САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ

ИССЛЕДОВАТЕЛЬСКИЙ

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 649375

Выполнил:

Студент:

*Казаев Максим Павлович*

*Группа : P3111*

Преподаватель:

*Харитонова Анастасия Евгеньевна*

Санкт-Петербург, 2022 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)

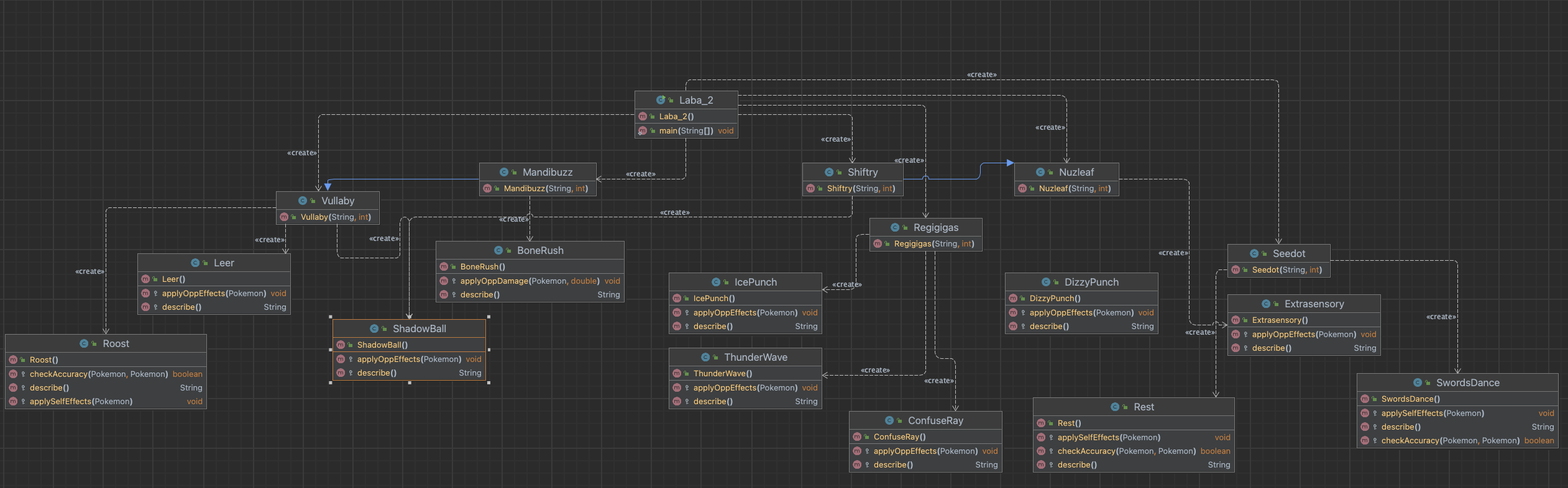
Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

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 Attack;  
import ru.ifmo.se.pokemon.\*;  
public class ConfuseRay extends StatusMove{  
 public ConfuseRay(){super(Type.*GHOST*, 0, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 pokemon.confuse();  
 if(Math.*random*() <= 0.33){  
 Effect.*confuse*(pokemon);  
 }  
 }  
  
 @Override  
 protected String describe() {return "ConfuseRay";}  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class DizzyPunch extends PhysicalMove{  
 public DizzyPunch(){super(Type.*NORMAL*, 70, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 if(Math.*random*() <= 0.2){  
 Effect.*confuse*(pokemon);  
 }  
 }  
  
 @Override  
 protected String describe() {return "DizzyPunch!";}  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class Extrasensory extends SpecialMove{  
 public Extrasensory(){super(Type.*PSYCHIC*, 80, 100);}  
  
 @Override  
 protected String describe() {return "Extrasensory";}  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 if(Math.*random*()<= 0.1){  
 Effect.*flinch*(pokemon);  
 }  
 }  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class IcePunch extends PhysicalMove{  
 public IcePunch(){super(Type.*ICE*, 75, 100);}  
 protected boolean flag = true;  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 if(pokemon.hasType(Type.*ICE*) == false){  
 Effect effect = new Effect().chance(0.1).condition(Status.*FREEZE*);  
 pokemon.addEffect(effect);}  
 else{flag = false;}  
  
  
 }  
 @Override  
 protected String describe(){  
 if(flag == true){return "Цель замороженна!";}  
 else{return "Цель не замороженна(";}  
 }  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class Leer extends StatusMove{  
 public Leer(){super(Type.*NORMAL*, 0, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 pokemon.setMod(Stat.*DEFENSE*, - 1);  
 }  
  
