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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
по дисциплине
Программирование

Вариант: 1650

Выполнил:
Герасюто Фадей Александрович
Группа: P3119
Преподаватель:
Харитонова Анастасия Евгеньевна

Санкт-Петербург, 2025

Содержание

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.

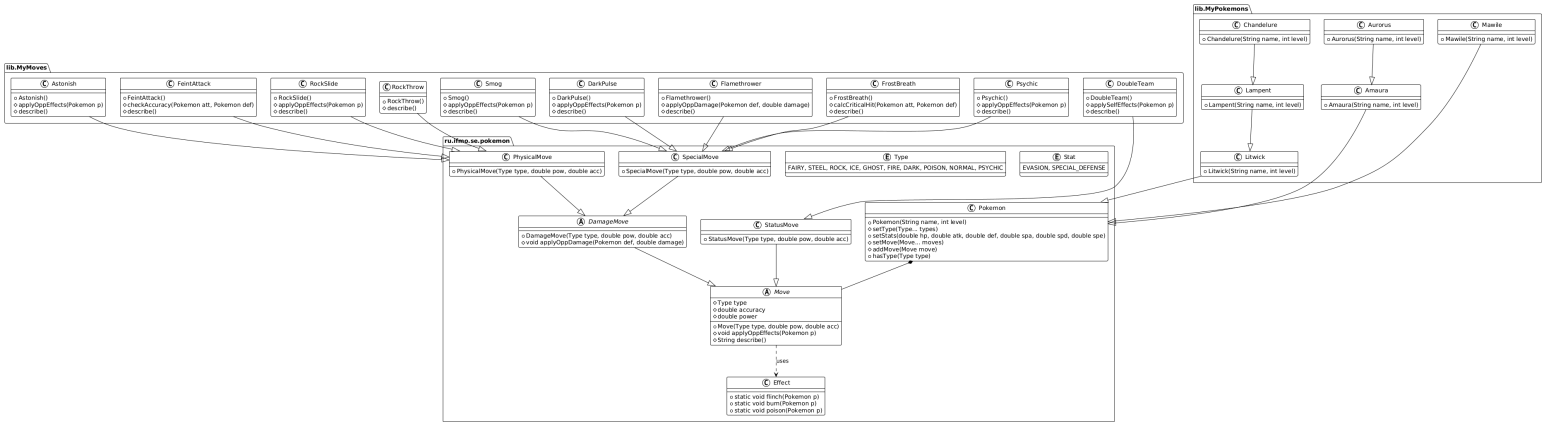
Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

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

Введите вариант: 30309

Ваши покемоны:

