

Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа №2

Вариант 39615

Преподаватель:

Выполнил: Хюинь Тан Куонг

R3138

Санкт-Петербург

2021

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.

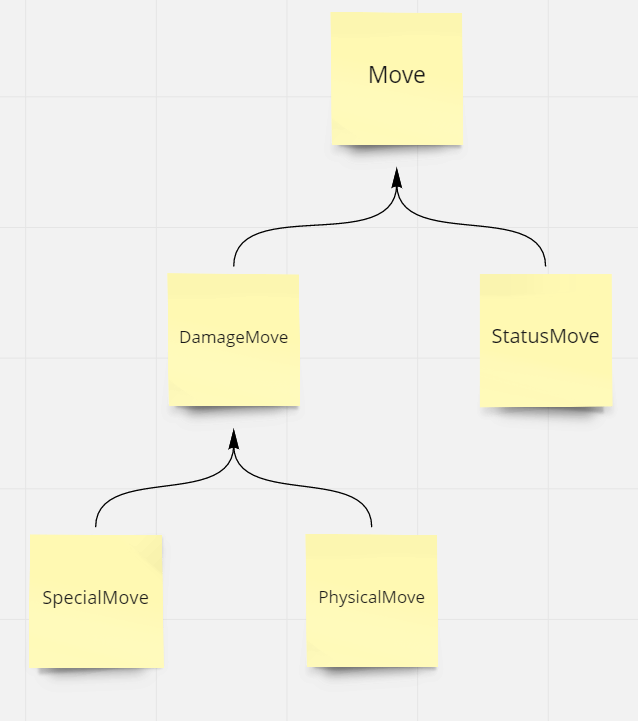
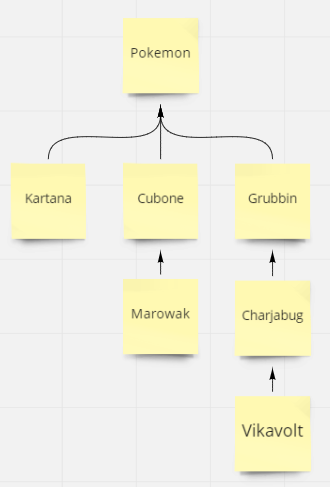
Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

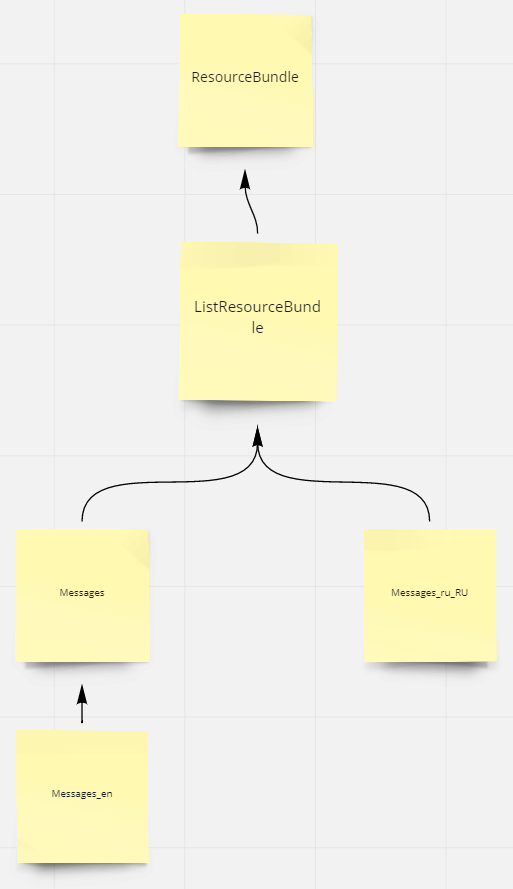
Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

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>



**Диаграмма классов:**





Исходный код:

1. Battleground.java

import ru.ifmo.se.pokemon.\*;

import Pokemons.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Battleground {

    public static void main(String[] args) {

    Battle field = new Battle();

        Pokemon p1 = new Kartana("Kartana", 1);

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

        Pokemon p3 = new Marowak("Marowak", 1);

        Pokemon p4 = new Grubbin("Grubbin", 1);

        Pokemon p5 = new Charjabug("Charjabug", 1);

        Pokemon p6 = new Vikavolt("Vikavolt", 1);

        field.addAlly(p1);

        field.addAlly(p2);

        field.addAlly(p3);

        field.addFoe(p4);

        field.addFoe(p5);

        field.addFoe(p6);

        field.go();

    }

}

1. Moves/Charge\_Beam.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Charge\_Beam extends SpecialMove {

    int raised\_stats = 0;

    public Charge\_Beam() {

        super(Type.ELECTRIC, 50, 0.9);

