

Федеральное государственное образовательное учреждение высшего образования «Национальный
исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа № 2

Вариант 31061

Выполнил: Зыков Андрей Алексеевич

Группа: Р3106

Проверил: Вербовой А. А.,

Преподаватель практики факультета ПИиКТ

Санкт-Петербург 2024

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.

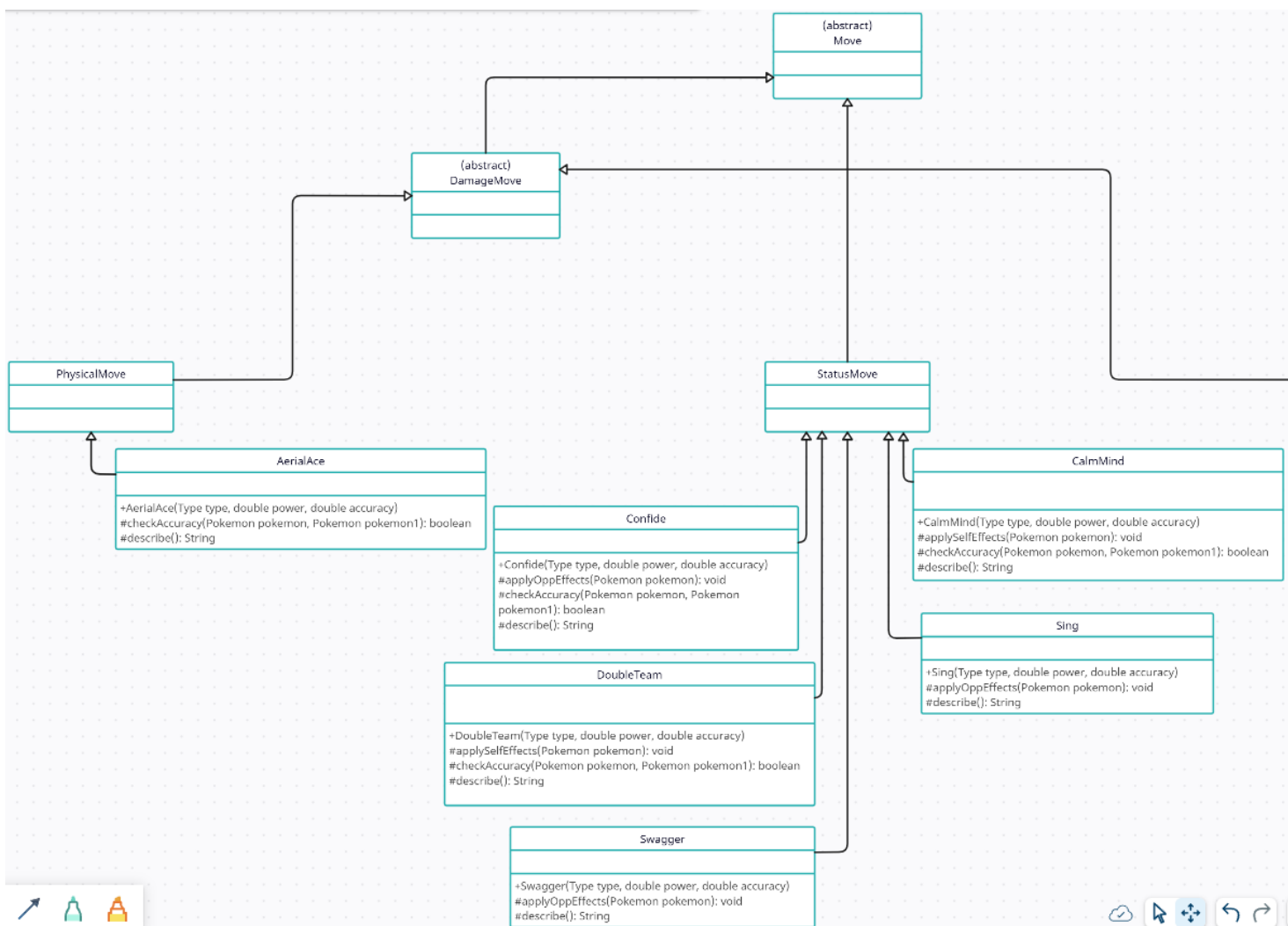
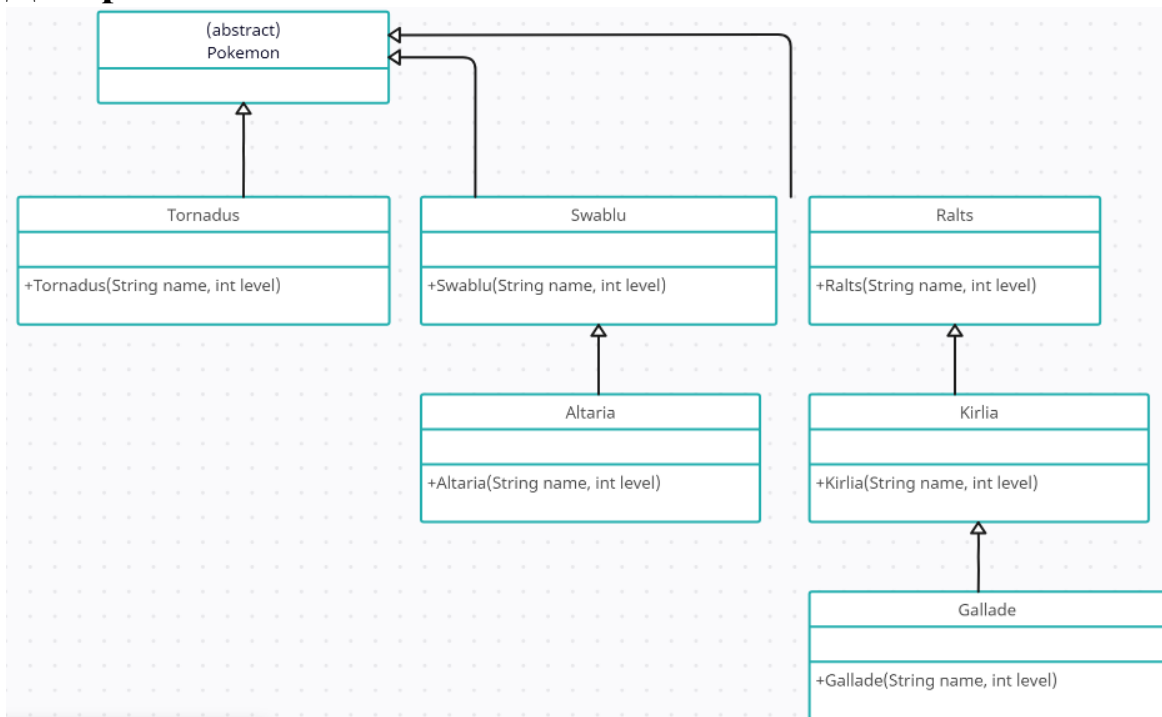
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

