Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант №31189976

Выполнил:

Студент группы P3118

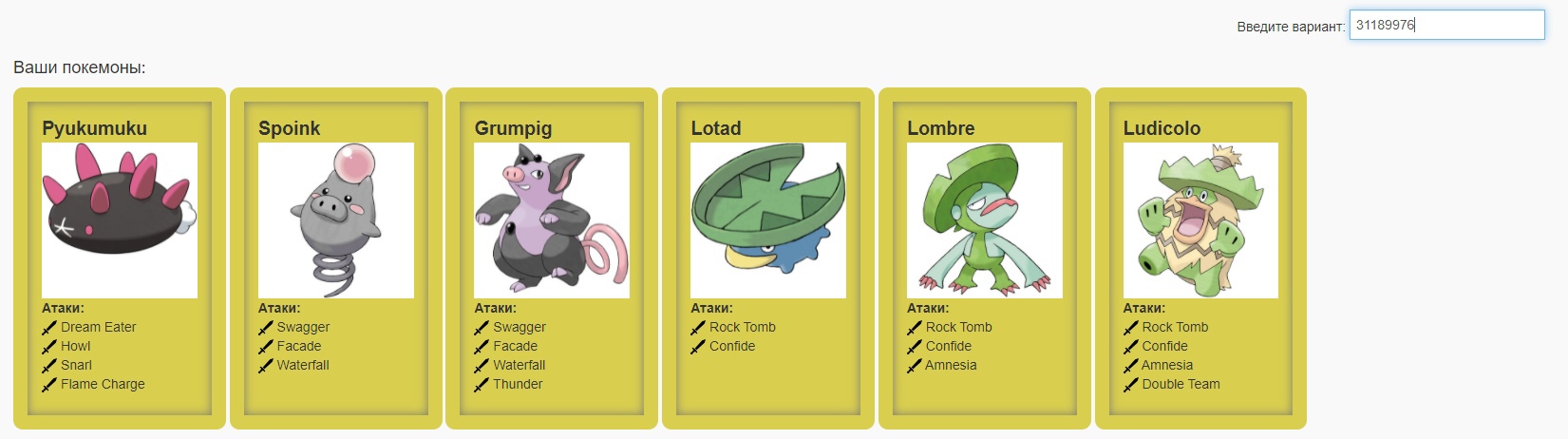
Шульга Артём Игоревич

Преподаватель:

Сорокин Роман Борисович



Санкт-Петербург, 2021

Задание:

// Main.java

package com.company;  
import ru.ifmo.se.pokemon.\*;  
  
public class Main {  
 public static void main(String[] args) {  
  
 Battle b = new Battle();  
  
 Ludicolo ludicolo = new Ludicolo("larymar",6);  
 Lombre lombre = new Lombre("bitway",3);  
 Lotad lotad = new Lotad("zhabakva",2);  
 Grumpig grumpig = new Grumpig("klimenkov",8);  
 Spoink spoink = new Spoink("pismak",8);  
 Pyukumuku pyukumuku = new Pyukumuku("artemka",1);  
  
 b.addAlly(lotad);  
 b.addAlly(grumpig);  
 b.addAlly(ludicolo);  
  
 b.addFoe(lombre);  
 b.addFoe(spoink);  
 b.addFoe(pyukumuku);  
  
 b.go();  
 }  
}

*// Ludicolo.java*

package com.company;  
  
public class Ludicolo extends Lombre{  
 Ludicolo(final String name,final int level){  
 super(name,level);  
 setStats(80,70,70,90,100,70);  
 addMove(new DoubleTeam());  
 }  
}

// Lombre.java

package com.company;  
  
public class Lombre extends Lotad{  
 Lombre(final String name,final int level){  
 super(name,level);  
 setStats(60,50,50,60,70,50);  
 addMove(new Amnesia());  
 }  
}

// Lotad.java

package com.company;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Lotad extends Pokemon {  
 Lotad(final String name,final int level){  
 super(name,level);  
 setStats(40,30,30,40,50,30);  
 addType(Type.*WATER*);  
 addType(Type.*GRASS*);  
 addMove(new RockTomb());  
 addMove(new Confide());  
 }  
}

Исходный код:

// Grumpig.java

package com.company;  
  
public class Grumpig extends Spoink{  
 Grumpig(final String name,final int level){  
 super(name,level);  
 setStats(80,45,65,90,110,80);  
 addMove(new Thunder());  
 }  
}

// Spoink.java

package com.company;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Spoink extends Pokemon {  
 Spoink(final String name,final int level) {  
 super(name,level);  
 addType(Type.*PSYCHIC*);  
 setStats(60,25,35,70,80,60);  
 addMove(new Swagger());  
 addMove(new Facade());  
 addMove(new Waterfall());  
 }  
}

// Pyukmuku.java

package com.company;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Pyukumuku extends Pokemon {  
 Pyukumuku(final String name,final int level){  
 super(name,level);  
 addType(Type.*WATER*);  
 setStats(55,60,130,30,130,5);  
 addMove(new DreamEater());  
 addMove(new Howl());  
 addMove(new Snarl());  
 addMove(new FlameCharge());  
 }  
}

