

НИУ ИТМО
Факультет программной инженерии и компьютерной
техники

Отчет по Лабораторной работе №2
по курсу Программирование

Вариант №178843

Выполнил:
Студент группы Р3117
Дегтев Артемий

Санкт-Петербург
2021

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.

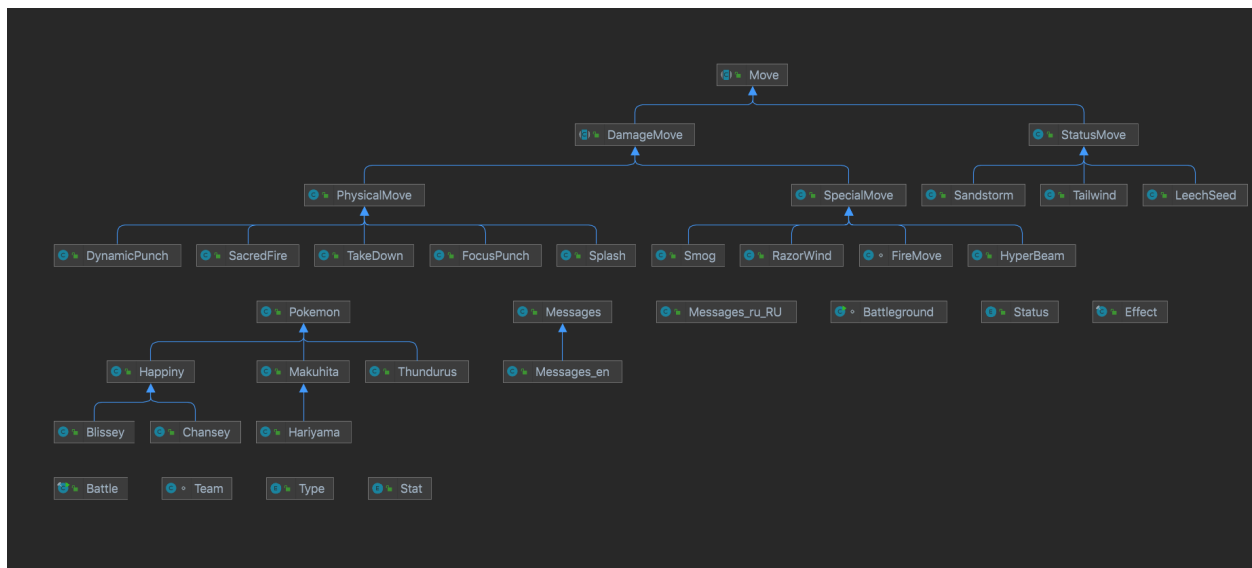
Базовые классы и симулятор сражения находятся в [jar-архиве](#) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](#).

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

Ваши покемоны:

<p>Thundurus</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Hyper Beam✓ Razor Wind✓ Splash✓ Tailwind	<p>Makuhita</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Leech Seed✓ Sandstorm✓ Hyper Beam	<p>Hariyama</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Leech Seed✓ Sandstorm✓ Hyper Beam✓ Sacred Fire	<p>Happiny</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Take Down✓ Focus Punch	<p>Chansey</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Take Down✓ Focus Punch✓ Smog	<p>Blissey</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Take Down✓ Focus Punch✓ Smog✓ Dynamic Punch
---	---	---	--	---	---

Диаграмма классов



Исходный код программы

Pokemons

```
package pokemons;

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

public class Blissey extends Happy {
    public Blissey(String name, int level) {
        super(name, level);
        setStats(v: 255, v1: 10, v2: 10, v3: 75, v4: 135, v5: 55);
        setType(Type.NORMAL);
        addMove(new DynamicPunch());
    }
}
```

```
package pokemons;

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

public class Makuhita extends Pokemon {
    public Makuhita(String name, int level) {
        super(name, level);
        setStats(v: 72, v1: 60, v2: 30, v3: 20, v4: 30, v5: 25);
        setType(Type.FIGHTING);
        setMove(new LeechSeed(), new Sandstorm(), new HyperBeam());
    }
}
```

```

package pokemons;

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

public class Chansey extends Happiny {
    public Chansey(String name, int level) {
        super(name, level);
        setStats(v: 250, v1: 5, v2: 5, v3: 35, v4: 105, v5: 50);
        setType(Type.NORMAL);
        addMove(new Smog());
    }
}

