

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной техники

Программирование

Лабораторная работа №2

Вариант №3109967

Выполнил:

Кучерявый Игорь Дмитриевич

Группа Р3109

Преподаватели:

Наумова Н.А

**Санкт-Петербург
2025**

1) 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.

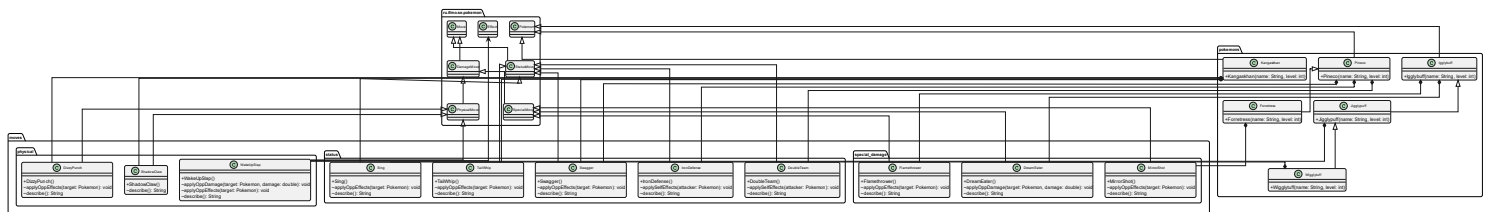
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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

Ваши покемоны:

Kangaskhan	Pineco	Forretress	Igglybuff	Jigglypuff	Wigglytuff
					
Атаки: <ul style="list-style-type: none">✂ Dizzy Punch✂ Shadow Claw✂ Swagger✂ Tail Whip	Атаки: <ul style="list-style-type: none">✂ Swagger✂ Iron Defense✂ Double Team	Атаки: <ul style="list-style-type: none">✂ Swagger✂ Iron Defense✂ Double Team✂ Mirror Shot	Атаки: <ul style="list-style-type: none">✂ Flamethrower✂ Dream Eater	Атаки: <ul style="list-style-type: none">✂ Flamethrower✂ Dream Eater✂ Wake-Up Slap	Атаки: <ul style="list-style-type: none">✂ Flamethrower✂ Dream Eater✂ Wake-Up Slap✂ Sing

2) Диаграмма классов



2) Исходный код

Покемоны:

1. Kangashan.java

```
package pokemons;

import moves.physical.DizzyPunch;
import moves.physical.ShadowClaw;

import moves.status.Swagger;

import moves.status.TailWhip;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;
```

```

public class Kangaskhan extends Pokemon{
    public Kangaskhan(String name, int level){
        super(name, level);
        super.setType(Type.NORMAL);
        super.setStats(105, 95, 80, 40, 80, 90);

        DizzyPunch dizzyPunch = new DizzyPunch();
        ShadowClaw shadowClaw = new ShadowClaw();
        Swagger swagger = new Swagger();
        TailWhip thailWhip = new TailWhip();
        super.setMove(dizzyPunch, shadowClaw, swagger, thailWhip);
    }
}

```

2. Pineco.java

```

package pokemons;

import moves.status.DoubleTeam;
import moves.status.IronDefense;
import moves.status.Swagger;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Pineco extends Pokemon{
    public Pineco(String name, int level){
        super(name, level);
        super.setType(Type.BUG);
        super.setStats(50, 65, 90, 35, 35, 15);

        Swagger swagger = new Swagger();
        IronDefense ironDefense = new IronDefense();
        DoubleTeam doubleTeam = new DoubleTeam();
        super.setMove(swagger, ironDefense, doubleTeam);
    }
}

```

3. Forretress.java

```

package pokemons;

import moves.special.MirrorShot;
import ru.ifmo.se.pokemon.Type;

public class Forretress extends Pineco{
    public Forretress(String name, int level){
        super(name, level);
        super.setType(Type.BUG, Type.STEEL);
        super.setStats(75, 90, 140, 60, 60, 40);

        MirrorShot mirrorShot = new MirrorShot();
        super.addMove(mirrorShot);
    }
}

```

4. Igglybuff.java

```

package pokemons;

import moves.physical.WakeUpSlap;
import moves.special.DreamEater;

```

```

import moves.special.Flamethrower;
import moves.special.MirrorShot;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Igglybuff extends Pokemon {
    public Igglybuff(String name, int level){
        super(name, level);
        super.setType(Type.NORMAL, Type.FAIRY);
        super.setStats(90, 30, 15, 40, 20, 15);

        Flamethrower flamethrower = new Flamethrower();
        DreamEater dreamEater = new DreamEater();
        super.setMove(flamethrower, dreamEater);
    }
}

```

5. Jigglypuff.java

```

package pokemons;

import moves.physical.WakeUpSlap;

public class Jigglypuff extends Igglybuff{
    public Jigglypuff(String name, int level){
        super(name, level);
        super.setStats(115, 45, 20, 45, 25, 20);

        WakeUpSlap wakeUpSlap = new WakeUpSlap();
        super.addMove(wakeUpSlap);
    }
}