 @Override  
 protected String describe() {return"Бьем броню";}  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class Rest extends StatusMove{  
 public Rest(){super(Type.*PSYCHIC*, 0, 0);}  
 @Override  
 protected String describe() {return "ZzZz";};  
 @Override  
 protected boolean checkAccuracy(Pokemon att, Pokemon def){  
 return true;  
 }  
  
  
 @Override  
 protected void applySelfEffects(Pokemon pokemon) {  
 Effect effect = new Effect().turns(2).condition(Status.*SLEEP*);  
 pokemon.addEffect(effect);  
 pokemon.restore();  
 }  
  
}

import ru.ifmo.se.pokemon.\*;  
import static ru.ifmo.se.pokemon.Type.*NONE*;  
  
public class Roost extends StatusMove {  
 public Roost() {  
 super(*NONE*, 0, 0);  
 }  
  
 @Override  
 protected String describe() {  
 return "Хилимся!";  
 }  
  
 @Override  
 protected boolean checkAccuracy(Pokemon att, Pokemon def) {  
 return true;  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon pokemon) {  
 pokemon.setMod(Stat.*HP*, -2);  
  
 }  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class ShadowBall extends SpecialMove{  
 public ShadowBall(){super(Type.*GHOST*, 80, 100);}  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 if(Math.*random*()<= 0.2){  
 pokemon.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {return "Shadow Ball";}  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class SwordsDance extends StatusMove{  
 public SwordsDance(){super(Type.*NORMAL*, 0, 0);}  
  
 @Override  
 protected String describe() {return "SwordsDance";}  
 @Override  
 protected boolean checkAccuracy(Pokemon att, Pokemon def){  
 return true;  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon pokemon) {  
 pokemon.setMod(Stat.*ATTACK*, +2);  
  
 }  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class ThunderWave extends StatusMove {  
 public ThunderWave() {  
 super(Type.*ELECTRIC*, 0, 90);  
 }  
  
 protected Boolean flag = true;  
  
  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 if (pokemon.hasType(Type.*ELECTRIC*) == false) {  
 Effect.*paralyze*(pokemon);  
 if (Math.*random*() <= 0.25) {  
 Effect.*flinch*(pokemon);  
 pokemon.setMod(Stat.*SPEED*, -2);  
 }  
 } else {flag = false;}  
 }  
  
 @Override  
 protected String describe() {  
 if(flag == true){return "ThuderWave";}  
 else{return "А он неплох";}  
 }  
}

package Attack;  
import ru.ifmo.se.pokemon.\*;  
public class BoneRush extends PhysicalMove {  
 public BoneRush() {  
 super(Type.*GROUND*, 25, 90, (int) Math.*random*(), (int) (Math.*random*() \* 3 + 2));  
 }  
 @Override  
 protected void applyOppEffects(Pokemon pokemon) {  
 for(int i = (int) Math.*random*()\*3 - 1;i==0;i--){  
 if(i==2){pokemon.setMod(Stat.*HP*, -2);}  
 if(i==3){pokemon.setMod(Stat.*HP*, -3);}  
 if(i==4){pokemon.setMod(Stat.*HP*, -4);}  
 if(i==5){pokemon.setMod(Stat.*HP*, -6);}  
 }  
 }  
  
 @Override  
 protected String describe(){return "Bone Rush";}  
}

*Покемоны:*

package Pokemon;  
import Attack.\*;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Regigigas extends Pokemon {  
 public Regigigas(String name, int level){  
 super(name,level);  
 setStats(110,160,110,80,110,100);  
 setType(Type.*NORMAL*);  
 setMove(new IcePunch(), new ThunderWave(),new DizzyPunch(), new ConfuseRay());  
 }  
}

package Pokemon;  
import Attack.\*;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
 public class Vullaby extends Pokemon {  
 public Vullaby(String name, int lavel) {  
 super(name, lavel);  
 setStats(70, 55, 75, 45, 65, 60);  
 setType(Type.*DARK*, Type.*FLYING*);  
 setMove(new Roost(), new Leer(), new ShadowBall());  
 }  
}

package Pokemon;  
import Attack.BoneRush;  
import ru.ifmo.se.pokemon.Type;  
  
public class Mandibuzz extends Vullaby {  
 public Mandibuzz(String name, int level){  
 super(name,level);  
 setStats(110,65,105,55,95,80);  
 setType(Type.*DARK*, Type.*FLYING*);  
 addMove(new BoneRush());  
 }  
}

package Pokemon;  
import Attack.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Seedot extends Pokemon {  
 public Seedot(String name, int level){  
 super(name,level);  
 setStats(40,40,50,30,30,30);  
 setType(Type.*GRASS*);  
 setMove(new Rest(), new SwordsDance());  
  
