

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа № 2

Вариант 4233

Выполнил студент: Маликов Глеб Игоревич

Группа № Р3124

Преподаватель: Харитонов Анастасия Евгеньевна

г. Санкт-Петербург

2022

Оглавление

Задание	3
Исходный код.....	4
Main.java	4
PhysicalAttacks.....	4
SpecialAttacks	5
StatusAttacks	7
Pokemons.....	9
Диаграмма классов.....	11
Результат работы программы	12
Вывод.....	14
Список литературы	15

Задание

Ознакомиться с документацией, обращая особое внимание на классы `Pokemon` и `Move`. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.

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.

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();
```

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.

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.

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.

Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Yveltal  Атаки: ✓ Psychic ✓ Dark Pulse ✓ Hurricane ✓ Rest	Hippopotas  Атаки: ✓ Double Team ✓ Bulldoze ✓ Confide	Hippowdon  Атаки: ✓ Double Team ✓ Bulldoze ✓ Confide ✓ Fire Fang	Nidoran-F  Атаки: ✓ Rest ✓ Swagger	Nidorina  Атаки: ✓ Rest ✓ Swagger ✓ Bite	Nidoqueen  Атаки: ✓ Rest ✓ Swagger ✓ Bite ✓ Double Team
--	---	---	--	--	--

Исходный код

Репозиторий: https://github.com/glebmavi/Lab2_Prog_oct_2022

Main.java

```
/**
 * Лабораторная работа №2 по программированию
 * Вариант 4233
 * @author Глеб Игоревич Маликов
 */
import Pokemons.*;
import ru.ifmo.se.pokemon.*;

public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();
        Yveltal p1 = new Yveltal("Ave Mamadisima", 1);
        Nidoqueen p2 = new Nidoqueen("Bolota Azul", 1);
        NidoranF p3 = new NidoranF("Bolita Azul", 1);
        Nidorina p4 = new Nidorina("Bola Azul", 1);
        Hippopotas p5 = new Hippopotas("Hippo", 1);
        Hippowdon p6 = new Hippowdon("Hippo Mamado", 1);
        b.addAlly(p1);
        b.addAlly(p2);
        b.addAlly(p3);
        b.addFoe(p4);
        b.addFoe(p5);
        b.addFoe(p6);
        b.go();
    }
}
```

PhysicalAttacks

Bite.java:

```
package Attacks.PhysicalAttacks;

import ru.ifmo.se.pokemon.*;

public class Bite extends PhysicalMove {
    public Bite() {
        super(Type.DARK, 60, 100);
    }
    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect bite_flinch = new Effect().chance(0.3);
        bite_flinch.flinch(pokemon);
        pokemon.addEffect(bite_flinch);
        super.applyOppEffects(pokemon);
    }

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

Bulldoze.java:

```
package Attacks.PhysicalAttacks;

import ru.ifmo.se.pokemon.*;

public class Bulldoze extends PhysicalMove {
    public Bulldoze() {
        super(Type.GROUND, 60, 1);
    }
    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect bulldoze_speed = new Effect().chance(1).stat(Stat.SPEED, -1);
        pokemon.addEffect(bulldoze_speed);
        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe() {
        return "used Bulldoze";
    }
}
```

Fire_Fang.java:

```
package Attacks.PhysicalAttacks;

import ru.ifmo.se.pokemon.*;

public class Fire_Fang extends PhysicalMove {
    public Fire_Fang() {
        super(Type.FIRE, 65, 0.95);
    }
    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect fire_fang = new Effect().chance(0.1).condition(Status.BURN);
        fire_fang.flinch(pokemon);
        pokemon.addEffect(fire_fang);
        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe() {
        return "used Fire Fang";
    }
}
```

SpecialAttacks

Dark_Pulse.java:

```
package Attacks.SpecialAttacks;

import ru.ifmo.se.pokemon.*;

public class Dark_Pulse extends SpecialMove {
```

```

public Dark_Pulse() {
    super(Type.DARK,80,1);
}

@Override
protected void applyOppEffects(Pokemon pokemon) {

    Effect dark_flinch = new Effect().chance(0.2);
    dark_flinch.flinch(pokemon);
    pokemon.addEffect(dark_flinch);
    super.applyOppEffects(pokemon);
}

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

