Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по программированию

Вариант № 87442

Выполнил:

Аринкин Андрей

Группа: P3113

Принял:

Райла Мартин

???

?

г. Санкт-Петербург

2021 г.

# 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.



# Диаграмма классов реализованной объектной модели

# Исходный код программы

import pokemons.\*;  
import ru.ifmo.se.pokemon.Battle;  
  
public class Main {  
 public static void main(String args[]) {  
 Battle warcraft = new Battle();  
  
 Registeel p1 = new Registeel("Робокоп", 1);  
 Shuppet p2 = new Shuppet("Каспер", 1);  
 Banette p3 = new Banette("Кошак", 1);  
 Poliwag p4 = new Poliwag("Спиралька", 1);  
 Poliwhirl p5 = new Poliwhirl("Боксер", 1);  
 Politoed p6 = new Politoed("Жаба", 1);  
  
 warcraft.addAlly(p1);  
 warcraft.addAlly(p2);  
 warcraft.addAlly(p3);  
  
 warcraft.addFoe(p4);  
 warcraft.addFoe(p5);  
 warcraft.addFoe(p6);  
  
 warcraft.go();  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Amnesia extends StatusMove{  
 public Amnesia() {  
 super(Type.*PSYCHIC*,0,0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon pm) {  
 pm.setMod(Stat.*SPECIAL\_DEFENSE*, +2);  
 }  
  
 @Override  
 protected String describe(){  
 return "Использует Amnesia";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Confide extends StatusMove{  
 public Confide(){  
 super(Type.*NORMAL*, 0, 0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon pm) {  
 pm.setMod(Stat.*SPECIAL\_ATTACK*, -1);  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Confide";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DoubleSlap extends PhysicalMove {  
 public DoubleSlap() {  
 super(Type.*NORMAL*, 15, 85);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 double n, rndm = Math.*random*();  
 if(rndm < 0.125){  
 if(Math.*random*() < 0.5)n = 4;  
 else n = 5;  
 }  
 else if(rndm >= 0.125 && rndm < 0.5){  
 if(Math.*random*() < 0.5)n = 2;  
 else n = 3;  
 }  
 else n = 1;  
 damage \*= n;  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Double Slap";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DoubleTeam extends StatusMove {  
 public DoubleTeam() {  
 super(Type.*NORMAL*, 0, 0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon pm) {  
 pm.setMod(Stat.*EVASION*, +1);  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Double Team";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Facade extends PhysicalMove {  
 public Facade() {  
 super(Type.*NORMAL*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 Status pm\_stat = pm.getCondition();  
 if (pm\_stat == Status.*BURN* || pm\_stat == Status.*POISON* || pm\_stat == Status.*PARALYZE*) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage \* 2));  
 }  
 else {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Facade";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class FocusBlast extends SpecialMove {  
 public FocusBlast() {  
 super(Type.*FIGHTING*, 120, 70);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
  
 if(Math.*random*() <= 0.1){  
 pm.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Focus Blast";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class MetalClaw extends PhysicalMove {  
 public MetalClaw() {  
 super(Type.*STEEL*, 50, 95);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon pm) {  
 pm.setMod(Stat.*ATTACK*, +1);  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Metal Claw";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Psychic extends SpecialMove {  
 public Psychic() {  
 super(Type.*PSYCHIC*, 90, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
  
 if(Math.*random*() <= 0.1){  
 pm.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Psychic";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Scald extends SpecialMove{  
 public Scald(){  
 super(Type.*WATER*, 80, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
  
 if(Math.*random*() <= 0.3){  
 Effect.*burn*(pm);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Scald";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ShadowClaw extends SpecialMove{  
 public ShadowClaw(){  
 super(Type.*GHOST*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
   
 }  
  
 @Override  
 protected String describe() {  
 return "использует Shadow Claw";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Thunderbolt extends SpecialMove{  
 public Thunderbolt(){  
 super(Type.*ELECTRIC*, 90, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 pm.setMod(Stat.*HP*, (int)Math.*round*(damage));  
  
 if(Math.*random*() <= 0.1){  
 Effect.*paralyze*(pm);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Thunderbolt";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class WakeUpSlap extends PhysicalMove {  
 public WakeUpSlap() {  
 super(Type.*FIGHTING*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon pm, double damage) {  
 Status pm\_stat = pm.getCondition();  
 if (pm\_stat == Status.*SLEEP*) {  
 Effect e = new Effect().condition(Status.*NORMAL*);  
 pm.setCondition(e);  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage \* 2));  
 }  
 else {  
 pm.setMod(Stat.*HP*, (int) Math.*round*(damage));  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "использует Wake-Up Slap";  
 }  
}

package pokemons;  
  
import moves.\*;  
  
public class Banette extends Shuppet {  
 public Banette(String name, int level) {  
 super(name, level);  
 double hp = 64,  
 attack = 115,  
 defense = 65,  
 specialAttack = 83,  
 specialDefense = 63,  
 speed = 65;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 addMove(new ShadowClaw());  
 }  
}

package pokemons;  
  
import moves.\*;  
  
public class Politoed extends Poliwhirl {  
 public Politoed(String name, int level) {  
 super(name, level);  
 double hp = 90,  
 attack = 75,  
 defense = 75,  
 specialAttack = 90,  
 specialDefense = 100,  
 speed = 70;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 addMove(new DoubleSlap());  
 }  
}

