

Санкт-Петербургский Национальный Исследовательский

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

По программированию

Вариант 311813

Выполнил:

Студент группы Р3117

Васильченко Роман Антонович

Преподаватель:

Письмак А.Е.



Санкт-Петербург

2021

Оглавление

Задание	3
Основные этапы вычисления	4
UML Диаграмма.....	4
Attacks	4
Pokemons	8
Lab2	9
Вывод программы.....	10
Список источников.....	14
Вывод	14

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>

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.

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

Введите вариант:

Ваши покемоны:

Tornadus

Атаки:
✓ Withdraw
✓ Sweet Scent
✓ Rock Slide
✓ Venoshock

Glameow

Атаки:
✓ Confide
✓ Swagger
✓ Double Team

Purugly

Атаки:
✓ Confide
✓ Swagger
✓ Double Team
✓ Fire Blast

Scatterbug

Атаки:
✓ Swagger
✓ Rest

Spewpa

Атаки:
✓ Swagger
✓ Rest
✓ Sludge

Vivillon

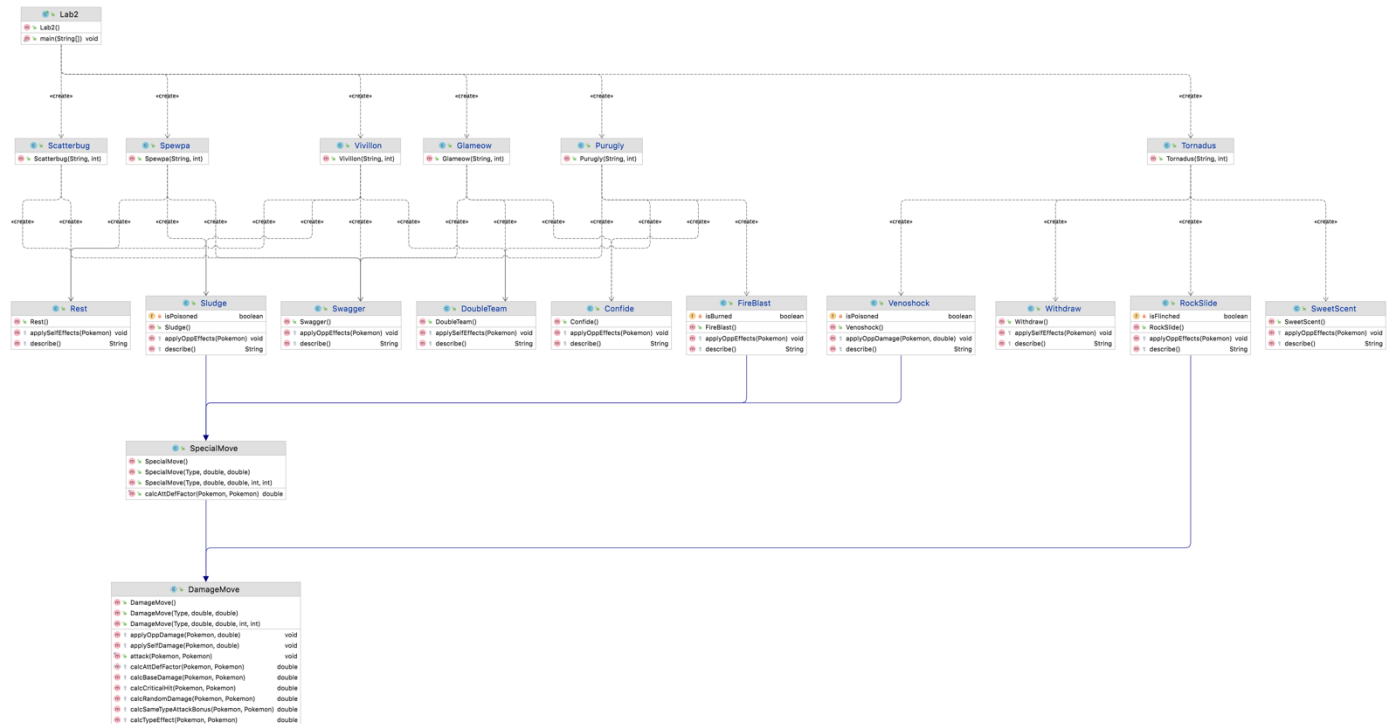
Атаки:
✓ Swagger
✓ Rest
✓ Sludge
✓ Double Team

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов реализованной объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

Основные этапы вычисления

UML Диаграмма



Attacks

```
package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;
```

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

    @Override
    protected void applyOppEffects(Pokemon p) {
        p.setMod(Stat.SPECIAL_ATTACK, -1);
    }

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

```
package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;
```