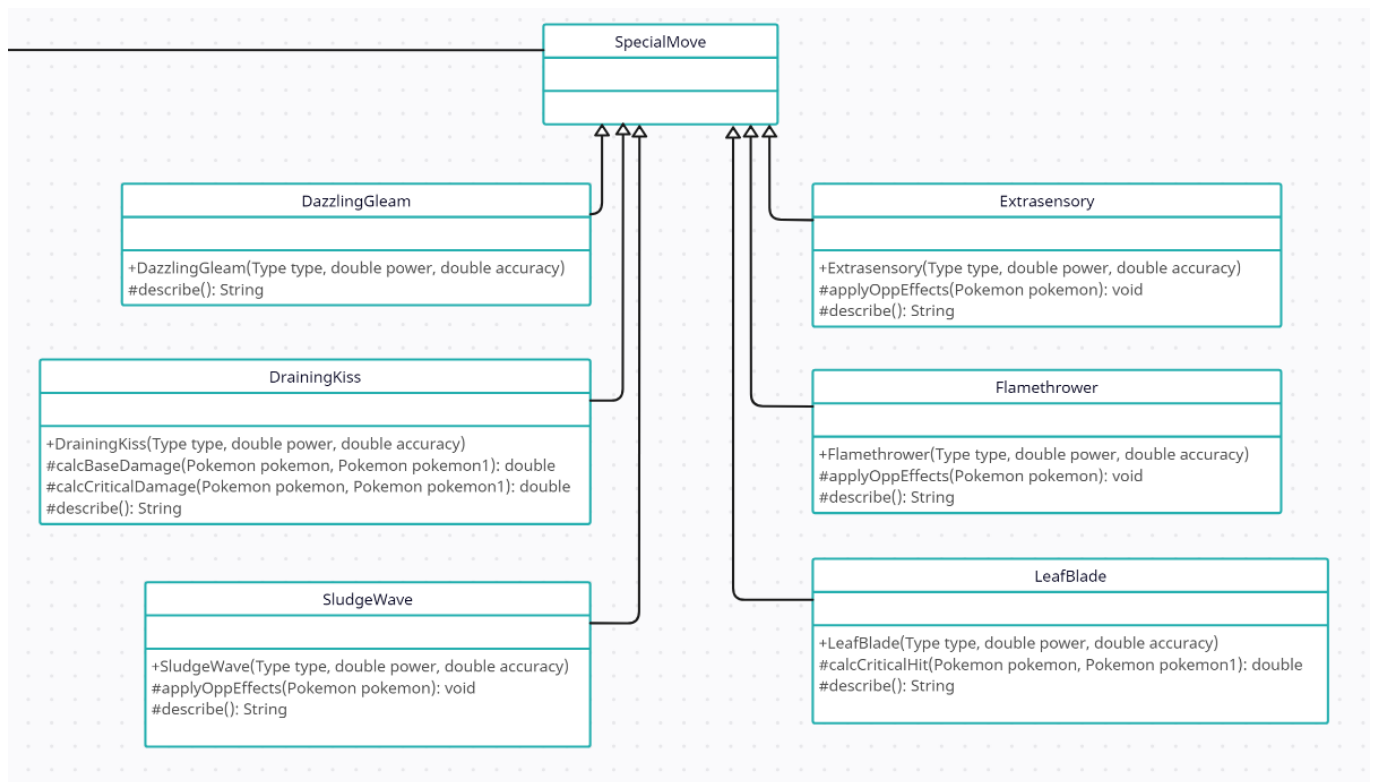
Используя класс симуляции боя `Battle`, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