Диаграмма классов реализованной объектной модели



Исходный код программы

```java
// Main.java
import ru.ifmo.se.pokemon.Battle;

import lib.MyPokemons.Mawile;
import lib.MyPokemons.Amaura;
import lib.MyPokemons.Aurorus;
import lib.MyPokemons.Litwick;
import lib.MyPokemons.Lampent;
import lib.MyPokemons.Chandelure;


public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();

        Mawile m1 = new Mawile("Apple", 25);
        Amaura a1 = new Amaura("Pineapple", 30);
        Aurorus au1 = new Aurorus("Coconut", 39);
        Litwick l1 = new Litwick("Strawberry", 38);
        Lampent la1 = new Lampent("Grape", 41);
        Chandelure c1 = new Chandelure("Pear", 41);

        b.addAlly(m1);
        b.addAlly(au1);
        b.addAlly(l1);

        b.addFoe(a1);
        b.addFoe(la1);
        b.addFoe(c1);

        b.go();
    }
}

// lib/Mypokemons/Mawile.java
package lib.MyPokemons;

import lib.MyMoves.Astonish;
import lib.MyMoves.DoubleTeam;
import lib.MyMoves.FeintAttack;
import lib.MyMoves.Flamethrower;

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


public final class Mawile extends Pokemon {
    public Mawile(String name, int level) {
        super(name, level);
        setType(Type.FAIRY, Type.STEEL);
        setStats(50, 85, 85, 55, 55, 50);
        setMove(
                new Astonish(),
                new Flamethrower(),
                new FeintAttack(),
                new DoubleTeam()
        );
    }
}
```

```java
// lib/Mypokemons/Amaura.java
package lib.MyPokemons;

import lib.MyMoves.FrostBreath;
import lib.MyMoves.RockSlide;
import lib.MyMoves.RockThrow;

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


public class Amaura extends Pokemon {
    public Amaura(String name, int level) {
        super(name, level);
        setType(Type.ROCK, Type.ICE);
        setStats(77, 59, 50, 67, 63, 46);
        setMove(
                new RockSlide(),
                new RockThrow(),
                new FrostBreath()
        );
    }
}

// lib/Mypokemons/Aurorus.java
package lib.MyPokemons;

import lib.MyMoves.Psychic;


public final class Aurorus extends Amaura {
    public Aurorus(String name, int level) {
        super(name, level);
        setStats(123, 77, 72, 99, 92, 58);
        addMove(new Psychic());
    }
}

// lib/Mypokemons/Litwick.java
package lib.MyPokemons;

import lib.MyMoves.Flamethrower;
import lib.MyMoves.Smog;

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


public class Litwick extends Pokemon {
    public Litwick(String name, int level) {
        super(name, level);
        setType(Type.GHOST, Type.FIRE);
        setStats(50, 30, 55, 65, 55, 20);
        setMove(
                new Flamethrower(),
                new Smog()
        );
    }
}
```

```java
// lib/Mypokemons/Lampent.java
package lib.MyPokemons;

import lib.MyMoves.Astonish;


public class Lampent extends Litwick {
    public Lampent(String name, int level) {
        super(name, level);
        setStats(60, 40, 60, 95, 60, 55);
        addMove(new Astonish());
    }
}

// lib/Mypokemons/Chandelure.java
package lib.MyPokemons;

import lib.MyMoves.DarkPulse;


public final class Chandelure extends Lampent {
    public Chandelure(String name, int level) {
        super(name, level);
        setStats(60, 55, 90, 145, 90, 80);
        addMove(new DarkPulse());
    }
}

// lib/MyMoves/Astonish.java
package lib.MyMoves;

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


public final class Astonish extends PhysicalMove {
    public Astonish() {
        super(Type.GHOST, 30.0, 100.0);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() <= 0.3) {
            Effect.flinch(p);
        }
    }

    @Override
    protected String describe() {
        return "uses Astonish";
    }
}

// lib/MyMoves/Flamethrower.java
package lib.MyMoves;

import ru.ifmo.se.pokemon.*;


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

```java
    @Override
    protected void applyOppDamage(Pokemon def, double damage) {
        super.applyOppDamage(def, damage);
        if (Math.random() <= 0.1) {
            Effect.burn(def);
        }
    }

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

// lib/MyMoves/FeintAttack.java
package lib.MyMoves;

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


public final class FeintAttack extends PhysicalMove {
    public FeintAttack() {
        super(Type.DARK, 60, 100);
    }

    @Override
    protected boolean checkAccuracy(Pokemon att, Pokemon def) {
        return true;
    }

    @Override
    protected String describe() {
        return "uses Feint Attack";
    }
}

// lib/MyMoves/DoubleTeam.java
package lib.MyMoves;

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


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

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

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

// lib/MyMoves/RockSlide.java
```

```java
package lib.MyMoves;

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


public final class RockSlide extends PhysicalMove {
    public RockSlide() {
        super(Type.ROCK, 75, 90);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() <= 0.3) {
            Effect.flinch(p);
        }
    }

    @Override
    protected String describe() {
        return "uses Rock Slide";
    }
}

// lib/MyMoves/RockThrow.java
package lib.MyMoves;

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


public final class RockThrow extends PhysicalMove {
    public RockThrow() {
        super(Type.ROCK, 50, 90);
    }

    @Override
    protected String describe() {
        return "uses Rock Throw";
    }
}

// lib/MyMoves/FrostBreath.java
package lib.MyMoves;

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


public final class FrostBreath extends SpecialMove {
    public FrostBreath() {
        super(Type.ICE, 60, 90);
    }

    @Override
    protected double calcCriticalHit(Pokemon att, Pokemon def) {
        System.out.println("Critical hit!");
        return 2.0D;
    }

    @Override
    protected String describe() {
```

```
        return "uses Frost Breath";
    }
}

// lib/MyMoves/Psychic.java
package lib.MyMoves;

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


public final class Psychic extends SpecialMove {
    public Psychic() {
        super(Type.PSYCHIC, 90, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() <= 0.1) {
            p.setMod(Stat.SPECIAL_DEFENSE, -1);
        }
    }

    @Override
    protected String describe() {
        return "uses Psychic";
    }
}

// lib/MyMoves/Smog.java
package lib.MyMoves;

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


public final class Smog extends SpecialMove {
    public Smog() {
        super(Type.POISON, 30, 70);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (!(p.hasType(Type.POISON) || p.hasType(Type.STEEL)) && Math.random() <= 0.4 ) {
            Effect.poison(p);
        }
    }

