|  |  |  |  |
| --- | --- | --- | --- |
| Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»  Факультет программной инженерии и компьютерной техники | | | |
|  | | | |
| Лабораторная работа №2  Вариант 982 | | | |
|  | | | |
|  | | | |
|  |  |  | |
| Выполнил(а): | Кайгородова Александра Андреевна группа P3113 | |
| Проверил(а): | | Абузов Ярослав Александрович |
| Санкт-Петербург 2024 | | |

Содержание

[Задание 3](#_Toc182146303)

[UML диаграмма классов 5](#_Toc182146304)

[Код программы 6](#_Toc182146305)

[Результат программы 13](#_Toc182146306)

[Заключение 13](#_Toc182146307)

# 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. Все разработанные классы, не имеющие наследников, должны быть реализованы таким образом, чтобы от них нельзя было наследоваться.

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.

Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>

Комментарии

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с [документацией](https://se.ifmo.ru/~tony/doc/), обращая особое внимание на классы Pokemon и Move. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл Pokemon.jar. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть как она работает.

Battle b = new Battle();

Pokemon p1 = new Pokemon("Чужой", 1);

Pokemon p2 = new Pokemon("Хищник", 1);

b.addAlly(p1);

b.addFoe(p2);

b.go();

1. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса Pokemon. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
2. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса PhysicalMove или SpecialMove. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод describe, чтобы выводилось нужное сообщение.
3. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники StatusMove), скорее всего придется разобраться с классом Effect. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
4. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение

Изображение выглядит как текст

Автоматически созданное описание

# UML диаграмма классов

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

# Код программы

Атаки

Body Slam

package **Moves.physical**;  
import **ru.ifmo.se.pokemon.**\*;  
  
  
public class **BodySlam** extends **PhysicalMove**{  
 public BodySlam() { super (**Type**.*NORMAL* , 85.0, 100.0); }  
  
 protected **String** describe() {return "использует атаку Body Slam";}  
  
 protected void applyOppDamage(**Pokemon** var1, double var2) {  
 var1.setCondition((new Effect()).chance(0.3).condition(**Status**.*PARALYZE*));  
 }  
}

Brutal Swing

package **Moves.physical**;  
import **ru.ifmo.se.pokemon.PhysicalMove**;  
import **ru.ifmo.se.pokemon.Type**;  
  
public class **BrutalSwing** extends **PhysicalMove**{  
 public BrutalSwing() { super (**Type**.*DARK*, 60.0, 100.0); }  
  
 protected **String** describe() {  
 return "использует атаку Brutal Swing";  
 }  
}

Quick Attack

package **Moves.physical**;  
import **ru.ifmo.se.pokemon.PhysicalMove**;  
import **ru.ifmo.se.pokemon.Type**;  
  
public class **QuickAttack** extends **PhysicalMove** {  
 public QuickAttack() { super (**Type**.*NORMAL*, 40.0, 60.0); }  
  
 protected **String** describe() {  
 return "использует атаку Quick Attack";  
 }  
}

Psybeam

package **Moves.special**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Psybeam** extends **SpecialMove**{  
 public Psybeam() {super (**Type**.*PSYCHIC*, 65, 100);}  
  
 protected **String** describe() {  
 return "использует атаку Psybeam";  
 }  
  
 protected void applyOppEffects(**Pokemon** var1){**Effect**.*confuse*(var1);}  
}

Shadow Ball

package **Moves.special**;  
  
  
import **ru.ifmo.se.pokemon.**\*;

public class **ShadowBall** extends **SpecialMove**{  
 public ShadowBall() {super (**Type**.*PSYCHIC*, 80, 100);}  
 protected **String** describe() {return "использует атаку Shadow Ball";}  
  
 protected void applyOppDamage(**Pokemon** var1, double var2) {  
 var1.setMod(**Stat**.*SPECIAL\_DEFENSE*, -1);  
 }  
}

Thunderbolt

package **Moves.special**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Thunderbolt** extends **SpecialMove** {  
 public Thunderbolt() {super(**Type**.*ELECTRIC*, 90.0, 100.0);}  
  
 protected **String** describe() {  
 return "использует атаку Thunderbolt";  
 }  
  
 protected void applyOppEffects(**Pokemon** var1) {  
 var1.setCondition((new Effect()).chance(0.1).condition(**Status**.*PARALYZE*));  
 }  
}