```

6. Wigglytuff.java

```

package pokemons;

import moves.physical.WakeUpSlap;
import moves.status.Sing;

public class Wigglytuff extends Jigglypuff{
    public Wigglytuff(String name, int level){
        super(name, level);
        super.setStats(140, 70, 45, 85, 50, 45);

        Sing sing = new Sing();
        super.addMove(sing);
    }
}

```

Атаки:

- **Physical:**

1. DizzyPunch.java

```

package moves.physical;

import ru.ifmo.se.pokemon.PhysicalMove;

```

```

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class DizzyPunch extends PhysicalMove {
    public DizzyPunch() {
        super(Type.NORMAL, 70, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon target) {
        if (Math.random() < 0.2) { // 20% шанс
            target.confuse(); // сбивает цель с толку
        }
    }

    @Override
    protected String describe() {
        return "использует Dizzy Punch";
    }
}

```

2. ShadowClaw.java

```

package moves.physical;

import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Type;

public class ShadowClaw extends PhysicalMove {
    public ShadowClaw() {
        super(Type.GHOST, 70, 100);
    }

    @Override
    protected String describe() {
        return "использует Shadow Claw";
    }
}

```

3. WakeUpSlap.java

```

package moves.physical;

import ru.ifmo.se.pokemon.*;

public class WakeUpSlap extends PhysicalMove {
    public WakeUpSlap() {
        super(Type.FIGHTING, 70, 100);
    }

    @Override
    protected void applyOppDamage(Pokemon target, double damage) {
        if (target.getCondition() == Status.SLEEP) {
            damage = damage * 2;
            target.setMod(Stat.HP, (int) Math.round(damage));
        }
    }

    @Override
    protected void applyOppEffects(Pokemon target) {

```

```

        target.setCondition(new Effect().condition(Status.NORMAL));
    }

    @Override
    protected String describe() {
        return "использует Wake-Up Slap";
    }
}

```

• Special:

1. DreamEater.java

```

package moves.special;

import ru.ifmo.se.pokemon.*;

public class DreamEater extends SpecialMove {
    public DreamEater() {
        super(Type.PSYCHIC, 100, 100);
    }

    @Override
    protected void applyOppDamage(Pokemon target, double damage) {
        if (target.getCondition() == Status.SLEEP) {
            target.setMod(Stat.HP, (int) Math.round(damage));
        }
    }

    @Override
    protected String describe() {
        return "использует Dream Eater";
    }
}

```

2. Flamethrower.java

```

package moves.special;

import ru.ifmo.se.pokemon.*;

import static ru.ifmo.se.pokemon.Effect.burn;

public class Flamethrower extends SpecialMove {
    public Flamethrower() {
        super(Type.FIRE, 90, 100);
    }

    @Override
    public void applyOppEffects(Pokemon target) {
        if (Math.random() < 0.1) { // 10% шанс
            burn(target); // сжигает цель
        }
    }

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

```

3. MirrorShot.java

```

package moves.special;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.Type;

public class MirrorShot extends SpecialMove {
    public MirrorShot() {
        super(Type.STEEL, 65, 85);
    }

    @Override
    protected void applyOppEffects(Pokemon target) {
        if (Math.random() < 0.3) { // 30% шанс
            target.setMod(Stat.ACCURACY, -1); // снижает точность цели на 1 ступень
        }
    }

    @Override
    protected String describe() {
        return "использует Mirror Shot";
    }
}

```

- **Status:**

1. DoubleTeam.java

```

package moves.status;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

public class DoubleTeam extends StatusMove {
    public DoubleTeam() {
        super(Type.NORMAL, 0, 0);
    }

    @Override
    protected void applySelfEffects(Pokemon attacker) {
        attacker.setMod(Stat.EVASION, 1); // повышает уклончивость на 1 ступень
    }

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

```

2. IronDefense.java

```

package moves.status;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

public class IronDefense extends StatusMove {
    public IronDefense() {

```

```

        super(Type.NORMAL, 20, 100);
    }

    @Override
    protected void applySelfEffects(Pokemon attacker){
        attacker.setMod(Stat.DEFENSE, 2); // повышает защиту цели на 2 ступени
    }

    @Override
    protected String describe() {
        return "использует Iron Defense";
    }
}

```

3. Sing.java

```

package moves.status;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

import static ru.ifmo.se.pokemon.Effect.sleep;

public class Sing extends StatusMove {
    public Sing(){
        super(Type.NORMAL, 0, 55);
    }

    @Override
    protected void applyOppEffects(Pokemon target){
        sleep(target);
    }
}

```

4. Swagger.java

```

package moves.status;

import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Swagger extends StatusMove {
    public Swagger(){
        super(Type.NORMAL, 0, 85);
    }

    @Override
    protected void applyOppEffects(Pokemon target) {
        target.confuse(); // сбивает цель с толку
        target.setMod(Stat.ATTACK, 2); // увеличивает ее урон на 2 ступени
    }

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

```



```
}
```

5. TailWhip.java

```
package moves.status;

import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class TailWhip extends StatusMove {
    public TailWhip() {
        super(Type.NORMAL, 0, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon target) {
        target.setMod(Stat.DEFENSE, -1); // снижает защиту цели на 1 ступень
    }

    @Override
    protected String describe() {
        return "использует Tail Whip";
    }
}
```

3) Результат работы

```
Forretress форетрес из команды черных вступает в бой!
Kangaskhan кангасхан из команды желтых вступает в бой!
Kangaskhan кангасхан использует Shadow Claw.
Forretress форетрес теряет 6 здоровья.

Forretress форетрес использует Iron Defense.
Forretress форетрес увеличивает защиту.
```

Kangaskhan кангасхан использует Swagger.
Forretress форетрес увеличивает атаку.