Базовые классы и симулятор сражения находятся в [jar-архиве](#) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](#).

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

Диаграмма классов





Исходный код программы

```
package Pokemons;

import Moves.Flamethrower;

public final class Altaria extends Swablu {

    private static final double HP = 75;
    private static final double ATTACK = 70;
    private static final double DEFENSE = 90;
    private static final double SPECIAL_ATTACK = 70;
    private static final double SPECIAL_DEFENCE = 105;
    private static final double SPEED = 80;

    public Altaria(String name, int level) {
        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new Flamethrower());
    }
}
```

```
package Pokemons;

import Moves.LeafBlade;

public final class Gallade extends Kirlia {

    private static final double HP = 68;
    private static final double ATTACK = 125;
    private static final double DEFENSE = 65;
    private static final double SPECIAL_ATTACK = 65;
    private static final double SPECIAL_DEFENCE = 115;
    private static final double SPEED = 80;

    public Gallade(String name, int level) {
        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new LeafBlade());
    }
}
```

```

package Pokemons;

import Moves.DrainingKiss;

public class Kirlia extends Ralts {

    private static final double HP = 38;
    private static final double ATTACK = 35;
    private static final double DEFENSE = 35;
    private static final double SPECIAL_ATTACK = 65;
    private static final double SPECIAL_DEFENCE = 55;
    private static final double SPEED = 50;

    public Kirlia(String name, int level) {
        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new DrainingKiss());
    }
}