Confide

package **Moves.status**;  
import **ru.ifmo.se.pokemon.**\*;  
public class **Confide** extends **StatusMove**{  
 public Confide() {super (**Type**.*NORMAL*, 0.0, 0.0);}  
  
 protected **String** describe() {return "использует Confide";}  
  
 protected void applyOppEffects(**Pokemon** var1){  
 var1.setMod(**Stat**.*SPECIAL\_ATTACK*,-1);}  
}

Defense Curl

package **Moves.status**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **DefenseCurl** extends **StatusMove** {  
 public DefenseCurl() {super (**Type**.*NORMAL*, 0.0, 0.0);}  
  
 protected **String** describe() {  
 return "использует DefenseCurl";  
 }  
 protected void applySelfDamage(**Pokemon** var1, double var2) {  
 var1.setMod(**Stat**.*SPECIAL\_DEFENSE*,1);  
 }  
}

Rest

package **Moves.status**;  
  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Rest** extends **StatusMove** {  
 public Rest() {super (**Type**.*PSYCHIC*, 0.0, 0.0);}  
  
 protected **String** describe() {return "использует Rest";}  
  
 protected void applySelfEffects(**Pokemon** var1) {  
 **Effect**.*sleep*(var1);  
  
 }  
   
}

Swagger

package **Moves.status**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Swagger** extends **StatusMove**{  
 public Swagger() {super (**Type**.*NORMAL*, 0.0, 85);}  
  
 protected **String** describe() {return "использует Swagger";}  
 protected void applyOppEffects(**Pokemon** var1){  
 **Effect**.*confuse*(var1);  
 var1.setMod(**Stat**.*ATTACK*, +2);  
 }  
  
}

Покемоны

Buneary

package **Pokemons**;  
import **Moves.special.ShadowBall**;  
import **Moves.physical.QuickAttack**;  
import **Moves.status.Rest**;  
import **ru.ifmo.se.pokemon.**\*;  
  
  
public class **Buneary** extends **Pokemon**{  
 public Buneary(**String** var1, int var2) {  
 super(var1, var2);  
 this.setType(**Type**.*NORMAL*);  
 this.setStats(55.0, 66.0, 44.0, 44.0, 56.0, 85.0);  
 this.setMove(new ShadowBall(), new QuickAttack(), new Rest());  
  
 }  
}

Igglybuff

package **Pokemons**;  
  
import **Moves.status.Rest**;  
import **Moves.status.DefenseCurl**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Igglybuff** extends **Pokemon**{  
 public Igglybuff(**String** var1, int var2){  
 super(var1, var2);  
 this.setType(**Type**.*NORMAL*, **Type**.*FAIRY*);  
 this.setStats(90, 30, 15, 40, 20, 15);  
 this.setMove(new Rest(), new DefenseCurl());  
 }  
}

Jiglypuff

package **Pokemons**;  
  
import **Moves.physical.BodySlam**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Jigglypuff** extends **Igglybuff**{  
 public Jigglypuff(**String** var1, int var2){  
 super(var1, var2);  
 this.setStats(115, 45, 20, 45, 25, 20);  
 this.setMove(new BodySlam());  
 }  
}

Kecleon

package **Pokemons**;  
  
import **Moves.special.Psybeam**;  
import **Moves.status.Rest**;  
import **Moves.special.Thunderbolt**;  
import **Moves.status.Confide**;  
import **ru.ifmo.se.pokemon.**\*;  
  
public class **Kecleon** extends **Pokemon**{  
 public Kecleon(**String** var1, int var2){  
 super(var1, var2);  
  
 this.setType(**Type**.*NORMAL*);  
 this.setStats(60.0, 90.0, 70.0, 60.0, 120.0, 40.0);  
 this.setMove(new Psybeam(), new Rest(), new Thunderbolt(), new Confide());  
 }  
}

Lopunny

package **Pokemons**;  
  
import **Moves.physical.BrutalSwing**;  
  
public class **Lopunny** extends **Buneary**{  
 public Lopunny(**String** var1, int var2) {  
 super(var1, var2);  
 this.setStats(65.0, 76.0, 84.0, 54.0, 96.0, 105.0);  
 this.setMove(new BrutalSwing());  
  
 }  
}

Wigglytuff

package **Pokemons**;  
import **Moves.status.Swagger**;  
  
  
public class **Wigglytuff** extends **Jigglypuff**{  
 public Wigglytuff(**String** var1, int var2){  
 super(var1, var2);  
 this.setStats(140, 70, 45, 85, 50, 45);  
 this.setMove(new Swagger());  
 }  
}