Forretress форетрес использует Swagger.
Kangaskhan кангасхан увеличивает атаку.

Kangaskhan кангасхан растерянно попадает по себе.
Kangaskhan кангасхан теряет 3 здоровья.

Forretress форетрес использует Mirror Shot.
Kangaskhan кангасхан теряет 8 здоровья.

Kangaskhan кангасхан использует Dizzy Punch.
Forretress форетрес теряет 4 здоровья.

Forretress форетрес растерянно попадает по себе.
Forretress форетрес теряет 5 здоровья.
Forretress форетрес теряет сознание.
Igglybuff иглибаф из команды черных вступает в бой!
Kangaskhan кангасхан использует Swagger.
Igglybuff иглибаф увеличивает атаку.

Igglybuff иглибаф использует Flamethrower.
Kangaskhan кангасхан теряет 6 здоровья.
Kangaskhan кангасхан воспламеняется
Kangaskhan кангасхан теряет сознание.
Pinesco пинеко из команды желтых вступает в бой!
Pinesco пинеко использует Swagger.
Pinesco пинеко увеличивает атаку.

Pinesco пинеко использует Swagger.
Pinesco пинеко увеличивает атаку.

Igglybuff иглибаф растерянно попадает по себе.
Igglybuff иглибаф теряет 4 здоровья.

Pinesco пинеко использует Iron Defense.
Pinesco пинеко увеличивает защиту.

Igglybuff иглибаф использует Dream Eater.

Igglybuff иглибаф использует Dream Eater.
Критический удар!

Igglybuff иглибаф использует Dream Eater.

Igglybuff иглибаф использует Dream Eater.

Pinesco пинеко использует Swagger.

Pinesco пинеко увеличивает атаку.

Pinesco пинеко использует Swagger.

Pinesco пинеко увеличивает атаку.

Pinesco пинеко использует Swagger.

Igglybuff иглибаф увеличивает атаку.

Igglybuff иглибаф использует Flamethrower.

Pinesco пинеко теряет 8 здоровья.

Pinesco пинеко использует Swagger.

Pinesco пинеко увеличивает атаку.

Pinesco пинеко использует Swagger.

Pinesco пинеко увеличивает атаку.

Igglybuff иглибаф использует Flamethrower.

Pinesco пинеко теряет 8 здоровья.

Pinesco пинеко теряет сознание.

Wigglytuff виглитаф из команды желтых вступает в бой!

Wigglytuff виглитаф атакует.

Igglybuff иглибаф засыпает

Igglybuff иглибаф растерянно попадает по себе.

Igglybuff иглибаф теряет 4 здоровья.

Wigglytuff виглитаф использует Wake-Up Slap.

Igglybuff иглибаф теряет 6 здоровья.

Igglybuff иглибаф

Igglybuff иглибаф теряет сознание.

Jigglypuff джиглипаф из команды черных вступает в бой!

Wigglytuff виглитаф использует Flamethrower.

Jigglypuff джиглипаф теряет 5 здоровья.

Jigglypuff джиглипаф использует Flamethrower.

Wigglytuff виглитаф теряет 4 здоровья.

Wigglytuff виглиаф атакует.

Jigglypuff джиглипаф засыпает

Jigglypuff джиглипаф использует Wake-Up Slap.

Wigglytuff виглиаф атакует.

Wigglytuff виглиаф использует Flamethrower.

Jigglypuff джиглипаф теряет 4 здоровья.

Wigglytuff виглиаф использует Dream Eater.

Jigglypuff джиглипаф использует Dream Eater.

Wigglytuff виглиаф использует Wake-Up Slap.

Jigglypuff джиглипаф использует Wake-Up Slap.

Wigglytuff виглиаф использует Dream Eater.

Jigglypuff джиглипаф использует Flamethrower.

Wigglytuff виглиаф теряет 5 здоровья.

Wigglytuff виглиаф использует Wake-Up Slap.

Jigglypuff джиглипаф использует Dream Eater.

Wigglytuff виглиаф атакует.

Jigglypuff джиглипаф засыпает

Jigglypuff джиглипаф использует Flamethrower.

Wigglytuff виглиаф теряет 6 здоровья.

Wigglytuff виглиаф теряет сознание.

В команде желтых не осталось покемонов.

Команда черных побеждает в этом бою!