```

```

package pokemons;

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

public class Happiny extends Pokemon {
    public Happiny(String name, int level) {
        super(name, level);
        setStats(v: 100, v1: 5, v2: 5, v3: 15, v4: 65, v5: 30);
        setType(Type.NORMAL);
        setMove(new TakeDown(), new FocusPunch());
    }
}

```

```

package pokemons;

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

public class Thundurus extends Pokemon {
    public Thundurus(String name, int level) {
        super(name, level);
        setStats(v: 79, v1: 115, v2: 70, v3: 125, v4: 80, v5: 111);
        setType(Type.FAIRY, Type.ELECTRIC);
        setMove(new Tailwind(), new RazorWind(), new Splash(), new HyperBeam());
    }
}

```

```

package pokemons;

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

public class Hariyama extends Makuhita {
    public Hariyama(String name, int level) {
        super(name, level);
        setStats(v: 144, v1: 120, v2: 60, v3: 40, v4: 60, v5: 50);
        setType(Type.FIGHTING);
        addMove(new SacredFire());
    }
}

```

Moves

```
package moves;

import ru.ifmo.se.pokemon.*;

public class DynamicPunch extends PhysicalMove {
    public DynamicPunch() {
        super(Type.FIGHTING, v: 100, v1: 50);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        Effect.confuse(p);
    }

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

```
package moves;

import ru.ifmo.se.pokemon.*;

public class FocusPunch extends PhysicalMove {
    public FocusPunch() {
        super(Type.FIGHTING, v: 150, v1: 100);
    }

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

```
package moves;

import ru.ifmo.se.pokemon.*;

public class HyperBeam extends SpecialMove {
    public HyperBeam() { super(Type.NORMAL, v: 150, v1: 90); }

    @Override
    protected void applySelfEffects(Pokemon p) { p.setCondition(new Effect().condition(Status.SLEEP).turns(2)); }

    @Override
    protected String describe() { return "применяет атаку Hyper Beam"; }
}
```

```
package moves;

import ru.ifmo.se.pokemon.*;

public class LeechSeed extends StatusMove {
    public LeechSeed() { super(Type.GRASS, v: 0, v1: 90); }

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

```

package moves;

import ru.ifmo.se.pokemon.*;

public class RazorWind extends SpecialMove {
    public RazorWind() { super(Type.NORMAL, v: 80, v1: 100); }

    @Override
    protected double calcCriticalHit(Pokemon att, Pokemon def) {
        if (Math.random() < att.getStat(Stat.SPEED) * 3 / 512)
            return 2;
        return 1;
    }

    @Override
    protected String describe() { return "применяет атаку Razor Wind"; }
}

```

```

package moves;

import ru.ifmo.se.pokemon.*;

public class SacredFire extends PhysicalMove {
    public SacredFire() { super(Type.FIRE, v: 100, v1: 95); }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() < 0.5) {
            Effect.burn(p);
        }
    }

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

```

```

package moves;

import ru.ifmo.se.pokemon.*;

public class Sandstorm extends StatusMove {
    public Sandstorm() { super(Type.GROUND, v: 0, v1: 0); }

    @Override
    protected void applySelfEffects(Pokemon p) {
        Effect e = new Effect().turns(5).stat(Stat.HP, i: -1/16);
        p.addEffect(e);
    }

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

```

```

package moves;

import ru.ifmo.se.pokemon.*;

public class Smog extends SpecialMove {
    public Smog() { super(Type.POISON, v: 30, v1: 70); }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if (Math.random() < 0.4) {
            Effect.poison(p);
        }
    }

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

```

```

package moves;

import ru.ifmo.se.pokemon.*;

public class Splash extends PhysicalMove {
    public Splash() { super(Type.NORMAL, v: 0, v1: 0); }