Main

import **ru.ifmo.se.pokemon.**\*;  
import **Pokemons.**\*;  
  
public class **Main** {  
 public static void main(**String**[] var0) {  
 **Battle** var1 = new Battle();  
 **Kecleon** var2 = new Kecleon("Кеклеон", 1);  
 **Buneary** var3= new Buneary("Банири", 1);  
 **Lopunny** var4= new Lopunny("Лопанни", 1);  
 **Igglybuff** var5= new Igglybuff("Иглибаф", 1);  
 **Jigglypuff** var6= new Jigglypuff("Джиглипаф", 1);  
 **Wigglytuff** var7= new Wigglytuff("Виглитаф", 1);  
 var1.addAlly(var2);  
 var1.addAlly(var4);  
 var1.addAlly(var6);  
 var1.addFoe(var3);  
 var1.addFoe(var5);  
 var1.addFoe(var7);  
 var1.go();  
 }  
}

# Результат программы

Kecleon Кеклеон from the team Greren enters the battle!

Buneary Банири from the team Purple enters the battle!

Buneary Банири использует атаку Quick Attack.

Kecleon Кеклеон loses 4 hit points.

Kecleon Кеклеон misses

Buneary Банири использует атаку Quick Attack.

Kecleon Кеклеон loses 4 hit points.

Kecleon Кеклеон misses

Buneary Банири использует атаку Shadow Ball.

Kecleon Кеклеон decreases special defense.

Kecleon Кеклеон misses

Buneary Банири misses

Kecleon Кеклеон использует атаку Psybeam.

Buneary Банири loses 5 hit points.

Buneary Банири использует атаку Quick Attack.

Kecleon Кеклеон loses 6 hit points.

Kecleon Кеклеон faints.

Lopunny Лопанни from the team Greren enters the battle!

Lopunny Лопанни использует атаку Brutal Swing.

Buneary Банири loses 5 hit points.

Buneary Банири использует атаку Shadow Ball.

Lopunny Лопанни decreases special defense.

Lopunny Лопанни использует атаку Brutal Swing.

Buneary Банири loses 5 hit points.

Buneary Банири faints.

Igglybuff Иглибаф from the team Purple enters the battle!

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 3 hit points.

Igglybuff Иглибаф misses

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 2 hit points.

Igglybuff Иглибаф misses

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 2 hit points.

Igglybuff Иглибаф misses

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 2 hit points.

Igglybuff Иглибаф misses

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 3 hit points.

Igglybuff Иглибаф misses

Lopunny Лопанни использует атаку Brutal Swing.

Igglybuff Иглибаф loses 2 hit points.

Igglybuff Иглибаф faints.

Wigglytuff Виглитаф from the team Purple enters the battle!

Lopunny Лопанни использует атаку Brutal Swing.

Wigglytuff Виглитаф loses 3 hit points.

Wigglytuff Виглитаф использует Swagger.

Lopunny Лопанни increases attack.

Lopunny Лопанни использует атаку Brutal Swing.

Wigglytuff Виглитаф loses 2 hit points.

Wigglytuff Виглитаф использует Swagger.

Lopunny Лопанни increases attack.

Lopunny Лопанни hits himself in confusion.

Lopunny Лопанни loses 5 hit points.

Wigglytuff Виглитаф использует Swagger.

Lopunny Лопанни increases attack.

Lopunny Лопанни hits himself in confusion.

Lopunny Лопанни loses 4 hit points.

Wigglytuff Виглитаф использует Swagger.

Lopunny Лопанни increases attack.

Lopunny Лопанни hits himself in confusion.

Lopunny Лопанни loses 6 hit points.

Lopunny Лопанни faints.

Jigglypuff Джиглипаф from the team Greren enters the battle!

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф is paralyzed

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф hits himself in confusion.

Jigglypuff Джиглипаф loses 4 hit points.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф hits himself in confusion.

Jigglypuff Джиглипаф loses 4 hit points.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф использует атаку Body Slam.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф hits himself in confusion.

Jigglypuff Джиглипаф loses 3 hit points.

Wigglytuff Виглитаф использует Swagger.

Jigglypuff Джиглипаф increases attack.

Jigglypuff Джиглипаф hits himself in confusion.

Jigglypuff Джиглипаф loses 5 hit points.

Jigglypuff Джиглипаф faints.

Team Greren loses its last Pokemon.

The team Purple wins the battle!

# Заключение

В ходе данной работы я научилась подключать внешние jar файлы и использовать их. Научилась работать с документацией, изучила объектно-ориентированный подход на языке Java. Научилась работать с классами, конструкторами, полями и модификаторами доступа.