«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Программирование»

Отчет

По лабораторной работе №2

Вариант 290002

Студент:

Ильин Н. С.

P3110 поток 2.9

Преподаватель:

Наумова Н. А.

Санкт-Петербург, 2023 г.

Оглавление

Задание:

Лабораторная работа #2

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

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, 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();
```

4. 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.
5. 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.
6. 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.
7. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Введите вариант: 290002

Ваши покемоны:

Sealeo

Атаки:
- Substitute
- Double-Edge
- Take Down
- Endeavor

Skarmory

Атаки:
- Hydro Pump
- Thunder Shock
- Endeavor

Houndour

Атаки:
- Hydro Pump
- Thunder Shock
- Endeavor
- Bite

Oddish

Атаки:
- Light Screen
- Hydro Pump

Gloom

Атаки:
- Light Screen
- Hydro Pump
- Charge

Totodile

Атаки:
- Light Screen
- Hydro Pump
- Charge
- Meditate

Выполнение работы:

Исходный код:

src/Main.java

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

public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();
        b.addAlly(new Sealeo("Силео", 1));
        b.addAlly(new Skarmory("Скармори", 1));
        b.addAlly(new Houndour("Хандур", 1));
        b.addFoe(new Oddish("Оддиш", 1));
        b.addFoe(new Gloom("Глум", 1));
        b.addFoe(new Totodile("Тотодил", 1));
        b.go();
    }
}
```

src/utils/Utils.java

```
package utils;

public class Utils {
    public static boolean chance(double chance) {
        return Math.random() <= chance;
    }
}
```

src/mypokemons/Gloom.java

```
package mypokemons;

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

public class Gloom extends Pokemon {
    public Gloom(String name, int level) {
        super(name, level);
        super.setType(Type.GRASS, Type.POISON);
        super.setStats(60,65,70,85,75,40);
        super.setMove(new LightScreen(), new HydroPump(), new Charge());
    }
}
```

src/mypokemons/Houndour.java

```
package mypokemons;

import mymoves.*;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;
public class Houndour extends Pokemon {
    public Houndour(String name, int level) {
        super(name, level);
        super.setType(Type.DARK, Type.FIRE);
        super.setStats(45,60, 30,80,50,65);
        super.setMove(new HydroPump(), new ThunderShock(), new Endeavor(),
new Bite());
    }
}
```

src/mypokemons/Oddish.java

```
package mypokemons;

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

public class Oddish extends Pokemon {
    public Oddish(String name, int level) {
        super(name, level);
        super.setType(Type.GRASS, Type.POISON);
        super.setStats(45,50,55,75,65,30);
        super.setMove(new LightScreen(), new HydroPump());
    }
}
```

src/mypokemons/Sealeo.java

```
package mypokemons;

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

public class Sealeo extends Pokemon {
    public Sealeo(String name, int level) {
        super(name, level);
        super.setType(Type.ICE, Type.GHOST);
        super.setStats(90, 60, 70, 75, 70, 45);
        super.setMove(new DoubleEdge(), new Endeavor(), new Substitute(),
new TakeDown());
    }
}
```

src/mypokemons/Skarmory.java

```
package mypokemons;

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

public class Skarmory extends Pokemon {
    public Skarmory(String name, int level) {
        super(name, level);
        super.setType(Type.STEEL, Type.FLYING);
        super.setStats(65, 80, 140, 40, 70, 70);
        super.setMove(new HydroPump(), new ThunderShock(), new Endeavor());
    }
}
```

src/mypokemons/Totodile.java

```
package mypokemons;

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


public class Totodile extends Pokemon {
    public Totodile(String name, int level) {
        super(name, level);
        super.setType(Type.WATER);
        super.setStats(50,65,64,44,48,43);
        super.setMove(new LightScreen(), new HydroPump(), new Charge(), new
Meditate());
    }
}
```

src/mymoves/Bite.java

```
package mymoves;

import ru.ifmo.se.pokemon.*;
import utils.Utils;

public class Bite extends PhysicalMove {

    public Bite() {
        super(Type.DARK, 60, 100);
    }
    @Override
    protected void applyOppDamage(Pokemon def, double damage) {
        super.applyOppDamage(def, damage);

        if(Utils.chance(0.3)) {
            Effect.flinch(def);
        }
    }

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

src/mymoves/Charge.java