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

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

```

    }

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

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

public class FireBlast extends SpecialMove {
    public FireBlast(){
        super(Type.FIRE,110,85);
    }
    private boolean isBurned = false;
    @Override
    protected void applyOppEffects(Pokemon p){
        if(Math.random() <= 0.1){
            isBurned = true;
            Effect.burn(p);
        }
    }

    @Override
    protected String describe() {
        return "Использует Fire Blast " + ((isBurned) ? "и burn цель" : "");
    }
}

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

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

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

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

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

public class RockSlide extends PhysicalMove {
    public RockSlide(){
        super(Type.ROCK, 75, 90);
    }
    private boolean isFlinched = false;
    @Override
    protected void applyOppEffects(Pokemon p) {
        if(Math.random() < 0.3){
            isFlinched = true;
            Effect.flinch(p);
        }
    }

    @Override
    protected String describe() {
        return "Использовал RockSlide " + ((isFlinched) ? "и flinch цель" : "");
    }
}

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

public class Sludge extends SpecialMove {
    public Sludge(){
        super(Type.POISON, 65,100);
    }

    private boolean isPoisoned = false;
    @Override
    protected void applyOppEffects(Pokemon p) {
        if(Math.random() < 0.3){
            Effect.poison(p);
            isPoisoned = true;
        }
    }

    @Override
    protected String describe() {
        return "использует Sludge " + ((isPoisoned) ? "и poison цель" : "");
    }
}

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

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

    @Override
    protected void applyOppEffects(Pokemon p) {
        p.setMod(Stat.ATTACK, 2);
    }
}

```

```

        Effect.confuse(p);
    }

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

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

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

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

    @Override
    protected String describe() {
        return "Использует Sweet Scent ";
    }
}

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

public class Venoshock extends SpecialMove {
    public Venoshock(){
        super(Type.POISON, 65,100);
    }

    private boolean isPoisoned = false;
    @Override
    protected void applyOppDamage(Pokemon p, double v) {
        Status cond = p.getCondition();
        isPoisoned = cond.equals(Status.POISON);
        if(isPoisoned){
            p.setMod(Stat.HP, -2*(int)Math.round(v));
        }
    }

    @Override
    protected String describe() {
        return "использует Venoshock " + ((isPoisoned) ? "и урон x2, так как цель отравлена" : "");
    }
}

```

```

package ru.roman.vassilchenko.attacks;
import ru.ifmo.se.pokemon.*;

public class Withdraw extends StatusMove {
    public Withdraw(){
        super(Type.WATER,0,0);
    }
    @Override
    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.DEFENSE,1);
    }

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

```

Pokemons

```

package ru.roman.vassilchenko.pokemons;
import ru.ifmo.se.pokemon.*;
import ru.roman.vassilchenko.attacks.*;

public class Glameow extends Pokemon {
    public Glameow(String name, int level){
        super(name,level);
        setStats(49,55,42,42,37,85);
        setType(Type.NORMAL);
        setMove(new Confide(), new Swagger(), new DoubleTeam());
    }
}

```

```

package ru.roman.vassilchenko.pokemons;
import ru.ifmo.se.pokemon.*;
import ru.roman.vassilchenko.attacks.*;

public class Purugly extends Pokemon {
    public Purugly(String name, int level){
        super(name, level);
        setStats(71,82,64,64,59,112);
        setType(Type.NORMAL);
        setMove(new Confide(), new Swagger(),new DoubleTeam(), new FireBlast());
    }
}

```

```

package ru.roman.vassilchenko.pokemons;
import ru.roman.vassilchenko.attacks.*;
import ru.ifmo.se.pokemon.*;

public class Scatterbug extends Pokemon {
    public Scatterbug(String name, int level){
        super(name,level);
        setStats(38,35,40,27,25,35);
    }
}

```



```

        setType(Type.BUG);
        setMove(new Swagger(), new Rest());
    }
}

```

```

package ru.roman.vassilchenko.pokemons;
import ru.roman.vassilchenko.attacks.*;
import ru.ifmo.se.pokemon.*;

public class Spewpa extends Pokemon {
    public Spewpa(String name, int level){
        super(name, level);
        setStats(45, 22, 60, 27, 30, 29);
        setType(Type.BUG);
        setMove(new Swagger(), new Rest(), new Sludge());
    }
}

```

```

package ru.roman.vassilchenko.pokemons;
import ru.ifmo.se.pokemon.*;
import ru.roman.vassilchenko.attacks.*;

public class Tornadus extends Pokemon {
    public Tornadus(String name, int level){
        super(name, level);
        setStats(79, 115, 70, 125, 80, 111);
        setType(Type.FLYING);
        setMove(new Withdraw(), new SweetScent(), new RockSlide(), new Venoshock());
    }
}

```

```

package ru.roman.vassilchenko.pokemons;
import ru.roman.vassilchenko.attacks.*;
import ru.ifmo.se.pokemon.*;

public class Vivillon extends Pokemon {
    public Vivillon(String name, int level){
        super(name, level);
        setStats(80, 52, 50, 90, 50, 89);
        setType(Type.BUG);
        setMove(new Swagger(), new Rest(), new Sludge(), new DoubleTeam());
    }
}

```

Lab2

```

package ru.roman.vassilchenko;

import ru.ifmo.se.pokemon.*;
import ru.roman.vassilchenko.pokemons.*;

public class Lab2 {

```

```

public static void main(String[] args) {
    Battle b = new Battle();
    Glameow p1 = new Glameow("Кот",1);
    Purugly p2 = new Purugly("Большой кот",1);
    Scatterbug p3 = new Scatterbug("Букашка",1);
    Spewpa p4 = new Spewpa("Большая букашка",1);
    Tornadus p5 = new Tornadus("Какой-то Джин", 1);
    Vivillon p6 = new Vivillon("Бабочка",1);

    b.addAlly(p1);
    b.addAlly(p2);
    b.addAlly(p3);
    b.addFoe(p4);
    b.addFoe(p5);
    b.addFoe(p6);
    b.go();
}
}

```

Вывод программы

Glameow Кот from the team Blue enters the battle!

