Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.01 - Информатика и вычислительная техника, Компьютерные системы и технологии*

*Дисциплина «Программирование»*

**Отчет**

**По лабораторной работе №2**

**Вариант №444333**

Выполнил:

Дворкин Борис Александрович

Группа: Р3131

Преподаватель:

Николаев Владимир Вячеславович

Г. Санкт-Петербург, 2022 г.

**Оглавление**

[Задание 2](#_Toc114643031)

[Основные этапы вычисления 3](#_Toc114643032)

[Программное решение: 6](#_Toc114643033)

[Результат выполнения программы: 6](#_Toc114643034)

[Заключение 7](#_Toc114643035)

[Список литературы. 8](#_Toc114643036)

# Задание

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:

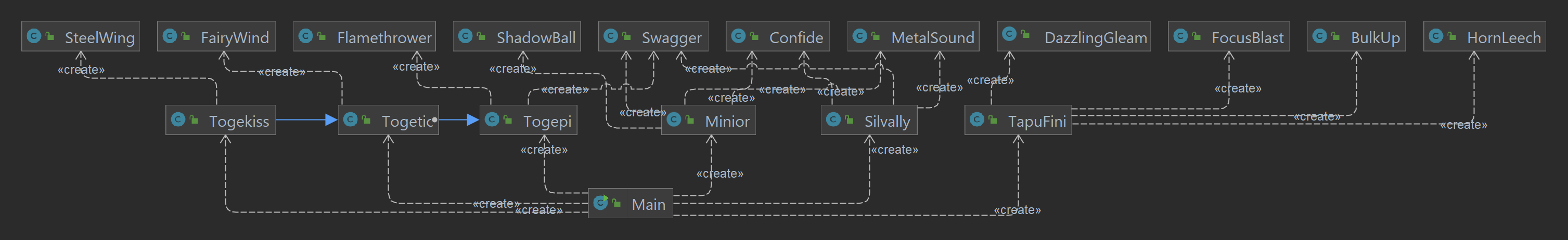


Рисунок 1

# Код программы:

# Покемоны:

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Minior extends Pokemon {  
 public Minior(String name, int level) {  
 super(name, level);  
 setStats(60, 60, 100, 60, 100, 60);  
 setType(Type.*ROCK*, Type.*FLYING*);  
 setMove(new Confide(), new Swagger(), new MetalSound(), new ShadowBall());  
 }  
}

package pokemons;  
  
import ru.ifmo.se.pokemon.\*;  
import moves.\*;  
  
public class Silvally extends Pokemon {  
 public Silvally(String name, int level) {  
 super(name, level);  
 setStats(95, 95, 95, 95,95, 95);  
 setType(Type.*NORMAL*);  
 setMove(new Confide(), new Swagger(), new MetalSound());  
 }  
}

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class TapuFini extends Pokemon {  
 public TapuFini(String name, int level) {  
 super(name, level);  
 setStats(70, 75, 115, 95, 130, 85);  
 setType(Type.*WATER*, Type.*FAIRY*);  
 setMove(new DazzlingGleam(), new HornLeech(), new FocusBlast(), new BulkUp());  
 }  
}

package pokemons;  
  
import moves.\*;  
  
public class Togekiss extends Togetic{  
 public Togekiss(String name, int level) {  
 super(name, level);  
 setStats(85, 50, 95, 120, 115, 80);  
 addMove(new SteelWing());  
 }  
   
}

package pokemons;  
  
import moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Togepi extends Pokemon{  
 public Togepi(String name, int level) {  
 super(name, level);  
 setStats(35, 20, 65, 40, 65, 20);  
 setType(Type.*FAIRY*);  
 setMove(new Swagger(), new Flamethrower());  
 }  
}

package pokemons;  
  
import moves.\*;  
  
public class Togetic extends Togepi {  
 public Togetic(String name, int level) {  
 super(name, level);  
 setStats(55, 40, 85, 80, 105, 40);  
 addMove(new FairyWind());  
 }  
}

# Атаки:

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class BulkUp extends StatusMove {  
 public BulkUp() {  
 super(Type.*FIGHTING*, 0, 0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*ATTACK*, 1);  
 p.setMod(Stat.*DEFENSE*, 1);  
 }  
  
