

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине
‘ПРОГРАММИРОВАНИЕ’

Вариант № 311802

Выполнил:

Студент группы Р3118

Павлов Александр

Сергеевич

Преподаватель:

Письмак Алексей

Евгеньевич



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2021

Оглавление

Задание:.....	3
Диаграмма классов реализованной объектной модели:	3
Исходный код программы:	4
Результат работы программы:	9
Вывод:	10

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 **PhysicalNew**, **SpecialNew** и **StatNew** реализовать свои классы для заданных видов атак. Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атак. Назначить каждому виду покемона атаку 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 [zip-архиве](#) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате `javaDoc` - [здесь](#).

Информацию о покемонах, цепочка эволюции и атаки можно найти на сайтах <http://pokein.ru/wiki/1>, <http://pokemondb.net/>, <http://www.kit.com/pokedex/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, обратив особое внимание на классы **Pokemon** и **New**. 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. Запустить минимально работающую программу и посмотреть как она работает:

```
Battle b = new Battle();
Pokemon p1 = new Pokemon("Чуири", 1);
Pokemon p2 = new Pokemon("Хинге", 1);
b.add(p1);
b.add(p2);
b.go();
```

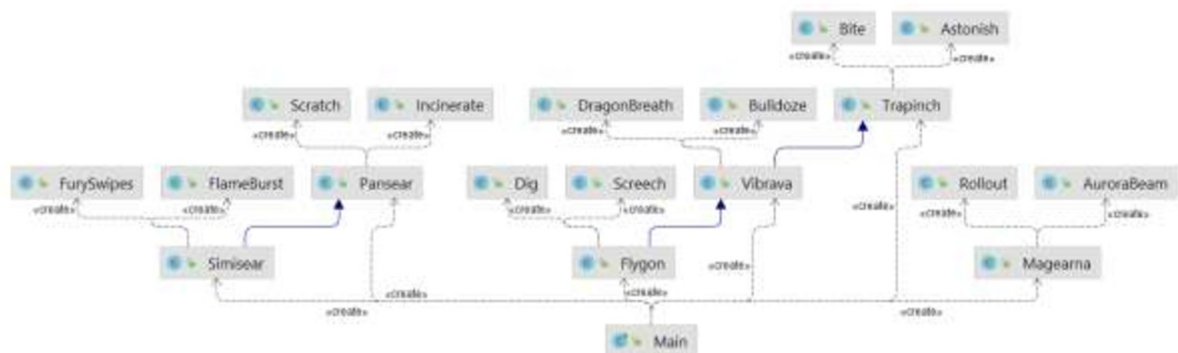
4. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса **Pokemon**. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
5. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса **PhysicalNew** или **SpecialNew**. 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 (последний **StatusNew**), скорее всего придется разобраться с классом **Effect**. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
7. Дополнить все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Введите вариант: 311502

Ваши покемоны:

Magearna	Pansear	Simisear	Trapinch	Vibrava	Flygon
					
Атаки: ✓ Ice Beam ✓ Ice Kick ✓ Blizzard ✓ Rest	Атаки: ✓ Confide ✓ Psychic ✓ Dazzling Gleam	Атаки: ✓ Confide ✓ Psychic ✓ Dazzling Gleam ✓ Baby-Doll Eyes	Атаки: ✓ Psychic ✓ Confuse Ray	Атаки: ✓ Psychic ✓ Confuse Ray ✓ Psywave	Атаки: ✓ Psychic ✓ Confuse Ray ✓ Psywave ✓ Double Team

Диаграмма классов реализованной объектной модели:



Исходный код программы:

Main.java

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

public class Main {

    public static void main(String[] args) {
        Battle b = new Battle();
        Pansear p1 = new Pansear("Чужой", 1);
        Magearna p2 = new Magearna("Хищник", 1);
        Simisear p3 = new Simisear("Кто", 1);
        Trapinch p4 = new Trapinch("Потом", 1);
        Vibrava p5 = new Vibrava("Лень", 1);
        Flygon p6 = new Flygon("Что", 1);
        b.addAlly(p1);
        b.addAlly(p3);
        b.addAlly(p5);
        b.addFoe(p2);
        b.addFoe(p4);
        b.addFoe(p6);
        b.go();
    }
}
```

Flygon.java

