САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ

ИССЛЕДРОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,

МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Программирование»

**Отчет**

По лабораторной работе №2

Вариант 441

Студент

*Митрофанов Егор Юрьевич*

*P3114*

Преподаватель

*Письмак Алексей Евгеньевич*

Санкт-Петербург, 2019 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)

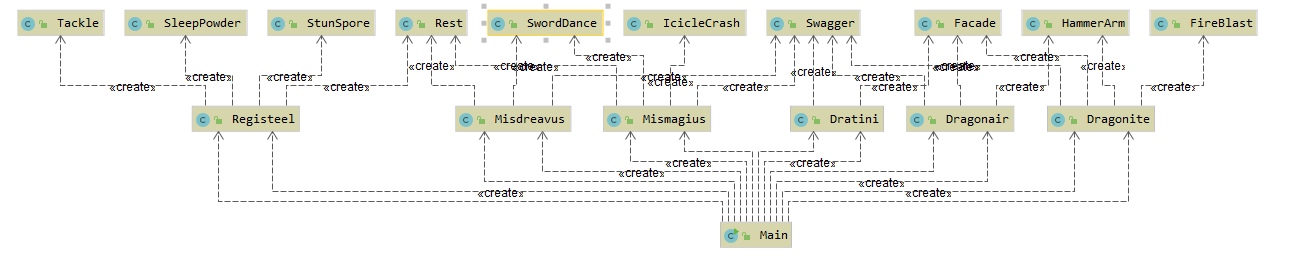
Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

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.

Покемоны:



UML-диаграмма классов

Код программы:

*Атаки:*

**package** info.moves;  
**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 бьет"**;  
 **else return "бьет"**;  
 }  
}

**public class** FireBlast **extends** SpecialMove{  
 **public** FireBlast(){  
 **super**(Type.***FIRE***, 120,85);  
 }  
 **private boolean flag**;  
 @Override  
 **public void** applyOppEffects(Pokemon poc) {  
 **if** (Math.*random*() <= 0.3) {  
 **flag** = **true**;  
 Effect.*burn*(poc);  
 }  
 }  
 @Override  
 **protected** String describe(){  
 **if**(**flag**) **return "дамажит и поджигает"**;  
 **else return "дамажит"**;  
 }  
}

**public class** HammerArm **extends** PhysicalMove{  
 **public** HammerArm(){  
 **super**(Type.***NORMAL***, 100, 90);  
 }  
 @Override  
 **public void** applySelfEffects (Pokemon p){  
 p.setMod(Stat.***SPEED***,-1);  
 }  
 @Override  
 **protected** String describe(){  
 **return "кидает молот"**;  
 }  
}

**public class** IcicleCrash **extends** PhysicalMove{  
 **public** IcicleCrash(){  
 **super**(Type.***NORMAL***, 85,90);  
 }  
 @Override  
 **public void** applyOppEffects(Pokemon p) {  
 **if** (Math.*random*() <= 0.3){  
 Effect.*flinch*(p);  
 }  
 }  
 @Override  
 **protected** String describe(){  
 **return "морозит"**;  
 }  
}

**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 "адыхает"**;  
 }  
}

**public class** SleepPowder **extends** SpecialMove {  
 **public** SleepPowder(){  
 **super**(Type.***GRASS***,0 ,75);  
 }  
 @Override  
 **public void** applyOppEffects(Pokemon p) {  
 Effect.*sleep*(p);  
 }  
 @Override  
 **protected** String describe() {  
 **return "усыпляет"**;  
 }  
}

**public class** StunSpore **extends** SpecialMove{  
 **public** StunSpore() {  
 **super**(Type.***GRASS***, 0, 75);  
 }  
 @Override  
 **public void** applyOppEffects(Pokemon p) {  
 **if** (p.hasType(Type.***ELECTRIC***) == **false** ){  
 Effect.*paralyze*(p);}  
 }  
 @Override  
 **protected** String describe(){**return "парализует"**;}  
}

**public class** Swagger **extends** SpecialMove{  
 **public** Swagger(){  
 **super**(Type.***NORMAL***,0,90);  
 }  
 @Override  
 **protected** String describe(){  
 **return "флексит"**;  
 }  
 @Override  
 **protected void** applyOppEffects(Pokemon opp){  
 opp.confuse();  
 opp.setMod(Stat.***ATTACK***,2);  
 }  
}