```

```

package Pokemons;

import Moves.CalmMind;
import Moves.DazzlingGleam;
import ru.ifmo.se.pokemon.*;

public class Ralts extends Pokemon {

    private static final double HP = 28;
    private static final double ATTACK = 25;
    private static final double DEFENSE = 25;
    private static final double SPECIAL_ATTACK = 45;
    private static final double SPECIAL_DEFENCE = 35;
    private static final double SPEED = 40;

    public Ralts(String name, int level) {
        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new CalmMind());
        addMove(new DazzlingGleam());
    }
}

```

```

package Pokemons;

import Moves.AerialAce;
import Moves.DoubleTeam;
import Moves.Sing;
import ru.ifmo.se.pokemon.*;

public class Swablu extends Pokemon {

    private static final double HP = 45;
    private static final double ATTACK = 40;
    private static final double DEFENSE = 60;
    private static final double SPECIAL_ATTACK = 40;
    private static final double SPECIAL_DEFENCE = 75;
    private static final double SPEED = 50;

    public Swablu(String name, int level) {
        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new DoubleTeam());
        addMove(new AerialAce());
        addMove(new Sing());
    }
}

```

```

package Pokemons;

import Moves.Confide;
import Moves.Extrasensory;
import Moves.SludgeWave;
import Moves.Swagger;
import ru.ifmo.se.pokemon.*;

public final class Tornadus extends Pokemon {

    private static final double HP = 79;
    private static final double ATTACK = 115;
    private static final double DEFENSE = 70;
    private static final double SPECIAL_ATTACK = 125;
    private static final double SPECIAL_DEFENCE = 80;
    private static final double SPEED = 111;

    public Tornadus(String name, int level) {

```

```

        super(name, level);
        setStats(HP, ATTACK, DEFENSE, SPECIAL_ATTACK, SPECIAL_DEFENCE,
SPEED);
        addMove(new Extrasensory());
        addMove(new SludgeWave());
        addMove(new Confide());
        addMove(new Swagger());
    }
}

```

```
package Moves;
```

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

```

public class AerialAce extends PhysicalMove {

    private static final Type TYPE = Type.FLYING;
    private static final double POWER = 60;
    private static final double ACCURACY = 0;

    public AerialAce() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {
        return true;
    }

    @Override
    protected String describe() {
        return "Использует атаку Aerial Ace";
    }
}

```

```
package Moves;
```

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

```
public class CalmMind extends StatusMove {
```



```

private static final Type TYPE = Type.PSYCHIC;
private static final double POWER = 0;
private static final double ACCURACY = 1;

public CalmMind() {
    super(TYPE, POWER, ACCURACY);
}

@Override
protected void applySelfEffects(Pokemon pokemon) {
    pokemon.setMod(Stat.SPECIAL_ATTACK, 1);
    pokemon.setMod(Stat.SPECIAL_DEFENSE, 1);
}

@Override
protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {
    return true;
}

@Override
protected String describe() {
    return "Использует атаку Calm Mind";
}
}

```

```
package Moves;
```

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

```

public class Confide extends StatusMove {

    private static final Type TYPE = Type.NORMAL;
    private static final double POWER = 0;
    private static final double ACCURACY = 0;

    public Confide() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {;
        pokemon.setMod(Stat.SPECIAL_ATTACK, -1);
    }
}

```

```

@Override
protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {
    return true;
}

@Override
protected String describe() {
    return "Использует атаку Confide";
}
}

```

```
package Moves;
```

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

```
public class DazzlingGleam extends SpecialMove {

    private static final Type TYPE = Type.FAIRY;
    private static final double POWER = 80;
    private static final double ACCURACY = 1;

    public DazzlingGleam() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected String describe() {
        return "Использует атаку Dazzling Gleam";
    }
}

```

```
package Moves;
```

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

```
public class DoubleTeam extends StatusMove {

    private static final Type TYPE = Type.NORMAL;
    private static final double POWER = 0;
    private static final double ACCURACY = 1;
}