 @Override  
 protected String describe(){  
 return "is using Bulk Up";  
 }  
}

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

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class DazzlingGleam extends SpecialMove {  
 public DazzlingGleam() {  
 super(Type.*FAIRY*, 80, 100);  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Dazzling Gleam";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class FairyWind extends SpecialMove {  
 public FairyWind() {  
 super(Type.*FAIRY*, 40, 100);  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Fairy Wind";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Flamethrower extends SpecialMove {  
 public Flamethrower() {  
 super(Type.FIRE, 90, 100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.random() < 0.1) {  
 Effect.burn(p);  
 }  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class FocusBlast extends SpecialMove {  
 public FocusBlast() {  
 super(Type.FIGHTING, 102, 70);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.random() < 0.1) {  
 p.setMod(Stat.SPECIAL\_DEFENSE, -1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Focus Blast";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class HornLeech extends PhysicalMove {  
 public HornLeech() {  
 super(Type.GRASS, 75, 100);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.HP, 50);  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Horn Leech";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class MetalSound extends StatusMove {  
 public MetalSound() {  
 super(Type.STEEL, 0, 85);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.SPECIAL\_DEFENSE, -2);  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Metal Sound";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class SteelWing extends PhysicalMove {  
 public SteelWing() {  
 super(Type.*STEEL*, 70, 90);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 if(Math.*random*() < 0.1){  
 p.setMod(Stat.*DEFENSE*, 1);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Steel Wing";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ShadowBall extends SpecialMove {  
 public ShadowBall() {  
 super(Type.*GHOST*, 80, 100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() < 0.2) {  
 p.setMod(Stat.*SPECIAL\_DEFENSE*, 2);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Shadow Ball";  
 }  
}

package moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Swagger extends StatusMove {  
 public Swagger() {  
 super(Type.NORMAL, 0, 85);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.confuse();  
 p.setMod(Stat.ATTACK, 2);  
 }  
  
 @Override  
 protected String describe() {  
 return "is using Swagger";  
 }  
}

# Главный класс (Main.java):

import pokemons.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Battle b = new Battle ();  
 Minior p1 = new Minior("rtx3090", 1);  
 TapuFini p2 = new TapuFini("ПчелоВеЧищЕ", 10);  
 Silvally p3 = new Silvally("АлександрⅠ", 10);  
 Togekiss p4 = new Togekiss("Клименков", 10);  
 Togetic p5 = new Togetic("Чернорабочий", 10);  
 Togepi p6 = new Togepi("Василий", 10);  
 b.addAlly(p1);  
 b.addAlly(p2);  
 b.addAlly(p3);  
 b.addFoe(p4);  
 b.addFoe(p5);  
 b.addFoe(p6);  
 b.go();  
 }  
}

# Результат работы программы:

Minior rtx3090 из команды фиолетовых вступает в бой!

Togekiss Клименков из команды полосатых вступает в бой!

Togekiss Клименков is using Steel Wing.

Minior rtx3090 теряет 7 здоровья.

Minior rtx3090 is using Shadow Ball.

Togekiss Клименков теряет 7 здоровья.

Togekiss Клименков атакует.

Minior rtx3090 теряет 4 здоровья.

Minior rtx3090 is using Shadow Ball.

Togekiss Клименков теряет 4 здоровья.

Togekiss Клименков is using Swagger.

Minior rtx3090 увеличивает атаку.

Minior rtx3090 промахивается

Togekiss Клименков is using Steel Wing.

Minior rtx3090 теряет 8 здоровья.

Minior rtx3090 растерянно попадает по себе.

Minior rtx3090 теряет 4 здоровья.

Minior rtx3090 теряет сознание.

TapuFini ПчелоВеЧищЕ из команды фиолетовых вступает в бой!

TapuFini ПчелоВеЧищЕ is using Dazzling Gleam.

Togekiss Клименков теряет 9 здоровья.

Togekiss Клименков is using Steel Wing.

TapuFini ПчелоВеЧищЕ теряет 2 здоровья.

TapuFini ПчелоВеЧищЕ промахивается

Togekiss Клименков is using Fairy Wind.

TapuFini ПчелоВеЧищЕ теряет 6 здоровья.

TapuFini ПчелоВеЧищЕ is using Focus Blast.

Togekiss Клименков теряет 4 здоровья.

Togekiss Клименков атакует.

TapuFini ПчелоВеЧищЕ теряет 2 здоровья.

TapuFini ПчелоВеЧищЕ is using Focus Blast.

Togekiss Клименков теряет 4 здоровья.

Togekiss Клименков теряет сознание.

Togetic Чернорабочий из команды полосатых вступает в бой!

TapuFini ПчелоВеЧищЕ is using Dazzling Gleam.

Togetic Чернорабочий теряет 9 здоровья.

Togetic Чернорабочий is using Fairy Wind.

TapuFini ПчелоВеЧищЕ теряет 7 здоровья.

TapuFini ПчелоВеЧищЕ is using Dazzling Gleam.