```

Hurricane.java:

```

package Attacks.SpecialAttacks;

import ru.ifmo.se.pokemon.*;

public class Hurricane extends SpecialMove {
    public Hurricane() {
        super(Type.FLYING,110,0.7);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect hurricane_confuse = new Effect().chance(0.3);
        hurricane_confuse.confuse(pokemon);
        pokemon.addEffect(hurricane_confuse);
        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe() {
        return "used Hurricane";
    }
}

```

Psychic.java:

```

package Attacks.SpecialAttacks;

import ru.ifmo.se.pokemon.*;

public class Psychic extends SpecialMove {
    public Psychic() {
        super(Type.PSYCHIC,90,0.5);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {

```

```

        Effect psy_low_def = new Effect().turns(1).chance(0.1).stat(Stat.SPECIAL_DEFENSE, -1);
        pokemon.addEffect(psy_low_def);
        super.applyOppEffects(pokemon);
    }
    @Override
    protected String describe() {
        return "used Psychic";
    }
}

```

StatusAttacks

Confide.java:

```

package Attacks.StatusAttacks;

import ru.ifmo.se.pokemon.*;

public class Confide extends StatusMove{
    public Confide() {
        super(Type.NORMAL, 0, 1);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect confide_s_a = new Effect().chance(1).stat(Stat.SPECIAL_ATTACK,
-1);
        pokemon.addEffect(confide_s_a);
        super.applyOppEffects(pokemon);
    }

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

```

Double_Team.java:

```

package Attacks.StatusAttacks;

import ru.ifmo.se.pokemon.*;

public class Double_Team extends StatusMove {
    public Double_Team() {
        super(Type.NORMAL, 0, 1);
    }

    @Override
    protected void applySelfEffects(Pokemon pokemon) {

        Effect double_team_evasion = new Effect().chance(1).stat(Stat.EVA-
SION, 1);
        pokemon.addEffect(double_team_evasion);
        super.applySelfEffects(pokemon);
    }

    @Override

```

```

        protected String describe() {
            return "used Double Team";
        }
    }
}

```

Rest.java:

```

package Attacks.StatusAttacks;

import ru.ifmo.se.pokemon.*;

public class Rest extends StatusMove {
    private int Total_HP;
    public Rest(int HP) {
        super(Type.PSYCHIC, 0, 1);
        this.Total_HP = HP;
    }

    @Override
    protected void applySelfEffects(Pokemon pokemon) {

        Effect rest_sleep = new Effect().chance(1).turns(2).stat(Stat.HP, Total_HP);
        rest_sleep.sleep(pokemon);
        pokemon.addEffect(rest_sleep);
        super.applySelfEffects(pokemon);
    }

    @Override
    protected String describe() {
        return "used Rest";
    }
}

```

Swagger.java:

```

package Attacks.StatusAttacks;

import ru.ifmo.se.pokemon.*;

public class Swagger extends StatusMove {
    public Swagger() {
        super(Type.PSYCHIC, 0, 0.85);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {

        Effect swagger = new Effect().stat(Stat.ATTACK, 2);
        swagger.confuse(pokemon);
        pokemon.addEffect(swagger);
        super.applyOppEffects(pokemon);
    }

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

```


Pokemons

Hippopotas.java:

```
package Pokemons;

import Attacks.PhysicalAttacks.*;
import Attacks.StatusAttacks.*;
import ru.ifmo.se.pokemon.*;

public class Hippopotas extends Pokemon {
    public Hippopotas(String name, int level) {
        super(name, level);
        setType(Type.GROUND);
        setStats(68, 72, 78, 38, 42, 32);
        setMove(new Double_Team(), new Bulldoze(), new Confide());
    }
}
```

Hippowdon.java:

```
package Pokemons;

import Attacks.PhysicalAttacks.*;
import ru.ifmo.se.pokemon.*;

public class Hippowdon extends Hippopotas {
    public Hippowdon(String name, int level) {
        super(name, level);
        setType(Type.GROUND);
        setStats(108, 112, 118, 68, 72, 47);
        addMove(new Fire_Fang());
    }
}
```

