НИУ ИТМО

факультет Программной инженерии и компьютерных технологий

Лабораторная работа №2

по дисциплине

«Программирование»

Вариант - 478

Выполнил: Студент группы P3114 Бахаруев Павел

Преподаватель: Письмак А. Е.

Санкт-Петербург

2019 г.

# Задание: (478 вар)

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](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>

# Исходный код программы

Main:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
public class Main {  
  
 public static void main(String[] args) {  
 Battle b = new Battle();  
 Cresselia p1 = new Cresselia("Чужой", 1);  
 Froakie p2 = new Froakie("Хищник", 1);  
 Pyroar p3 = new Pyroar("Рональд Макдональд", 1);  
 Frogadier p4 = new Frogadier("Кыса", 2);  
 Litleo p5 = new Litleo("Мурка", 2);  
 Greninja p6 = new Greninja("Псжшка", 3);  
 b.addAlly(p1);  
 b.addFoe(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addAlly(p5);  
 b.addFoe(p6);  
 b.go();  
  
 }  
}

Cresselia:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Cresselia extends Pokemon {  
 Cresselia(String name, int level){  
 super(name,level);  
 this.setStats(120, 70, 120, 75, 130, 85);  
 this.setType(Type.*PSYCHIC*);  
 this.setMove(new RockPolish(), new Smog(), new Flamethrower(), new RockThrow());  
 }  
}

LitLeo:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Litleo extends Pokemon {  
 Litleo(String name, int level){  
 super(name,level);  
 this.setStats(62, 50, 58, 73, 54, 72);  
 this.setType(Type.*FIRE*, Type.*NORMAL*);  
 this.setMove(new Swagger(), new ChargeBeam(), new Facade());  
 }  
}

Pyroar:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Litleo extends Litleo {  
 Litleo(String name, int level){  
 super(name,level);  
 this.setStats(62, 50, 58, 73, 54, 72);  
 this.setType(Type.*FIRE*, Type.*NORMAL*);  
 this.setMove(new Swagger(), new ChargeBeam(), new Facade());  
 }  
}

Froakie:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Froakie extends Pokemon {  
 Froakie(String name, int level){  
 super(name,level);  
 this.setStats(41, 56, 40, 62, 44, 71);  
 this.setType(Type.*WATER*);  
 this.setMove(new Facade(), new Rest());  
 }  
}

Frogadier:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Frogadier extends Froakie {  
 Frogadier (String name, int level){  
 super(name,level);  
 this.setStats(54, 63, 52, 83, 56, 97);  
 this.setType(Type.*WATER*);  
 this.setMove(new Rest(), new Twister(), new Facade());  
 }  
}

Greninja:

package com.pavel;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Greninja extends Frogadier {  
 Greninja(String name, int level){  
 super(name,level);  
 this.setStats(72, 95, 67, 103, 71, 122);  
 this.setType(Type.*WATER*, Type.*DARK*);  
 this.setMove(new Rest(), new Facade(), new Twister());  
 }  
}

ChargeBeam:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
public class ChargeBeam extends SpecialMove{  
 protected ChargeBeam(){  
 super(Type.*ELECTRIC*,50,90);  
 }  
 protected String describe() {  
 return "атакаует Charge Beam";  
 }  
}

Façade:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
class Facade extends PhysicalMove {  
 protected Facade() {  
 super(Type.*NORMAL*,70,100);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon ally) {  
 Status condition = ally.getCondition();  
 if (condition.equals(Status.*BURN*) || condition.equals(Status.*PARALYZE*) || condition.equals(Status.*POISON*)) {  
 ally.setMod(Stat.*ATTACK*, (int) ally.getStat(Stat.*ATTACK*));  
 }  
 }  
 protected boolean checkAccuracy(Pokemon ally,Pokemon foe) {  
 return true;  
 }  
 protected String describe() {  
 return "атакует Facade";  
 }  
}

Flamethrower:

package com.pavel;  
  
import ru.ifmo.se.pokemon.\*;  
public class Flamethrower extends SpecialMove{  
 protected Flamethrower(){  
 super(Type.*FIRE*,95,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon foe) {  
 if (Math.*random*()<=0.1){  
 Effect.*burn*(foe);  
 }  
 }  
 protected String describe() {  
 return "атакует Flamethrower";  
 }  
}

Rest:

package com.pavel;  
  
import ru.ifmo.se.pokemon.\*;  
class Rest extends StatusMove{  
 protected Rest() {  
 super(Type.*PSYCHIC*,0,100);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon ally){  
 Effect e = new Effect().condition(Status.*SLEEP*).turns(2);  
 ally.restore();  
 ally.addEffect(e);  
 }  
 protected String describe() {  
 return "атакует Rest";  
 }  
}

RockPolish:

package com.pavel;  
  
import ru.ifmo.se.pokemon.\*;  
class Rest extends StatusMove{  
 protected Rest() {  
 super(Type.*PSYCHIC*,0,100);  
 }  
 @Override  
 protected void applySelfEffects(Pokemon ally){  
 Effect e = new Effect().condition(Status.*SLEEP*).turns(2);  
 ally.restore();  
 ally.addEffect(e);  
 }  
 protected String describe() {  
 return "атакует Rest";  
 }  
}

RockThrow:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
public class RockThrow extends PhysicalMove{  
 protected RockThrow(){  
 super(Type.*ROCK*,50,90);  
 }  
 protected String describe() {  
 return "атакует Rock Throw";  
 }  
}