// Amnesia.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Amnesia extends StatusMove {  
 Amnesia(){  
 super(Type.*PSYCHIC*,0,0);  
 }  
  
 @Override  
 protected String describe() {  
 return "страдает амнезией";  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 p.addEffect(new Effect().turns(2).stat(Stat.*SPECIAL\_DEFENSE*,(int)(Math.*random*()\*5+1)));  
 }  
}

// Confide.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Confide extends StatusMove {  
 Confide(){  
 super(Type.*NORMAL*,0,0);  
 }  
  
 @Override  
 protected String describe() {  
 return "делает людишек слабее";  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.addEffect(new Effect().turns(1).stat(Stat.*SPECIAL\_ATTACK*,-1\*(int)(Math.*random*()\*5+1)));  
 }  
}

// DoubleTeam.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DoubleTeam extends StatusMove {  
 DoubleTeam(){  
 super(Type.*NORMAL*,0,0);  
 }  
  
 @Override  
 protected String describe() {  
 return "делается более уклончивым";  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.addEffect(new Effect().turns(1).stat(Stat.*EVASION*,(int)(Math.*random*()\*5+1)));  
 }  
}

// DreameEater.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DreamEater extends SpecialMove {  
 DreamEater() {  
 super(Type.*PSYCHIC*,100,100);  
 }  
 @Override  
 protected String describe(){  
 return "начинает кошмарить";  
 }  
 @Override  
 protected boolean checkAccuracy(Pokemon att, Pokemon def) {  
 return def.getCondition() == Status.*SLEEP*;  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*HP*,-1\*(int)(p.getStat(Stat.*HP*)-p.getHP()));  
 }  
  
}

// Facade.java

package com.company;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Status;  
import ru.ifmo.se.pokemon.Type;  
  
public class Facade extends PhysicalMove {  
 Facade(){  
 super(Type.*NORMAL*,70,100);  
 }  
 @Override  
 protected String describe(){  
 return "дерётся";  
 }  
 @Override  
 protected double calcBaseDamage(Pokemon att, Pokemon def){  
 if(def.getCondition() == Status.*BURN* ||  
 def.getCondition() == Status.*PARALYZE* ||  
 def.getCondition() == Status.*POISON*)  
 return super.calcBaseDamage(att,def)\*2;  
 return super.calcBaseDamage(att,def);  
 }  
}

// FlameCharge

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class FlameCharge extends PhysicalMove {  
 FlameCharge(){  
 super(Type.*PSYCHIC*,50,100);  
 }  
 @Override  
 protected String describe(){  
 return "рвётся с огоньком";  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p){  
 p.addEffect(new Effect().turns(1).stat(Stat.*SPEED*,(int)(Math.*random*()\*5+1)));  
 }  
}

// Howl

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Howl extends StatusMove {  
 Howl() {  
 super(Type.*NORMAL*,0,0);  
 }  
 @Override  
 protected String describe(){  
 return "повыл";  
 }  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.addEffect(new Effect().turns(1).stat(Stat.*ATTACK*,(int)(Math.*random*()\*5+1)));  
 }  
}

// RockTomb.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class RockTomb extends PhysicalMove {  
 RockTomb(){  
 super(Type.*ROCK*,60,95);  
 }  
  
 @Override  
 protected String describe() {  
 return "начинает закидывать камнями";  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.addEffect(new Effect().turns(1).stat(Stat.*SPEED*,-1\*(int)(Math.*random*()\*5+1)));  
 }  
}

// Snarl.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Snarl extends SpecialMove {  
 Snarl() {  
 super(Type.*DARK*,55,95);  
 }  
 @Override  
 protected String describe(){  
 return "рычит";  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.addEffect(new Effect().turns(1).stat(Stat.*ATTACK*,-1\*(int)(Math.*random*()\*5+1)));  
 }  
}

// Swagger.java

package com.company;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 Swagger(){  
 super(Type.*NORMAL*,0,85);  
 }  
 @Override  
 protected String describe(){  
 return "начинает конфузить";  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.confuse();  
 p.addEffect(new Effect().turns(2).stat(Stat.*ATTACK*,(int)(Math.*random*()\*5+1)));  
 }  
}

// Thunder.java

package com.company;  
  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class Thunder extends SpecialMove {  
 Thunder(){  
 super(Type.*ELECTRIC*,110,70);  
 }  
  
 @Override  
 protected String describe() {  
 return "метает молнии";  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if((int)(Math.*random*()+100) <= 30)  
 Effect.*paralyze*(p);  
 }  
}

// Waterfall.java

package com.company;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class Waterfall extends PhysicalMove {  
 Waterfall(){  
 super(Type.*PSYCHIC*,80,100);  
 }  
 @Override  
 protected String describe(){  
 return "льёт водичку";  
 }  
}

Результат работы:

Результат 1.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Вывод:

Во время выполнения данной лабораторной работы я познакомился с основами Объектно-ориентированного программирования, изучил его применение на языке программирования Java. Также я узнал об основных парадигмах ООП, в особенности о наследовании. Эти знания пригодятся мне при разработке собственных проектов.