    @Override
    protected String describe() {
        return "uses Smog";
    }
}

// lib/MyMoves/DarkPulse.java
package lib.MyMoves;

import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Type;
```

```java
public final class DarkPulse extends SpecialMove {
    public DarkPulse() {
        super(Type.DARK, 80, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() <= 0.2) {
            Effect.flinch(p);
        }
    }

    @Override
    protected String describe() {
        return "uses Dark Pulse";
    }
}
```

Результат работы программы

```
$ javac -cp lib/Pokemon.jar:lib/MyMoves:lib/MyPokemons:. Main.java
$ java -cp lib/Pokemon.jar:. Main
```

```
Mawile Apple from the team Purple enters the battle!
Amaura Pineapple from the team Red enters the battle!
Amaura Pineapple uses Rock Throw.
Mawile Apple loses 5 hit points.

Mawile Apple uses Astonish.
Amaura Pineapple loses 9 hit points.

Amaura Pineapple uses Rock Throw.
Mawile Apple loses 5 hit points.

Mawile Apple uses Astonish.
Amaura Pineapple loses 8 hit points.

Amaura Pineapple uses Frost Breath.
Critical hit!
Mawile Apple loses 18 hit points.

Mawile Apple uses Flamethrower.
Amaura Pineapple loses 9 hit points.

Amaura Pineapple uses Frost Breath.
Critical hit!
Mawile Apple loses 22 hit points.

Mawile Apple uses Feint Attack.
Critical hit!
Amaura Pineapple loses 18 hit points.

Amaura Pineapple uses Rock Throw.
Mawile Apple loses 7 hit points.

Mawile Apple uses Astonish.
Amaura Pineapple loses 7 hit points.

Amaura Pineapple uses Rock Slide.
Mawile Apple loses 9 hit points.
Mawile Apple faints.
Aurorus Coconut from the team Purple enters the battle!
Aurorus Coconut uses Psychic.
Amaura Pineapple loses 33 hit points.

Amaura Pineapple uses Rock Slide.
Aurorus Coconut loses 29 hit points.

Amaura Pineapple uses Rock Slide.
Aurorus Coconut loses 33 hit points.


Amaura Pineapple uses Rock Throw.
Aurorus Coconut loses 18 hit points.


Aurorus Coconut uses Psychic.
Amaura Pineapple loses 36 hit points.
Amaura Pineapple faints.
Lampent Grape from the team Red enters the battle!
Lampent Grape uses Astonish.
Aurorus Coconut loses 8 hit points.
```

```
Aurorus Coconut uses Rock Slide.
Lampent Grape loses 73 hit points.

Lampent Grape uses Smog.
Aurorus Coconut loses 4 hit points.

Aurorus Coconut uses Rock Slide.
Lampent Grape loses 48 hit points.
Lampent Grape faints.
Chandelure Pear from the team Red enters the battle!
Chandelure Pear uses Astonish.
Aurorus Coconut loses 14 hit points.

Aurorus Coconut uses Rock Slide.
Chandelure Pear loses 48 hit points.

Chandelure Pear uses Astonish.
Aurorus Coconut loses 12 hit points.

Aurorus Coconut uses Rock Throw.
Chandelure Pear loses 40 hit points.

Chandelure Pear uses Dark Pulse.
Aurorus Coconut loses 30 hit points.


Chandelure Pear uses Astonish.
Aurorus Coconut loses 12 hit points.
Aurorus Coconut faints.
Litwick Strawberry from the team Purple enters the battle!
Chandelure Pear uses Astonish.
Litwick Strawberry loses 22 hit points.

Litwick Strawberry uses Smog.
Chandelure Pear loses 3 hit points.

Chandelure Pear uses Dark Pulse.
Litwick Strawberry loses 95 hit points.
Litwick Strawberry faints.
Team Purple loses its last Pokemon.
The team Red wins the battle!
```

Собрать .jar файл

MANIFEST.MF

```
Manifest-Version: 1.0
Main-Class: Main
Class-Path: lib/Pokemon.jar
```

```
jar -cfm Main.jar MANIFEST.MF Main.class lib/MyPokemons/*.class lib/MyMoves/*.class
```

Вывод

В ходе выполнения лабораторной работы была реализована система покемонов с использованием принципов объектно-ориентированного программирования: наследования, инкапсуляции и полиморфизма. Созданы классы для шести покемонов и их атак, включая реализацию статусных эффектов (например, поджог от Flamethrower). Работа позволила углубить понимание ООП и научиться интегрировать внешние библиотеки в Java-проекты.