```
import moves.Dig;
import moves.Screech;
import ru.ifmo.se.pokemon.Type;

public class Flygon extends Vibrava{
    public Flygon(String name,int level){
        super (name,level);
        setStats(80,100,80,80,80,100);
        setType(Type.GROUND,Type.DRAGON);
        setMove(new Screech(),new Dig());
    }
}
```

Magearna.java

```
import moves.AuroraBeam;
import moves.Rollout;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Magearna extends Pokemon {

    public Magearna(String name, int level) {
        super(name,level);
        setStats(80,95,115,130,115,65);
        setMove(new Rollout(),new AuroraBeam());
        setType(Type.STEEL,Type.FAIRY);
    }
}
```

Pansear.java

```
import moves.Incinerate;
import moves.Scratch;
import ru.ifmo.se.pokemon.*;

public class Pansear extends Pokemon {
```

```

    public Pansear(String name, int level){
        super(name, level);
        setStats(50, 53, 48, 53, 48, 64);
        setMove(new Scratch(), new Incinerate());
        setType(Type.FIRE);
    }
}

```

Simisear.java

```

import moves.FlameBurst;
import moves.FurySwipes;
import ru.ifmo.se.pokemon.Type;

public class Simisear extends Pansear {
    public Simisear(String name, int level) {
        super(name, level);
        setStats(75, 98, 63, 98, 63, 101);
        setType(Type.FIRE);
        setMove(new FlameBurst(), new FurySwipes());
    }
}

```

Trapinch.java

```

import moves.Astonish;
import moves.Bite;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Trapinch extends Pokemon {
    public Trapinch(String name, int level) {
        super(name, level);
        setStats(45, 100, 45, 45, 45, 10);
        setType(Type.GROUND);
        setMove(new Astonish(), new Bite());
    }
}

```

Vibrava.java

```

import moves.Bulldoze;
import moves.DragonBreath;
import ru.ifmo.se.pokemon.Type;

public class Vibrava extends Trapinch{
    public Vibrava(String name, int level){
        super(name, level);
        setStats(50, 70, 50, 50, 50, 70);
        setType(Type.GROUND, Type.DRAGON);
        setMove(new Bulldoze(), new DragonBreath());
    }
}

```

Astonish.java

```

package moves;

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

public class Astonish extends PhysicalMove {
    public Astonish(){
        super(Type.GHOST, 30, 100);
    }
}

```

```

    @Override
    public String describe(){
        return "использует Astonish";
    }
}

```

AuroraBeam.java

```

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

public class AuroraBeam extends SpecialMove {
    public AuroraBeam(){
        super(Type.ICE, 65, 100);
    }
    @Override
    public String describe(){
        return "использует Aurora Beam";
    }
}

```

Bite.java

```

package moves;

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

public class Bite extends PhysicalMove {
    public Bite(){
        super(Type.DARK, 60, 100);
    }
    @Override
    public String describe(){
        return "использует Bite";
    }
}

```

Bulldoze.java

```

package moves;

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

public class Bite extends PhysicalMove {
    public Bite(){
        super(Type.DARK, 60, 100);
    }
    @Override
    public String describe(){
        return "использует Bite";
    }
}

```

Dig.java

```

package moves;

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

public class Dig extends PhysicalMove {
    public Dig(){
        super(Type.GROUND, 80, 100);
    }
}

```

```

        @Override
        public String describe(){
            return "использует Dig";
        }
    }
}

```

DragonBreath.java

```

package moves;

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

public class DragonBreath extends SpecialMove {
    public DragonBreath(){
        super(Type.DRAGON, 60, 100);
    }
    @Override
    public String describe(){
        return "использует Dragon Breath";
    }
}

```

FlameBurst.java