Smog:

package com.pavel;  
  
import ru.ifmo.se.pokemon.\*;  
public class Smog extends SpecialMove{  
 protected Smog(){  
 super(Type.*POISON*,20,70);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon foe) {  
 if (Math.*random*() <= 0.4){  
 Effect.*poison*(foe);  
 }  
 }  
 protected String describe() {  
 return "атакует Smog";  
 }  
}

Swagger:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
class Swagger extends StatusMove {  
 protected Swagger() {  
 super(Type.*NORMAL*,0,85);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon foe) {  
 foe.setMod(Stat.*ATTACK*, 2);  
 Effect.*confuse*(foe);  
 }  
 protected String describe() {  
 return "атакует Swagger";  
 }  
}

SweetKiss:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
public class SweetKiss extends StatusMove{  
 protected SweetKiss(){  
 super(Type.*NORMAL*,0,75);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon foe) {  
 Effect.*confuse*(foe);  
 }  
 protected String describe() {  
 return "атакует Rock Polish";  
 }  
}

Twister:

package com.pavel;  
import ru.ifmo.se.pokemon.\*;  
public class Twister extends SpecialMove{  
 protected Twister(){  
 super(Type.*DRAGON*,40,100);  
 }  
  
 protected String describe() {  
 return "атакует Twister";  
 }  
}

# Результат работы программы

# Cresselia Чужой из команды белых вступает в бой!

# Froakie Хищник из команды красных вступает в бой!

# Cresselia Чужой атакует Smog.

# Froakie Хищник теряет 4 здоровья.

# Froakie Хищник отравлен

# Froakie Хищник атакует Rest.

# Froakie Хищник засыпает

# Cresselia Чужой атакует Smog.

# Критический удар!

# Froakie Хищник теряет 8 здоровья.

# Froakie Хищник атакует Facade.

# Cresselia Чужой теряет 4 здоровья.

# Cresselia Чужой атакует Rock Throw.

# Froakie Хищник теряет 4 здоровья.

# Froakie Хищник теряет сознание.

# Frogadier Кыса из команды красных вступает в бой!

# Frogadier Кыса атакует Rest.

# Frogadier Кыса засыпает

# Cresselia Чужой атакует Flamethrower.

# Frogadier Кыса теряет 3 здоровья.

# Frogadier Кыса атакует Rest.

# Frogadier Кыса засыпает

# Cresselia Чужой атакует Smog.

# Frogadier Кыса теряет 4 здоровья.

# Frogadier Кыса атакует Facade.

# Cresselia Чужой теряет 6 здоровья.

# Cresselia Чужой атакует Flamethrower.

# Frogadier Кыса теряет 2 здоровья.

# Frogadier Кыса атакует Rest.

# Frogadier Кыса засыпает

# Cresselia Чужой атакует Flamethrower.

# Frogadier Кыса теряет 3 здоровья.

# Frogadier Кыса атакует Facade.

# Cresselia Чужой теряет 6 здоровья.

# Cresselia Чужой теряет сознание.

# Pyroar Рональд Макдональд из команды белых вступает в бой!

# Frogadier Кыса атакует Rest.

# Frogadier Кыса засыпает

# Pyroar Рональд Макдональд атакует Swagger.

# Frogadier Кыса увеличивает атаку.

# Frogadier Кыса растерянно попадает по себе.

# Frogadier Кыса теряет 6 здоровья.

# Pyroar Рональд Макдональд атакует Swagger.

# Frogadier Кыса увеличивает атаку.

# Frogadier Кыса атакует Facade.

# Pyroar Рональд Макдональд теряет 7 здоровья.

# Pyroar Рональд Макдональд атакует Rock Polish.

# Frogadier Кыса растерянно попадает по себе.

# Frogadier Кыса теряет 4 здоровья.

# Pyroar Рональд Макдональд атакует Rock Polish.

# Frogadier Кыса атакует Facade.

# Pyroar Рональд Макдональд теряет 8 здоровья.

# Pyroar Рональд Макдональд теряет сознание.

# Litleo Мурка из команды белых вступает в бой!

# Frogadier Кыса атакует Twister.

# Litleo Мурка теряет 5 здоровья.

# Litleo Мурка атакует Swagger.

# Frogadier Кыса увеличивает атаку.

# Frogadier Кыса атакует Rest.

# Frogadier Кыса засыпает

# Litleo Мурка атакаует Charge Beam.

# Frogadier Кыса теряет 8 здоровья.

# Frogadier Кыса растерянно попадает по себе.

# Frogadier Кыса теряет 3 здоровья.

# Litleo Мурка атакаует Charge Beam.

# Frogadier Кыса теряет 7 здоровья.

# Frogadier Кыса теряет сознание.

# Greninja Псжшка из команды красных вступает в бой!

# Greninja Псжшка атакует Rest.

# Litleo Мурка атакует Facade.

# Greninja Псжшка теряет 8 здоровья.

# Greninja Псжшка атакует Rest.

# Litleo Мурка атакует Swagger.

# Greninja Псжшка увеличивает атаку.

# Greninja Псжшка атакует Facade.

# Litleo Мурка теряет 6 здоровья.

# Litleo Мурка атакует Facade.

# Greninja Псжшка теряет 6 здоровья.

# Greninja Псжшка атакует Rest.

# Litleo Мурка атакаует Charge Beam.

# Greninja Псжшка теряет 7 здоровья.

# Greninja Псжшка теряет сознание.

# В команде красных не осталось покемонов.

# Команда белых побеждает в этом бою!

# Process finished with exit code 0

# Вывод

В ходе выполнения данной лабораторной работы я изучил основные принципы ООП, конструкторы, подключение внешних jar архивов, научился составлять UML диаграммы. Я сделал большой шаг в изучении языка Java и программирования в целом