```
package mymoves;

import ru.ifmo.se.pokemon.*;
import utils.Utils;

public class Charge extends PhysicalMove {
    public Charge() {
        super(Type.ELECTRIC, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        p.addEffect(new Effect().turns(5).stat(Stat.SPECIAL_DEFENSE, 1));
    }

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

src/mymoves/DoubleEdge.java

```
package mymoves;

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

public class DoubleEdge extends PhysicalMove{
    public DoubleEdge(){
        super(Type.NORMAL,120,100);
    }
    @Override
    protected void applySelfDamage(Pokemon p, double damage){
        p.setMod(Stat.HP,(int)(damage/3.0));
    }

    @Override
    protected String describe(){
        return "использовал Double-Edge";
    }
}
```

src/mymoves/Endeavor.java

```
package mymoves;

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

public class Endeavor extends PhysicalMove {
    public Endeavor() {
        super(Type.NORMAL,0,100);
    }

    @Override
    protected double calcBaseDamage(Pokemon att, Pokemon def) {
        double attHP = att.getHP();
        double defHP = def.getHP();
        if (defHP > attHP) {
            def.setMod(Stat.HP, (int)(defHP-attHP));
        }
        return 0.0;
    }

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

src/mymoves/HydroPump.java

```
package mymoves;

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

public class HydroPump extends PhysicalMove {
    public HydroPump() {
        super(Type.WATER, 110, 80);
    }

    @Override
    protected String describe(){
        return "использует Hydro Pump";
    }
}
```

src/mymoves/LightScreen.java

```
package mymoves;

import ru.ifmo.se.pokemon.*;
import utils.Utils;

public class LightScreen extends PhysicalMove {
    public LightScreen() {
        super(Type.PSYCHIC, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        p.addEffect(new Effect().turns(5).stat(Stat.SPECIAL_DEFENSE,
(int)p.getStat(Stat.SPECIAL_DEFENSE)/2));
    }

    @Override
    protected String describe(){
        return "использует Light Screen";
    }
}
```

src/mymoves/Meditate.java

```
package mymoves;

import ru.ifmo.se.pokemon.*;

public class Meditate extends PhysicalMove {
    public Meditate() {
        super(Type.PSYCHIC, 0, 100);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        p.addEffect(new Effect().turns(5).stat(Stat.ATTACK, 1));
    }

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

src/mymoves/Substitute.java

```
package mymoves;

import ru.ifmo.se.pokemon.*;

public class Substitute extends PhysicalMove {
    public Substitute(){
        super(Type.NORMAL,0,100);
    }

    @Override
    protected void applySelfDamage(Pokemon p, double damage) {
        double currSelfHp = p.getHP();
        p.setMod(Stat.HP,(int)(currSelfHp*(-2)));
    }

    @Override
    protected String describe(){
        return "использовал Substitute";
    }
}
```

src/mymoves/TakeDown.java

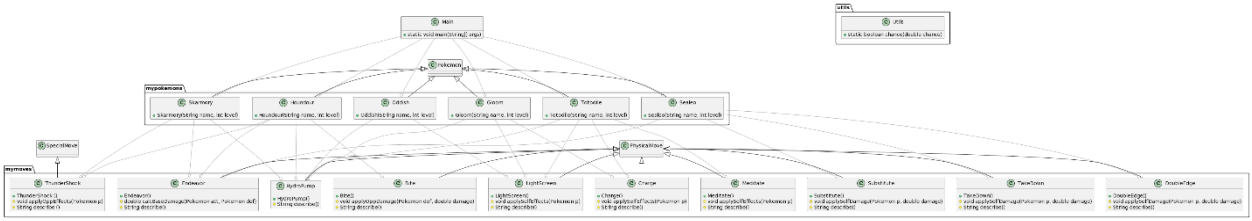
```
package mymoves;

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

public class TakeDown extends PhysicalMove {
    public TakeDown(){
        super(Type.NORMAL,90,85);
    }
    @Override
    protected void applySelfDamage(Pokemon p, double damage){
        p.setMod(Stat.HP,(int)(damage/4));
    }

    @Override
    protected String describe(){
        return "использовал Double-Edge";
    }
}
```

