gun.
Registeel теряет 7 здоровья.

Registeel применил Facade.
Marill теряет 3 здоровья.

Marill применил Muddy Water.
Registeel теряет 8 здоровья.
Registeel теряет сознание.

Eevee из команды желтых вступает в бой!

Eevee применил Sand Attack.
Eevee уменьшает точность.

Marill применил Ice Beam.
Eevee теряет 5 здоровья.

Eevee применил Facade.
Marill теряет 6 здоровья.
Marill теряет сознание.

Azumarill из команды синих вступает в бой!
Eevee применил Facade.
Azumarill теряет 6 здоровья.

Azumarill применил Ice Beam.
Eevee теряет 6 здоровья.

Eevee применил Sand Attack.
Eevee уменьшает точность.

Azumarill применил Bubble.
Eevee теряет 4 здоровья.
Eevee теряет сознание.

Glaceon из команды желтых вступает в бой!

Glaceon хочет спать!.
Glaceon засыпает

Azumarill применил Water Gun.
Glaceon теряет 4 здоровья.

Glaceon хочет спать!.
Glaceon засыпает

Azumarill применил Muddy Water.
Glaceon теряет 10 здоровья.

Glaceon применил Facade.
Azumarill теряет 5 здоровья.

Azumarill применил Water Gun.
Glaceon теряет 7 здоровья.
Glaceon теряет сознание.

В команде желтых не осталось покемонов.
Команда синих побеждает в этом бою!

Вывод:

В процессе выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования при использовании языка Java. Так же я научился подключать внешний jar-файл к своей программе.