NidoranF.java:

```
package Pokemons;

import Attacks.StatusAttacks.*;
import ru.ifmo.se.pokemon.*;

public class NidoranF extends Pokemon {
    public NidoranF(String name, int level) {
        super(name, level);
        setType(Type.POISON);
        setStats(55, 47, 52, 40, 40, 41);
        setMove(new Rest(55), new Swagger());
    }
}
```

Nidorina.java:

```
package Pokemons;

import Attacks.PhysicalAttacks.*;
import Attacks.StatusAttacks.*;
```

```

import ru.ifmo.se.pokemon.*;

public class Nidorina extends NidoranF {
    public Nidorina(String name, int level) {
        super(name, level);
        setType(Type.POISON);
        setStats(70, 62, 67, 55, 55, 56);
        setMove(new Rest(70), new Swagger(), new Bite());
    }
}

```

Nidoqueen.java:

```

package Pokemons;

import Attacks.PhysicalAttacks.*;
import Attacks.StatusAttacks.*;
import ru.ifmo.se.pokemon.*;
public class Nidoqueen extends Nidorina {
    public Nidoqueen(String name, int level) {
        super(name, level);
        setType(Type.POISON, Type.GROUND);
        setStats(90, 92, 87, 75, 85, 76);
        setMove(new Rest(90), new Swagger(), new Bite(), new Double_Team());
    }
}

```

Yveltal.java:

```

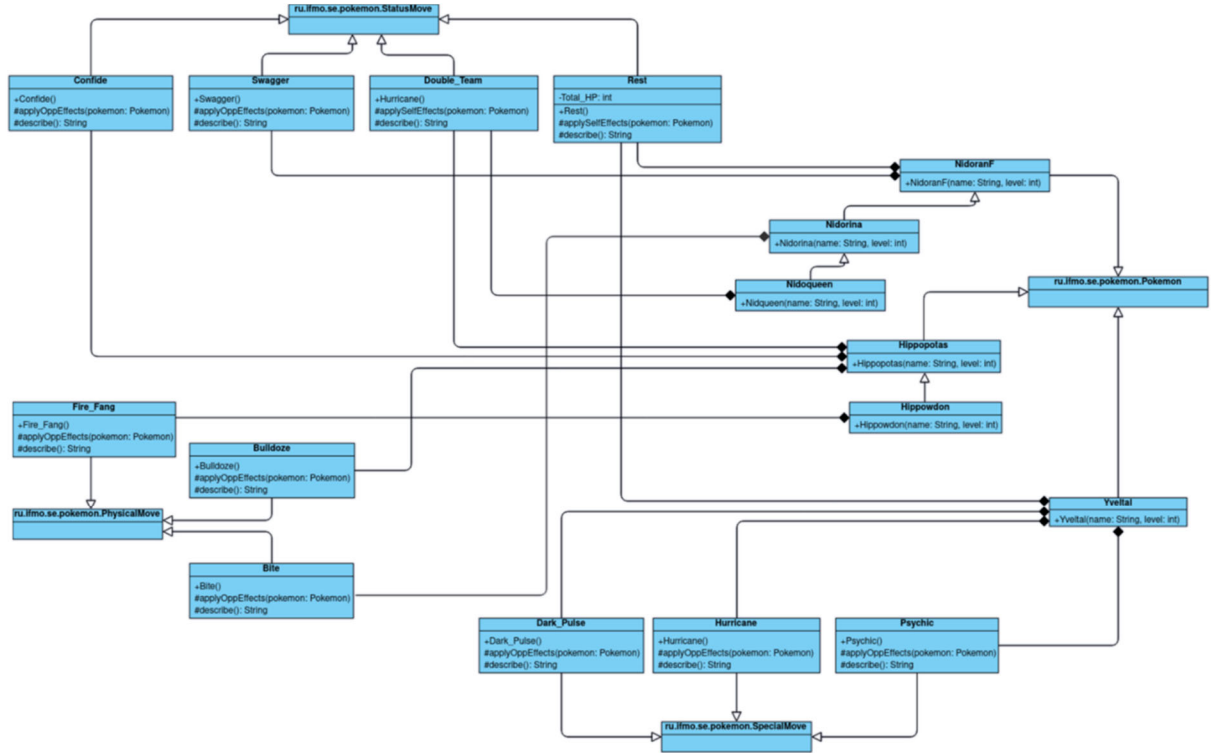
package Pokemons;

import Attacks.SpecialAttacks.*;
import Attacks.StatusAttacks.*;
import ru.ifmo.se.pokemon.*;

public class Yveltal extends Pokemon {
    public Yveltal(String name, int level) {
        super(name, level);
        setType(Type.DARK, Type.FLYING);
        setStats(126, 131, 95, 131, 98, 99);
        setMove(new Psychic(), new Dark_Pulse(), new Hurricane(), new
Rest(126));
    }
}