```

package moves;

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

public class FlameBurst extends SpecialMove {
    public FlameBurst(){
        super(Type.FIRE, 70, 100);
    }
    @Override
    public String describe(){
        return "использует Flame Burst";
    }
}

```

FurySwipes.java

```

package moves;

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

public class FurySwipes extends PhysicalMove {
    public FurySwipes(){
        super(Type.NORMAL, 18, 80);
    }
    @Override
    public String describe(){
        return "использует Fury Swipes";
    }
}

```

Incinerate.java

```

package moves;

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

public class Incinerate extends SpecialMove {
    public Incinerate(){

```



```

        super(Type.FIRE, 60, 100);
    }
    @Override
    public String describe(){
        return "использует Incinerate";
    }
}

```

Rollout.java

```

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

public class Rollout extends PhysicalMove {
    public Rollout() {
        super(Type.ROCK, 30, 90);
    }
    @Override
    public String describe(){
        return "использует супер пупер Rollout";
    }
}

```

Scratch.java

```

package moves;

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

public class Scratch extends PhysicalMove {
    public Scratch() {
        super(Type.NORMAL, 40, 100);
    }
    @Override
    public String describe(){
        return "козает Scratch";
    }
}

```

Screech.java

```

package moves;

import ru.ifmo.se.pokemon.*;

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

    @Override
    protected void applyOppEffects(Pokemon p){
        Effect e = new Effect().turns(2).stat(Stat.DEFENSE, -1);
        p.addEffect(e);
    }

    @Override
    public String describe() {
        return "использует Screech";
    }
}

```


Результат работы программы:

Pansear Чужой из команды фиолетовых вступает в бой!

Magearna Хищник из команды красных вступает в бой!

Magearna Хищник использует супер пупер Rollout.

Pansear Чужой теряет 6 здоровья.

Pansear Чужой юзает Scratch.

Magearna Хищник теряет 2 здоровья.

Magearna Хищник использует супер пупер Rollout.

Pansear Чужой теряет 8 здоровья.

Pansear Чужой теряет сознание.

Simisear Кто из команды фиолетовых вступает в бой!

Simisear Кто использует Fury Swipes.

Magearna Хищник теряет 2 здоровья.

Magearna Хищник использует супер пупер Rollout.

Simisear Кто теряет 6 здоровья.

Simisear Кто использует Fury Swipes.

Magearna Хищник теряет 1 здоровья.

Magearna Хищник использует Aurora Beam.

Simisear Кто теряет 2 здоровья.

Simisear Кто использует Flame Burst.

Magearna Хищник теряет 17 здоровья.

Magearna Хищник теряет сознание.

Trapinch Потом из команды красных вступает в бой!

Simisear Кто использует Flame Burst.

Trapinch Потом теряет 8 здоровья.

Trapinch Потом использует Bite.

Simisear Кто теряет 3 здоровья.

Simisear Кто использует Fury Swipes.

Trapinch Потом теряет 4 здоровья.

Trapinch Потом использует Bite.

Simisear Кто теряет 4 здоровья.

Simisear Кто теряет сознание.

Vibrava Лень из команды фиолетовых вступает в бой!

Vibrava Лень использует Dragon Breath.

Trapinch Потом теряет 9 здоровья.

Trapinch Потом теряет сознание.

Flygon Что из команды красных вступает в бой!

Flygon Что использует Screech.

Vibrava Лень использует Dragon Breath.

Flygon Что теряет 11 здоровья.

Flygon Что использует Dig.

Vibrava Лень теряет 9 здоровья.

Vibrava Лень использует Bulldoze.

Flygon Что теряет 6 здоровья.

Flygon Что теряет сознание.

В команде красных не осталось покемонов.

Команда фиолетовых побеждает в этом бою!

Вывод:

В ходе данной лабораторной работы я познакомился с основами ООП в Java. Полученные навыки можно будет применить в дальнейшей профессиональной и учебной деятельности.