    }

    @Override

    protected void applySelfEffects(Pokemon p) {

        if (Math.random() <= 0.7) {

            if (raised\_stats < 6) {

                p.setMod(Stat.ATTACK, +1);

                raised\_stats+=2;

            }

        }

    }

    @Override

    protected String describe() {

        return "used Charge Beam";

    }

}

1. Moves/Facade.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Facade extends PhysicalMove {

    public Facade() {

        super(Type.NORMAL, 70, 1.0);

    }

    @Override

    protected void applyOppDamage(Pokemon p, double damage) {

        if(p.getCondition()==Status.BURN || p.getCondition()==Status.POISON || p.getCondition()==Status.PARALYZE) {

            p.setMod(Stat.HP, (int)damage\*2);

        }

    }

    @Override

    protected String describe() {

        return "used Facade";

    }

}

1. Moves/Ice\_Fang.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Ice\_Fang extends PhysicalMove{

    public Ice\_Fang() {

        super(Type.ICE, 65, 0.95);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.1) Effect.freeze(p);

        if (Math.random() < 0.1) Effect.flinch(p);

    }

    @Override

    protected String describe() {

        return "used Ice Fang";

    }

}

1. Moves/Rest.java

package Moves;

import ru.ifmo.se.pokemon.\*;

import Pokemons.Cubone;

/\*\*

 \*

 \* @author huynh

 \*/

public class Rest extends StatusMove{

    public Rest() {

        super(Type.PSYCHIC, 0, 0);

    }

    @Override

    protected void applySelfEffects(Pokemon p) {

        Effect e1 = new Effect().turns(0).stat(Stat.HP, -99);

        Effect e2 = new Effect().turns(2).condition(Status.SLEEP);

        p.addEffect(e1);

        p.setCondition(e2);

    }

    @Override

    protected java.lang.String describe() {

        return "used Rest";

    }

}

1. Moves/Scald.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Scald extends SpecialMove {

    public Scald() {

        super(Type.WATER, 80, 1.0);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        Effect.burn(p);

    }

    @Override

    protected String describe() {

        return "used Scald";

    }

}

1. Moves/Shadow\_Claw.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Shadow\_Claw extends PhysicalMove {

    public Shadow\_Claw() {

        super(Type.GHOST, 70, 1.0);

    }

    @Override

    protected String describe() {

        return "used Shadow Claw";

    }

}

1. Moves/Spark.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Spark extends PhysicalMove{

    public Spark() {

        super(Type.ELECTRIC, 65, 1.0);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() < 0.3) Effect.paralyze(p);

    }

    @Override

    protected String describe() {

        return "used Spark";

    }

}

1. Moves/Swagger.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Swagger extends StatusMove{

    int raised\_stats = 0;

    public Swagger() {

        super(Type.NORMAL, 0, 85);

    }

    @Override

    protected void applySelfEffects(Pokemon p) {

        Effect.confuse(p);

        if (raised\_stats < 6) {

            p.setMod(Stat.ATTACK, +2);

            raised\_stats +=2;

        }

    }

    @Override

    protected String describe() {

        return "used Swagger";

    }

}

1. Moves/Thunderbolt.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Thunderbolt extends SpecialMove {

    public Thunderbolt() {

        super(Type.ELECTRIC, 90, 1.0);

    }

   @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() <= 0.1) Effect.paralyze(p);

    }

    @Override

    protected String describe() {

        return "used Thunderbolt";

    }

}

1. Move/Waterfall.java

package Moves;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Waterfall extends PhysicalMove {

    public Waterfall() {

        super(Type.WATER, 80, 1.0);

    }

    @Override

    protected void applyOppEffects(Pokemon p) {

        if (Math.random() <= 0.2) Effect.flinch(p);

    }

    @Override

    protected String describe() {

        return "used Waterfall";

    }

}

1. Pokemons/Charjabug.java

package Pokemons;

import Moves.\*;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Charjabug extends Grubbin{

    public Charjabug(String name, int lvl) {

        super(name, lvl);

        setStats(57, 82, 95, 55, 75, 36);

        addType(Type.ELECTRIC);

        addMove(new Scald());

    }

}

1. Pokemons/Cubone.java

package Pokemons;

import ru.ifmo.se.pokemon.\*;

import Moves.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Cubone extends Pokemon{

     public Cubone(String name, int lvl) {

        super(name, lvl);

        setStats(50, 50, 95, 40, 50, 35);

        setType(Type.GROUND);

        setMove(new Rest(), new Swagger(), new Shadow\_Claw());

    }

}

1. Pokemons/Grubbin.java

package Pokemons;

import Moves.\*;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Grubbin extends Pokemon{

    public Grubbin(String name, int lvl) {

        super(name, lvl);

        setStats(47, 62, 45, 55, 45, 46);

        setType(Type.BUG);

        setMove(new Facade(), new Rest());