Spewpa Большая букашка from the team Yellow enters the battle!

Glameow Кот misses

Spewpa Большая букашка использует Swagger .

Glameow Кот increases attack.

Glameow Кот использует Swagger .

Spewpa Большая букашка increases attack.

Spewpa Большая букашка использует Sludge .

Glameow Кот loses 3 hit points.

Glameow Кот is poisoned

Glameow Кот loses 1 hit points.

Glameow Кот hits himself in confusion.

Glameow Кот loses 3 hit points.

Spewpa Большая букашка hits himself in confusion.

Spewpa Большая букашка loses 3 hit points.

Glameow Кот loses 1 hit points.

Glameow Кот hits himself in confusion.

Glameow Кот loses 3 hit points.

Spewpa Большая букашка использует Sludge и poison цель.

Glameow Кот loses 3 hit points.

Glameow Кот faints.

Purugly Большой кот from the team Blue enters the battle!

Purugly Большой кот misses

Spewpa Большая букашка hits himself in confusion.

Spewpa Большая букашка loses 5 hit points.

Purugly Большой кот использует Swagger .

Spewpa Большая букашка increases attack.

Spewpa Большая букашка использует Swagger .

Purugly Большой кот increases attack.

Purugly Большой кот hits himself in confusion.

Purugly Большой кот loses 4 hit points.

Spewpa Большая букашка использует Swagger .

Purugly Большой кот increases attack.

Purugly Большой кот hits himself in confusion.

Purugly Большой кот loses 3 hit points.

Spewpa Большая букашка использует Swagger .

Purugly Большой кот increases attack.

Purugly Большой кот использует Swagger .

Spewpa Большая букашка increases attack.

Spewra Большая букашка использует Swagger .

Purugly Большой кот increases attack.

Purugly Большой кот использует Swagger .

Spewra Большая букашка increases attack.

Spewra Большая букашка misses

Purugly Большой кот hits himself in confusion.

Purugly Большой кот loses 3 hit points.

Spewra Большая букашка hits himself in confusion.

Spewra Большая букашка loses 5 hit points.

Spewra Большая букашка faints.

Tornadus Какой-то Джин from the team Yellow enters the battle!

Purugly Большой кот Использует Fire Blast .

Tornadus Какой-то Джин loses 6 hit points.

Tornadus Какой-то Джин использует Venoshock .

Purugly Большой кот Использует Fire Blast .

Tornadus Какой-то Джин loses 7 hit points.

Tornadus Какой-то Джин faints.

Vivillon Бабочка from the team Yellow enters the battle!

Purugly Большой кот misses

Vivillon Бабочка использует Sludge .

Purugly Большой кот loses 3 hit points.

Purugly Большой кот faints.

Scatterbug Букашка from the team Blue enters the battle!

Vivillon Бабочка misses

Scatterbug Букашка misses

Vivillon Бабочка использует Swagger .

Scatterbug Букашка increases attack.

Scatterbug Букашка использует Swagger .

Vivillon Бабочка increases attack.

Vivillon Бабочка misses

Scatterbug Букашка misses

Vivillon Бабочка использует Swagger .

Scatterbug Букашка increases attack.

Scatterbug Букашка использует Swagger .

Vivillon Бабочка increases attack.

Vivillon Бабочка hits himself in confusion.

Vivillon Бабочка loses 5 hit points.

Scatterbug Букашка использует Swagger .

Vivillon Бабочка increases attack.

Vivillon Бабочка использует Sludge .

Scatterbug Букашка loses 5 hit points.

Scatterbug Букашка hits himself in confusion.

Scatterbug Букашка loses 5 hit points.

Vivillon Бабочка hits himself in confusion.

Vivillon Бабочка loses 5 hit points.

Scatterbug Букашка использует Swagger .

Vivillon Бабочка increases attack.

Vivillon Бабочка hits himself in confusion.

Vivillon Бабочка loses 4 hit points.

Vivillon Бабочка faints.

Team Yellow loses its last Pokemon.

The team Blue wins the battle!

Список источников

1) <https://se.ifmo.ru/~tony/doc/>

2) <http://pokemondb.net>

Вывод

Во время выполнения лабораторной работы я выяснил, как упаковывать множество классов в один jar архив и как подключать дополнительные модули. Также использовал наследование, чтобы дать классам моих покемонов свойства из jar архива Pokemon.jar