Togetic Чернорабочий теряет 7 здоровья.

Togetic Чернорабочий атакует.

TapuFini ПчелоВеЧищЕ теряет 2 здоровья.

TapuFini ПчелоВеЧищЕ промахивается

Togetic Чернорабочий is using Swagger.

TapuFini ПчелоВеЧищЕ увеличивает атаку.

TapuFini ПчелоВеЧищЕ промахивается

Togetic Чернорабочий is using Swagger.

TapuFini ПчелоВеЧищЕ увеличивает атаку.

TapuFini ПчелоВеЧищЕ is using Focus Blast.

Togetic Чернорабочий теряет 4 здоровья.

Togetic Чернорабочий is using Swagger.

TapuFini ПчелоВеЧищЕ увеличивает атаку.

TapuFini ПчелоВеЧищЕ растерянно попадает по себе.

TapuFini ПчелоВеЧищЕ теряет 8 здоровья.

TapuFini ПчелоВеЧищЕ теряет сознание.

Silvally АлександрⅠ из команды фиолетовых вступает в бой!

Silvally АлександрⅠ is using Metal Sound.

Togetic Чернорабочий уменьшает специальную защиту.

Togetic Чернорабочий is using Swagger.

Silvally АлександрⅠ увеличивает атаку.

Silvally АлександрⅠ is using Swagger.

Togetic Чернорабочий увеличивает атаку.

Togetic Чернорабочий is using Fairy Wind.

Silvally АлександрⅠ теряет 7 здоровья.

Silvally АлександрⅠ is using Swagger.

Togetic Чернорабочий увеличивает атаку.

Togetic Чернорабочий растерянно попадает по себе.

Togetic Чернорабочий теряет 5 здоровья.

Togetic Чернорабочий теряет сознание.

Togepi Василий из команды полосатых вступает в бой!

Silvally АлександрⅠ растерянно попадает по себе.

Silvally АлександрⅠ теряет 6 здоровья.

Togepi Василий атакует.

Silvally АлександрⅠ теряет 6 здоровья.

Silvally АлександрⅠ воспламеняется

Silvally АлександрⅠ теряет 1 здоровья.

Silvally АлександрⅠ is using Swagger.

Togepi Василий увеличивает атаку.

Togepi Василий is using Swagger.

Silvally АлександрⅠ увеличивает атаку.

Silvally АлександрⅠ теряет 1 здоровья.

Silvally АлександрⅠ промахивается

Togepi Василий растерянно попадает по себе.

Togepi Василий теряет 4 здоровья.

Silvally АлександрⅠ теряет 1 здоровья.

Silvally АлександрⅠ is using Metal Sound.

Togepi Василий уменьшает специальную защиту.

Togepi Василий is using Swagger.

Silvally АлександрⅠ увеличивает атаку.

Silvally АлександрⅠ теряет 1 здоровья.

Silvally АлександрⅠ is using Metal Sound.

Togepi Василий уменьшает специальную защиту.

Togepi Василий атакует.

Silvally АлександрⅠ теряет 4 здоровья.

Silvally АлександрⅠ теряет сознание.

В команде фиолетовых не осталось покемонов.

Команда полосатых побеждает в этом бою!

# Заключение

В процессе выполнения лабораторной работы я получил навыки использования

объектно-ориентированного подхода программирования при использовании языка Java.

Научился работать с объектами, классами, супер-классами, конструкторами, полями и модификаторами доступа. Понял основные концепции ООП. Узнал о наследовании как классов, так и конструкторов. Научился создавать UML-диаграммы.

# Список литературы.

1. https://se.ifmo.ru/courses/programming
2. <https://pokemondb.net/pokedex/>
3. <https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhnih-interfeysih>
4. <https://javarush.ru/groups/posts/591-vot-tak-final>
5. <https://www.youtube.com/watch?v=3Xo6zSBgdgk>
6. <https://javarush.ru/groups/posts/1985-raznica-mezhdu-abstraktnihmi-klassami-i-interfeysami>
7. <https://skillbox.ru/media/base/klassy-i-obekty-v-java/>
8. <https://www.jetbrains.com/help/idea/class-diagram.html#analyze_graph>
9. <https://metanit.com/java/tutorial/3.3.php>
10. JAVA from EPAM, 2-е издание, исправленное. И. Н. Блинов, В. С. Романчук
11. <https://javarush.ru/groups/posts/1967-otnoshenija-mezhdu-klassami-nasledovanie-kompozicija-i-agregirovanie->
12. <https://javarush.ru/groups/posts/599-inicializacija-poley>
13. https://javarush.ru/groups/posts/2347-klass-string-v-java