```

```

public DoubleTeam() {
    super(TYPE, POWER, ACCURACY);
}

@Override
protected void applySelfEffects(Pokemon pokemon) {
    pokemon.setMod(Stat.EVASION, 1);
}

@Override
protected boolean checkAccuracy(Pokemon pokemon, Pokemon pokemon1) {
    return true;
}

@Override
protected String describe() {
    return "Использует атаку Double Team";
}
}

```

```

package Moves;

```

```

import ru.ifmo.se.pokemon.*;

```

```

public class DrainingKiss extends SpecialMove {

    private static final Type TYPE = Type.FAIRY;
    private static final double POWER = 50;
    private static final double ACCURACY = 1;

    public DrainingKiss() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected double calcBaseDamage(Pokemon pokemon, Pokemon pokemon1) {
        double damage = pokemon.getStat(Stat.SPECIAL_ATTACK) /
pokemon1.getStat(Stat.SPECIAL_DEFENSE);
        double hp = pokemon.getStat(Stat.HP);
        double attack = pokemon.getStat(Stat.ATTACK);
        double defense = pokemon.getStat(Stat.DEFENSE);
        double specialAttack = pokemon.getStat(Stat.SPECIAL_ATTACK);
        double specialDefence = pokemon.getStat(Stat.DEFENSE);
        double speed = pokemon.getStat(Stat.SPEED);
    }
}

```

```

        pokemon.setStats(hp + 0.75D * damage, attack, defense, specialAttack,
specialDefence, speed);
        return 0.25D * damage;
    }

    @Override
    protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {
        double damage = pokemon.getStat(Stat.SPECIAL_ATTACK) /
pokemon1.getStat(Stat.SPECIAL_DEFENSE);
        double hp = pokemon.getStat(Stat.HP);
        double attack = pokemon.getStat(Stat.ATTACK);
        double defense = pokemon.getStat(Stat.DEFENSE);
        double specialAttack = pokemon.getStat(Stat.SPECIAL_ATTACK);
        double specialDefence = pokemon.getStat(Stat.DEFENSE);
        double speed = pokemon.getStat(Stat.SPEED);
        pokemon.setStats(hp + 0.75D * damage, attack, defense, specialAttack,
specialDefence, speed);
        return 0.25D * damage;
    }

    @Override
    protected String describe() {
        return "Использует атаку Draining Kiss";
    }
}

```

```
package Moves;
```

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

```
public class Extrasensory extends SpecialMove {

    private static final Type TYPE = Type.PSYCHIC;
    private static final double POWER = 80;
    private static final double ACCURACY = 1;

    public Extrasensory() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        Effect effect = new Effect();
        effect.turns(1).chance(0.1);
    }
}

```

```

        effect.flinch(pokemon);
    }

    @Override
    protected String describe() {
        return "Использует атаку Extrasensory";
    }
}

package Moves;

import ru.ifmo.se.pokemon.*;

public class Flamethrower extends SpecialMove {

    private static final Type TYPE = Type.FIRE;
    private static final double POWER = 90;
    private static final double ACCURACY = 1;

    public Flamethrower() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        Effect effect = new Effect();
        effect.turns(1).chance(0.1);
        effect.burn(pokemon);
    }

    @Override
    protected String describe() {
        return "Использует атаку Flamethrower";
    }
}

```

```

package Moves;

import ru.ifmo.se.pokemon.*;
import java.util.Random;

```

```

public class LeafBlade extends PhysicalMove {

    private static final Type TYPE = Type.GRASS;
    private static final double POWER = 90;
    private static final double ACCURACY = 1;

    public LeafBlade() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {
        Random random = new Random();
        double percentage = random.nextDouble(1, 101);
        if (percentage <= 12.5D) {
            return 2;
        }
        return 1;
    }

    @Override
    protected String describe() {
        return "Использует атаку Leaf Blade";
    }
}

```

```

package Moves;

```

```

import ru.ifmo.se.pokemon.*;
import java.util.Random;

```

```

public class Sing extends StatusMove {

    private static final Type TYPE = Type.NORMAL;
    private static final double POWER = 0;
    private static final double ACCURACY = 0.55D;

    public Sing() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        Random random = new Random();
    }
}