```

Диаграмма классов



Результат работы программы

```
[s372819@helios ~/lab2]$ java -jar Lab2.jar
```

Yveltal Ave Mamadisima из команды зеленых вступает в бой!

Nidorina Bola Azul из команды фиолетовых вступает в бой!

Yveltal Ave Mamadisima used Psychic.

Nidorina Bola Azul теряет 8 здоровья.

Nidorina Bola Azul used Swagger.

Yveltal Ave Mamadisima не замечает воздействие типа PSYCHIC

Yveltal Ave Mamadisima used Hurricane.

Nidorina Bola Azul теряет 11 здоровья.

Nidorina Bola Azul теряет сознание.

Hippopotas Hippo из команды фиолетовых вступает в бой!

Yveltal Ave Mamadisima промахивается

Hippopotas Hippo used Bulldoze.

Yveltal Ave Mamadisima теряет 1 здоровья.

Yveltal Ave Mamadisima не замечает воздействие типа GROUND

Yveltal Ave Mamadisima used Psychic.

Hippopotas Hippo теряет 7 здоровья.

Hippopotas Hippo used Bulldoze.

Yveltal Ave Mamadisima теряет 1 здоровья.

Yveltal Ave Mamadisima не замечает воздействие типа GROUND

Yveltal Ave Mamadisima used Psychic.

Hippopotas Hippo теряет 6 здоровья.

Hippopotas Hippo теряет сознание.

Hippowdon Hippo Mamado из команды фиолетовых вступает в бой!

Yveltal Ave Mamadisima промахивается

Hippowdon Hippo Mamado used Confide.

Yveltal Ave Mamadisima used Hurricane.

Hippowdon Hippo Mamado теряет 12 здоровья.

Hippowdon Hippo Mamado used Bulldoze.

Yveltal Ave Mamadisima теряет 1 здоровья.

Yveltal Ave Mamadisima не замечает воздействие типа GROUND

Yveltal Ave Mamadisima used Rest.

Hippowdon Hippo Mamado used Double Team.

Yveltal Ave Mamadisima used Rest.

Hippowdon Hippo Mamado used Bulldoze.

Yveltal Ave Mamadisima теряет 1 здоровья.

Yveltal Ave Mamadisima не замечает воздействие типа GROUND

Yveltal Ave Mamadisima промахивается

Hippowdon Hippo Mamado used Fire Fang.

Yveltal Ave Mamadisima теряет 3 здоровья.

Yveltal Ave Mamadisima промахивается

Hippowdon Hippo Mamado растерянно попадает по себе.

Hippowdon Hippo Mamado теряет 3 здоровья.

Hippowdon Hippo Mamado теряет сознание.

В команде фиолетовых не осталось покемонов.

Команда зеленых побеждает в этом бою!

[s372819@helios ~/lab2]\$

Вывод

Для выполнения работы были применены возможности ООП на языке Java. Основой для написанных классов является пакет `ru.ifmo.se.pokemon`, от которого был унаследован класс `Pokemon` для 6 созданных классов покемонов, а также остальные классы пакета для классов атак. Использовал ключевые слова `import` и `package` а также использовал ключевые слова `public`, `private`, `this` и `super` внутри создаваемых классов.

Список литературы

А.В. Гаврилов, С.В. Клименков, Ю.А. Королёва, А.Е. Харитонова, Е.А. Цопа (2019) ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ JAVA – СПб: Университет ИТМО.

Письмак А.Е. (2022) Программирование. Конспект лекций – СПб: Университет ИТМО.

Metanit.com (2018) Классы. Объектно-ориентированное программирование.
<https://metanit.com/java/tutorial/3.1.php>