**public class** SwordDance **extends** SpecialMove{  
 **public** SwordDance(){  
 **super**(Type.***NORMAL***,0,95);  
 }  
 @Override  
 **public void** applySelfEffects(Pokemon p) {  
 p.setMod(Stat.***ATTACK***,+2);  
 }  
  
 @Override  
 **protected** String describe(){  
 **return "усилил атаку"**;  
 }  
}

**public class** Tackle **extends** PhysicalMove{  
 **public** Tackle(){  
 **super**(Type.***NORMAL***,35,95);  
 }  
 @Override  
 **protected** String describe(){  
 **return "толкнул"**;  
 }  
}

*Покемоны:*

**package** info.pockemons;  
**import** info.moves.\*;  
**import** ru.ifmo.se.pokemon.\*;  
  
**public class** Dragonair **extends** Pokemon {  
 **public** Dragonair(String name, **int** level){  
 **super**(name,level);  
 setStats(61,84,65,70,70,70);  
 setType(Type.***DRAGON***);  
 setMove(**new** Swagger(), **new** Facade(), **new** HammerArm());  
 }  
}

**public class** Dragonite **extends** Pokemon {  
 **public** Dragonite(String name, **int** level){  
 **super**(name,level);  
 setStats(91,134,95,100,100,80);  
 setType(Type.***DRAGON***, Type.***FLYING***);  
 setMove(**new** Swagger(), **new** Facade(), **new** HammerArm(), **new** FireBlast());  
 }  
}

**public class** Dratini **extends** Pokemon {  
 **public** Dratini(String name, **int** level){  
 **super**(name,level);  
 setStats(41,64,45,50,50,50);  
 setType(Type.***DRAGON***);  
 setMove(**new** Swagger(), **new** Facade());  
 }  
}

**public class** Misdreavus **extends** Pokemon {  
 **public** Misdreavus(String name,**int** level){  
 **super**(name,level);  
 setStats(60,60,60,85,85,85);  
 setType(Type.***GHOST***);  
 setMove(**new** Rest(), **new** SwordDance(), **new** Swagger());  
 }  
}

**public class** Mismagius **extends** Pokemon {  
 **public** Mismagius(String name, **int** level){  
 **super**(name,level);  
 setStats(60,60,60,105,105,105);  
 setType(Type.***GHOST***);  
 setMove(**new** Rest(), **new** SwordDance(), **new** Swagger(), **new** IcicleCrash());  
 }  
}

**public class** Registeel **extends** Pokemon{  
 **public** Registeel (String name, **int** lvl){  
 **super** (name, lvl);  
 setStats(57,80,91,80,87,75);  
 setType(Type.***STEEL***);  
 setMove(**new** Tackle(), **new** SleepPowder(),**new** Rest(), **new** StunSpore());  
 }  
}

*main.java*

**import** ru.ifmo.se.pokemon.\*;  
**import** info.pockemons.\*;  
**import** info.moves.\*;  
  
**public class** Main {  
 **public static void** main(String[] args) {  
 Battle b = **new** Battle();  
 Registeel p1 = **new** Registeel(**"Круглоголовый"**,2);  
 Mismagius p2 = **new** Mismagius(**"Шляпник"**, 2);;  
 Misdreavus p3 = **new** Misdreavus(**"Медузич"**,2);  
 Dratini p4 = **new** Dratini(**"Драк"**, 2);;  
 Dragonair p6 = **new** Dragonair(**"ДракПокруче"**, 2);  
 Dragonite p5 = **new** Dragonite(**"ДракКрутой"**,2);  
  
 b.addAlly(p1);  
 b.addAlly(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addFoe(p5);  
 b.addFoe(p6);  
 b.go();  
 }  
}

*Результат работы программы:*

Registeel Круглоголовый из команды красных вступает в бой!

Dratini Драк из команды белых вступает в бой!

Registeel Круглоголовый усыпляет.

Dratini Драк теряет 1 здоровья.

Dratini Драк засыпает

Dratini Драк флексит.

Registeel Круглоголовый теряет 2 здоровья.

Registeel Круглоголовый увеличивает атаку.

Registeel Круглоголовый парализует.