```

```

        int moves = random.nextInt(2, 5);
        Effect effect = new Effect().chance(1).turns(moves);
        effect.sleep(pokemon);
    }

    @Override
    protected String describe() {
        return "Использует атаку Sing";
    }
}

```

```
package Moves;
```

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

```

public class SludgeWave extends SpecialMove {

    private static final Type TYPE = Type.POISON;
    private static final double POWER = 95;
    private static final double ACCURACY = 1;

    public SludgeWave() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        Effect effect = new Effect();
        effect.turns(1).chance(0.1D);
        effect.poison(pokemon);
    }

    @Override
    protected String describe() {
        return "Использует атаку Sludge Wave";
    }
}

```

```
package Moves;
```

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

```

import java.util.Random;

public class Swagger extends StatusMove {

    private static final Type TYPE = Type.NORMAL;
    private static final double POWER = 0;
    private static final double ACCURACY = 0.85D;

    public Swagger() {
        super(TYPE, POWER, ACCURACY);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        pokemon.setMod(Stat.ATTACK, 2);
        Random random = new Random();
        int turns = random.nextInt(2, 6);
        Effect effect = new Effect().turns(turns);
        effect.confuse(pokemon);
    }

    @Override
    protected String describe() {
        return "Использует атаку Swagger";
    }
}

```

Результат работы программы

Tornadus Tornadus из команды зеленых вступает в бой!

Swablu Swablu из команды синих вступает в бой!

Tornadus Tornadus Использует атаку Sludge Wave.

Критический удар!

Swablu Swablu теряет 315 здоровья.

Swablu Swablu теряет сознание.

Ralts Ralts из команды синих вступает в бой!

Tornadus Tornadus Использует атаку Extrasensory.

Критический удар!

Ralts Ralts теряет 213 здоровья.

Ralts Ralts теряет сознание.

Gallade Gallade из команды синих вступает в бой!

Tornadus Tornadus Использует атаку Swagger.

Gallade Gallade увеличивает атаку.

Gallade Gallade Использует атаку Dazzling Gleam.

Критический удар!

Tornadus Tornadus теряет 79 здоровья.

Tornadus Tornadus Использует атаку Sludge Wave.

Gallade Gallade теряет 39 здоровья.

Gallade Gallade Использует атаку Leaf Blade.

Tornadus Tornadus теряет 127 здоровья.

Tornadus Tornadus Использует атаку Swagger.

Gallade Gallade увеличивает атаку.

Gallade Gallade Использует атаку Calm Mind.

Gallade Gallade увеличивает специальную атаку.

Gallade Gallade увеличивает специальную защиту.

Tornadus Tornadus Использует атаку Sludge Wave.

Gallade Gallade теряет 29 здоровья.

Gallade Gallade Использует атаку Draining Kiss.

Tornadus Tornadus теряет 5 здоровья.

Gallade Gallade Использует атаку Leaf Blade.

Tornadus Tornadus теряет 12476 здоровья.

Tornadus Tornadus теряет сознание.

Altaria Altaria из команды зеленых вступает в бой!

Gallade Gallade Использует атаку Draining Kiss.
Altaria Altaria теряет 8892 здоровья.
Altaria Altaria теряет сознание.
Kirlia Kirlia из команды зеленых вступает в бой!
Gallade Gallade Использует атаку Dazzling Gleam.
Критический удар!
Kirlia Kirlia теряет 8993 здоровья.
Kirlia Kirlia теряет сознание.
В команде зеленых не осталось покемонов.
Команда синих побеждает в этом бою!

Создание jar-архива

```
cd /home/studs/s465993/Lab2  
  
javac -cp Pokemon.jar src/Main.java src/Pokemons/*.java src/Moves/*.java -d  
out  
  
jar -cfm Lab2.jar MANIFEST.MF Pokemon.jar -C out .  
  
java -jar Lab2.jar
```

Выводы по работе

В ходе выполнения лабораторной работы я повторил основы объектно-ориентированного программирования и построения диаграмм классов, попрактиковался в работе со сторонними источниками информации и документацией.