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

```

```

package moves;
import ru.ifmo.se.pokemon.*;
public class Tailwind extends StatusMove{
    public Tailwind() { super(Type.FLYING, v: 0, v1: 0); }

    @Override
    protected void applySelfEffects(Pokemon p) {
        Effect e = new Effect().turns(4).stat(Stat.SPEED, i: 2);
        p.addEffect(e);
    }

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

```

```

package moves;
import ru.ifmo.se.pokemon.*;
public class TakeDown extends PhysicalMove{
    public TakeDown(){
        super(Type.NORMAL, v: 90, v1: 85);
    }
    @Override
    protected void applySelfDamage(Pokemon att, double damage){
        att.setMod(Stat.HP, (int) (damage/4));
    }
    @Override
    protected String describe(){
        return "использует Take Down";
    }
}

```

Класс main

```

import pokemons.*;
import ru.ifmo.se.pokemon.Battle;

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

        Hariyama p1 = new Hariyama( name: "", level: 1);
        Happiny p2 = new Happiny( name: "", level: 2);
        Chansey p3 = new Chansey( name: "", level: 2);

        Thundurus p4 = new Thundurus( name: "", level: 1);
        Makuhita p5 = new Makuhita( name: "", level: 3);
        Blissey p6 = new Blissey( name: "", level: 1);

        b.addAlly(p1);
        b.addAlly(p2);
        b.addAlly(p3);

        b.addFoe(p4);
        b.addFoe(p5);
        b.addFoe(p6);

        b.go();
    }
}

```


Результаты работы программы

Hariyama из команды синих вступает в бой!

Thundurus из команды фиолетовых вступает в бой!

Thundurus промахивается

Hariyama промахивается

Thundurus применяет атаку Razor Wind.

Hariyama теряет 6 здоровья.

Hariyama применяет атаку Hyper Beam.

Thundurus теряет 6 здоровья.

Hariyama засыпает

Thundurus применяет атаку Hyper Beam.

Hariyama теряет 7 здоровья.

Thundurus засыпает

Hariyama использует Sacred Fire.

Thundurus теряет 7 здоровья.

Thundurus теряет сознание.

Makuhita из команды фиолетовых вступает в бой!

Makuhita использует Leech Seed.

Hariyama использует Leech Seed.

Makuhita использует Leech Seed.

Hariyama промахивается

Makuhita промахивается

Hariyama промахивается

Makuhita промахивается

Hariyama промахивается

Makuhita применяет атаку Hyper Beam.

Hariyama теряет 7 здоровья.

Makuhita засыпает

Hariyama теряет сознание.

Happiny из команды синих вступает в бой!

Makuhita применяет атаку Hyper Beam.

Happiny теряет 7 здоровья.

Happiny использует Focus Punch.

Makuhita теряет 7 здоровья.

Makuhita использует Leech Seed.

Happiny использует Take Down.

Makuhita теряет 10 здоровья.

Happiny теряет 2 здоровья.

Makuhita промахивается

Happiny использует Take Down.

Makuhita теряет 8 здоровья.

Happiny теряет 2 здоровья.

Makuhita теряет сознание.

Blissey из команды фиолетовых вступает в бой!

Happiny использует Take Down.

```
Blissey теряет 6 здоровья.  
Happiny теряет 1 здоровья.
```

```
Blissey использует Focus Punch.  
Happiny теряет 10 здоровья.  
Happiny теряет сознание.  
Chansey из команды синих вступает в бой!  
Chansey использует Take Down.  
Blissey теряет 8 здоровья.  
Chansey теряет 2 здоровья.
```

```
Blissey использует Focus Punch.  
Chansey теряет 16 здоровья.
```

```
Chansey использует Focus Punch.  
Blissey теряет 19 здоровья.  
Blissey теряет сознание.  
В команде фиолетовых не осталось покемонов.  
Команда синих побеждает в этом бою!
```

```
Process finished with exit code 0
```

Выводы

В процессе выполнения данной лабораторной работы я научился использовать ООП в своих проектах, научился создавать собственные классы, наследовать их и переопределять методы. Научился интегрировать jar-архивы в мои проекты.