    }

}

1. Pokemons/Kartana.java

package Pokemons;

import ru.ifmo.se.pokemon.\*;

import Moves.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Kartana extends Pokemon{

    public Kartana(String name, int lvl) {

        super(name, lvl);

        setStats(59, 181, 131, 59, 31, 109);

        setType(Type.STEEL, Type.GRASS);

        setMove(new Thunderbolt(), new Ice\_Fang(), new Spark(), new Swagger());

    }

}

1. Pokemons/Marowak.java

package Pokemons;

import Moves.\*;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Marowak extends Cubone{

    public Marowak(String name, int lvl) {

        super(name, lvl);

        setStats(60, 80, 110, 50, 80, 45);

        addMove(new Waterfall());

    }

}

1. Pokemons/Vikavolt.java

package Pokemons;

import Moves.\*;

import ru.ifmo.se.pokemon.\*;

/\*\*

 \*

 \* @author huynh

 \*/

public class Vikavolt extends Charjabug {

    public Vikavolt(String name, int lvl) {

        super(name, lvl);

        setStats(77, 70, 90, 145, 75, 43);

        addMove(new Charge\_Beam());

    }

}

Output:

Kartana Kartana from the team Striped enters the battle!

Grubbin Grubbin from the team Yellow enters the battle!

Kartana Kartana used Spark.

Grubbin Grubbin loses 4 hit points.

Grubbin Grubbin used Facade.

Kartana Kartana used Thunderbolt.

Grubbin Grubbin loses 5 hit points.

Grubbin Grubbin misses

Kartana Kartana used Swagger.

Kartana Kartana increases attack.

Grubbin Grubbin used Facade.

Kartana Kartana used Swagger.

Kartana Kartana increases attack.

Grubbin Grubbin misses

Kartana Kartana hits himself in confusion.

Kartana Kartana loses 7 hit points.

Grubbin Grubbin used Facade.

Critical hit!

Kartana Kartana used Swagger.

Kartana Kartana increases attack.

Grubbin Grubbin misses

Kartana Kartana used Ice Fang.

Grubbin Grubbin loses 8 hit points.

Grubbin Grubbin faints.

Charjabug Charjabug from the team Yellow enters the battle!

Kartana Kartana used Thunderbolt.

Charjabug Charjabug loses 2 hit points.

Charjabug Charjabug used Facade.

Kartana Kartana used Thunderbolt.

Charjabug Charjabug loses 2 hit points.

Charjabug Charjabug used Scald.

Kartana Kartana loses 3 hit points.

Kartana Kartana is burned

Kartana Kartana hits himself in confusion.

Kartana Kartana loses 6 hit points.

Kartana Kartana faints.

Cubone Cubone from the team Striped enters the battle!

Charjabug Charjabug used Scald.

Cubone Cubone loses 8 hit points.

Cubone Cubone is burned

Cubone Cubone used Shadow Claw.

Charjabug Charjabug restores 1 hit points.

Charjabug Charjabug used Facade.

Cubone Cubone loses 8 hit points.

Cubone Cubone faints.

Marowak Marowak from the team Striped enters the battle!

Marowak Marowak used Swagger.

Marowak Marowak increases attack.

Charjabug Charjabug misses

Marowak Marowak hits himself in confusion.

Marowak Marowak loses 4 hit points.

Charjabug Charjabug misses

Marowak Marowak hits himself in confusion.

Marowak Marowak loses 5 hit points.

Charjabug Charjabug used Facade.

Marowak Marowak used Waterfall.

Charjabug Charjabug loses 6 hit points.

Charjabug Charjabug used Facade.

Marowak Marowak used Shadow Claw.

Charjabug Charjabug loses 4 hit points.

Charjabug Charjabug faints.

Vikavolt Vikavolt from the team Yellow enters the battle!

Marowak Marowak used Swagger.

Marowak Marowak increases attack.

Vikavolt Vikavolt misses

Marowak Marowak used Swagger.

Marowak Marowak increases attack.

Vikavolt Vikavolt used Facade.

Marowak Marowak hits himself in confusion.

Marowak Marowak loses 3 hit points.

Vikavolt Vikavolt misses

Marowak Marowak used Swagger.

Vikavolt Vikavolt misses

Marowak Marowak used Shadow Claw.

Vikavolt Vikavolt loses 7 hit points.

Vikavolt Vikavolt misses

Marowak Marowak used Waterfall.

Vikavolt Vikavolt loses 8 hit points.

Vikavolt Vikavolt faints.

Team Yellow loses its last Pokemon.

The team Striped wins the battle!

**Вывод:**

After this lab, I know about Object-Oriented Programming, Class, Object, Access Modified, Encapsulation, Inheritance, Polymorphism…