Dratini Драк теряет 1 здоровья.

Dratini Драк парализован

Dratini Драк бьет.

Критический удар!

Registeel Круглоголовый растерянно попадает по себе.

Registeel Круглоголовый теряет 6 здоровья.

Dratini Драк сильно бьет.

Критический удар!

Registeel Круглоголовый усыпляет.

Dratini Драк теряет 1 здоровья.

Dratini Драк засыпает

Registeel Круглоголовый парализует.

Dratini Драк теряет 1 здоровья.

Dratini Драк парализован

Dratini Драк сильно бьет.

Критический удар!

Dratini Драк флексит.

Критический удар!

Registeel Круглоголовый теряет 3 здоровья.

Registeel Круглоголовый увеличивает атаку.

Registeel Круглоголовый парализует.

Dratini Драк теряет 1 здоровья.

Dratini Драк флексит.

Критический удар!

Registeel Круглоголовый теряет 3 здоровья.

Registeel Круглоголовый увеличивает атаку.

Registeel Круглоголовый адыхает.

Registeel Круглоголовый засыпает

Dratini Драк сильно бьет.

Критический удар!

Registeel Круглоголовый растерянно попадает по себе.

Registeel Круглоголовый теряет 6 здоровья.

Dratini Драк сильно бьет.

Критический удар!

Registeel Круглоголовый толкнул.

Dratini Драк теряет 4 здоровья.

Dratini Драк сильно бьет.

Критический удар!

Registeel Круглоголовый растерянно попадает по себе.

Registeel Круглоголовый теряет 4 здоровья.

Registeel Круглоголовый растерянно попадает по себе.

Registeel Круглоголовый теряет 5 здоровья.

Registeel Круглоголовый теряет сознание.

Mismagius Шляпник из команды красных вступает в бой!

Mismagius Шляпник морозит.

Dratini Драк теряет 4 здоровья.

Dratini Драк флексит.

Критический удар!

Mismagius Шляпник теряет 1 здоровья.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник усилил атаку.

Dratini Драк теряет 2 здоровья.

Dratini Драк теряет сознание.

Dragonite ДракКрутой из команды белых вступает в бой!

Mismagius Шляпник усилил атаку.

Dragonite ДракКрутой теряет 3 здоровья.

Dragonite ДракКрутой дамажит.

Mismagius Шляпник теряет 8 здоровья.

Mismagius Шляпник морозит.

Dragonite ДракКрутой теряет 4 здоровья.

Dragonite ДракКрутой бьет.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник флексит.

Dragonite ДракКрутой теряет 2 здоровья.

Dragonite ДракКрутой увеличивает атаку.

Dragonite ДракКрутой флексит.

Mismagius Шляпник теряет 1 здоровья.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник усилил атаку.

Dragonite ДракКрутой теряет 4 здоровья.

Dragonite ДракКрутой сильно бьет.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник флексит.

Dragonite ДракКрутой теряет 3 здоровья.

Dragonite ДракКрутой увеличивает атаку.

Dragonite ДракКрутой теряет сознание.

Dragonair ДракПокруче из команды белых вступает в бой!

Mismagius Шляпник адыхает.

Dragonair ДракПокруче флексит.

Mismagius Шляпник теряет 1 здоровья.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник морозит.

Dragonair ДракПокруче теряет 4 здоровья.

Dragonair ДракПокруче флексит.

Mismagius Шляпник теряет 1 здоровья.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник адыхает.

Dragonair ДракПокруче бьет.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник флексит.

Dragonair ДракПокруче теряет 3 здоровья.

Dragonair ДракПокруче увеличивает атаку.

Dragonair ДракПокруче флексит.

Mismagius Шляпник теряет 1 здоровья.

Mismagius Шляпник не замечает воздействие типа NORMAL

Mismagius Шляпник усилил атаку.

Критический удар!

Dragonair ДракПокруче теряет 8 здоровья.

Dragonair ДракПокруче теряет сознание.

В команде белых не осталось покемонов.

Команда красных побеждает в этом бою!

Выводы:

В процессе выполнения лабораторной работы я получил навыки использования

объектно-ориентированного подхода программирования при использовании языка Java.

Научился работать с классами, конструкторами, полями и модификаторами доступа.