src/mymoves/ThunderShock.java

```
package mymoves;

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

public class ThunderShock extends SpecialMove{
    public ThunderShock () {
        super(Type.ELECTRIC, 40, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        p.addEffect(new Effect().chance(0.1).condition(Status.PARALYZE));
    }

    @Override
    protected String describe (){
        return "использует Thunder Shock";
    }
}
```

Диаграмма классов:

(svg)



Вывод программы:

```
Sealeo Силео из команды фиолетовых вступает в бой!
Oddish Оддиш из команды желтых вступает в бой!
Sealeo Силео использовал Double-Edge.
Oddish Оддиш теряет 5 здоровья.
Sealeo Силео теряет 1 здоровья.

Oddish Оддиш использует Hydro Pump.
Sealeo Силео теряет 5 здоровья.

Sealeo Силео использовал Double-Edge.
Oddish Оддиш теряет 5 здоровья.
Sealeo Силео теряет 1 здоровья.

Oddish Оддиш использует Light Screen.
Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.
Oddish Оддиш теряет 2 здоровья.
Sealeo Силео восстанавливает 7 здоровья.

Oddish Оддиш использует Light Screen.
Sealeo Силео теряет 3 здоровья.

Sealeo Силео использовал Substitute.
Oddish Оддиш теряет 3 здоровья.
Sealeo Силео восстанавливает 15 здоровья.
Oddish Оддиш теряет сознание.
Gloom Глум из команды желтых вступает в бой!
Sealeo Силео использовал Substitute.
Gloom Глум теряет 3 здоровья.
Sealeo Силео восстанавливает 45 здоровья.

Gloom Глум использует Charge.
Sealeo Силео теряет 2 здоровья.

Sealeo Силео использует Endeavor.
Gloom Глум теряет 2 здоровья.
```

```
Gloom Глум использует Hydro Pump.
Sealeo Силео теряет 4 здоровья.

Sealeo Силео использует Endeavor.
Gloom Глум теряет 3 здоровья.

Gloom Глум использует Light Screen.
Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.
Gloom Глум теряет 3 здоровья.
Sealeo Силео восстанавливает 119 здоровья.

Gloom Глум использует Charge.
Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Substitute.
Gloom Глум теряет 4 здоровья.
Sealeo Силео восстанавливает 353 здоровья.
Gloom Глум теряет сознание.
Totodile Тотодил из команды желтых вступает в бой!
Sealeo Силео использовал Double-Edge.
Totodile Тотодил теряет 5 здоровья.
Sealeo Силео теряет 1 здоровья.

Totodile Тотодил использует Light Screen.
Sealeo Силео теряет 2 здоровья.

Sealeo Силео использовал Double-Edge.
Totodile Тотодил теряет 5 здоровья.
Sealeo Силео теряет 1 здоровья.

Totodile Тотодил использует Meditate.
Sealeo Силео теряет 3 здоровья.

Sealeo Силео использовал Double-Edge.
Totodile Тотодил теряет 4 здоровья.
Sealeo Силео теряет 1 здоровья.
Totodile Тотодил теряет сознание.
В команде желтых не осталось покемонов.
Команда фиолетовых побеждает в этом бою!
```

Выводы:

В результате проделанной работы, я познакомился с основами ООП, научился подключать внешние библиотеки, изучил работу методов, классов и объектов.