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Poliwag extends Pokemon {  
 public Poliwag(String name, int level) {  
 super(name, level);  
 setType(Type.*WATER*);  
 double hp = 40,  
 attack = 50,  
 defense = 40,  
 specialAttack = 40,  
 specialDefense = 40,  
 speed = 90;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 setMove(new Scald(), new Facade());  
 }  
}

package pokemons;  
  
import moves.\*;  
  
public class Poliwhirl extends Poliwag {  
 public Poliwhirl(String name, int level) {  
 super(name, level);  
 double hp = 65,  
 attack = 65,  
 defense = 65,  
 specialAttack = 50,  
 specialDefense = 50,  
 speed = 90;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 addMove(new WakeUpSlap());  
 }  
}

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Registeel extends Pokemon {  
 public Registeel(String name, int level) {  
 super(name, level);  
 setType(Type.*STEEL*);  
 double hp = 80,  
 attack = 75,  
 defense = 150,  
 specialAttack = 75,  
 specialDefense = 150,  
 speed = 50;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 setMove(new DoubleTeam(), new MetalClaw(), new FocusBlast(), new Amnesia());  
 }  
}

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Shuppet extends Pokemon {  
 public Shuppet(String name, int level) {  
 super(name, level);  
 setType(Type.*GHOST*);  
 double hp = 44,  
 attack = 75,  
 defense = 35,  
 specialAttack = 63,  
 specialDefense = 33,  
 speed = 45;  
 setStats(hp, attack, defense, specialAttack, specialDefense, speed);  
 setMove(new Psychic(), new Thunderbolt(), new Confide());  
 }  
}

# Результат работы программы

# Registeel Робокоп из команды фиолетовых вступает в бой!

# Poliwag Спиралька из команды полосатых вступает в бой!

# Poliwag Спиралька использует Facade.

# Registeel Робокоп теряет 2 здоровья.

# Registeel Робокоп использует Metal Claw.

# Poliwag Спиралька теряет 3 здоровья.

# Registeel Робокоп увеличивает атаку.

# Poliwag Спиралька использует Scald.

# Registeel Робокоп теряет 6 здоровья.

# Registeel Робокоп использует Metal Claw.

# Poliwag Спиралька теряет 3 здоровья.

# Registeel Робокоп увеличивает атаку.

# Poliwag Спиралька использует Facade.

# Registeel Робокоп теряет 2 здоровья.

# Registeel Робокоп промахивается

# Poliwag Спиралька использует Facade.

# Registeel Робокоп теряет 2 здоровья.

# Registeel Робокоп использует Metal Claw.

# Poliwag Спиралька теряет 2 здоровья.

# Registeel Робокоп увеличивает атаку.

# Poliwag Спиралька использует Facade.

# Registeel Робокоп теряет 2 здоровья.

# Registeel Робокоп теряет сознание.

# Shuppet Каспер из команды фиолетовых вступает в бой!

# Poliwag Спиралька использует Facade.

# Shuppet Каспер теряет 1 здоровья.

# Shuppet Каспер не замечает воздействие типа NORMAL

# Shuppet Каспер использует Thunderbolt.

# Poliwag Спиралька теряет 10 здоровья.

# Poliwag Спиралька теряет сознание.

# Poliwhirl Боксер из команды полосатых вступает в бой!

# Poliwhirl Боксер использует Wake-Up Slap.

# Shuppet Каспер теряет 1 здоровья.

# Shuppet Каспер не замечает воздействие типа FIGHTING

# Shuppet Каспер использует Psychic.

# Poliwhirl Боксер теряет 6 здоровья.

# Poliwhirl Боксер использует Wake-Up Slap.

# Shuppet Каспер теряет 1 здоровья.

# Shuppet Каспер не замечает воздействие типа FIGHTING

# Shuppet Каспер промахивается

# Poliwhirl Боксер использует Scald.

# Критический удар!

# Shuppet Каспер теряет 15 здоровья.

# Shuppet Каспер воспламеняется

# Shuppet Каспер теряет сознание.

# Banette Кошак из команды фиолетовых вступает в бой!

# Poliwhirl Боксер использует Wake-Up Slap.

# Banette Кошак теряет 1 здоровья.

# Banette Кошак не замечает воздействие типа FIGHTING

# Banette Кошак использует Shadow Claw.

# Poliwhirl Боксер теряет 6 здоровья.

# Poliwhirl Боксер использует Scald.

# Banette Кошак теряет 8 здоровья.

# Banette Кошак воспламеняется

# Banette Кошак использует Thunderbolt.

# Poliwhirl Боксер теряет 11 здоровья.

# Poliwhirl Боксер теряет сознание.

# Politoed Жаба из команды полосатых вступает в бой!

# Politoed Жаба использует Scald.

# Banette Кошак теряет 8 здоровья.

# Banette Кошак теряет сознание.

# В команде фиолетовых не осталось покемонов.

# Команда полосатых побеждает в этом бою!

# Process finished with exit code 0

# Вывод

Выполнив лабораторную работу, я ознакомился с основными объектно-ориентировочного программирования, работой с классами и объектами, сторонними библиотеками и подключением их к своей программе.