 }  
}

package Pokemon;  
import Attack.Extrasensory;  
import ru.ifmo.se.pokemon.\*;  
public class Nuzleaf extends Seedot {  
 public Nuzleaf(String name, int level){  
 super(name,level);  
 setStats(70, 70, 40, 60, 40, 60);  
 setType(Type.*GRASS*, Type.*DARK*);  
 addMove(new Extrasensory());  
 }  
}

package Pokemon;  
import Attack.ShadowBall;  
import ru.ifmo.se.pokemon.\*;  
public class Shiftry extends Nuzleaf {  
 public Shiftry(String name, int level){  
 super(name,level);  
 setStats(90,100,60,90,60,80);  
 setType(Type.*DARK*, Type.*GRASS*);  
 addMove(new ShadowBall());  
 }  
}

*main.java*

import ru.ifmo.se.pokemon.\*;  
import Pokemon.\*;  
public class Laba\_2 {  
 public static void main(String[] args) {  
 Battle b = new Battle();  
 Regigigas p1 = new Regigigas("гигант", 1);  
 Vullaby p2 = new Vullaby("Волк", 1);  
 Mandibuzz p3 = new Mandibuzz("Майонез", 1);  
 Nuzleaf p4 = new Nuzleaf("Леаф", 1);  
 Seedot p5 = new Seedot("Удот", 1);  
 Shiftry p6 = new Shiftry("Шифт", 1);  
 b.addAlly(p3);  
 b.addAlly(p5);  
 b.addAlly(p4);  
 b.addFoe(p2);  
 b.addFoe(p1);  
 b.addFoe(p6);  
 b.go();  
 }  
}

***Результат работы программы:***

Mandibuzz Майонез из команды черных вступает в бой!

Vullaby Волк из команды синих вступает в бой!

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Vullaby Волк Shadow Ball.

Mandibuzz Майонез теряет 2 здоровья.

Mandibuzz Майонез уменьшает специальную защиту.

Mandibuzz Майонез Бьем броню.

Vullaby Волк уменьшает защиту.

Vullaby Волк Хилимся!.

Vullaby Волк восстанавливает 2 здоровья.

Mandibuzz Майонез Shadow Ball.

Vullaby Волк теряет 3 здоровья.

Vullaby Волк уменьшает специальную защиту.

Vullaby Волк Shadow Ball.

Mandibuzz Майонез теряет 2 здоровья.

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Vullaby Волк Хилимся!.

Vullaby Волк восстанавливает 2 здоровья.

Mandibuzz Майонез Хилимся!.

Mandibuzz Майонез восстанавливает 2 здоровья.

Vullaby Волк Shadow Ball.

Mandibuzz Майонез теряет 3 здоровья.

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Vullaby Волк Shadow Ball.

Mandibuzz Майонез теряет 2 здоровья.

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Mandibuzz Майонез Bone Rush.

Vullaby Волк теряет 1 здоровья.

Vullaby Волк не замечает воздействие типа GROUND

Vullaby Волк Shadow Ball.

Mandibuzz Майонез теряет 2 здоровья.

Mandibuzz Майонез Shadow Ball.

Vullaby Волк теряет 3 здоровья.

Vullaby Волк теряет сознание.

Regigigas гигант из команды синих вступает в бой!

Regigigas гигант Цель замороженна!.

Mandibuzz Майонез теряет 11 здоровья.

Mandibuzz Майонез теряет сознание.

Seedot Удот из команды черных вступает в бой!

Regigigas гигант ConfuseRay.

Seedot Удот SwordsDance.

Seedot Удот увеличивает атаку.

Regigigas гигант Цель замороженна!.

Seedot Удот теряет 14 здоровья.

Seedot Удот теряет сознание.

Nuzleaf Леаф из команды черных вступает в бой!

Regigigas гигант ConfuseRay.

Nuzleaf Леаф SwordsDance.

Nuzleaf Леаф увеличивает атаку.

Regigigas гигант ConfuseRay.

Nuzleaf Леаф ZzZz.

Regigigas гигант Цель замороженна!.

Nuzleaf Леаф теряет 11 здоровья.

Nuzleaf Леаф растерянно попадает по себе.

Nuzleaf Леаф теряет 3 здоровья.

Nuzleaf Леаф теряет сознание.

В команде черных не осталось покемонов.

Команда синих побеждает в этом бою!

**Вывод:**

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.