

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа
по Программированию №2

Преподаватель: Гаврилов А. В.

Выполнил: Щербаков Александр Валерьевич

Группа: Р3110

Вариант: 10872

Санкт-Петербург
2020

Текст задания:

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://pokeuniverse.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

Покемоны:

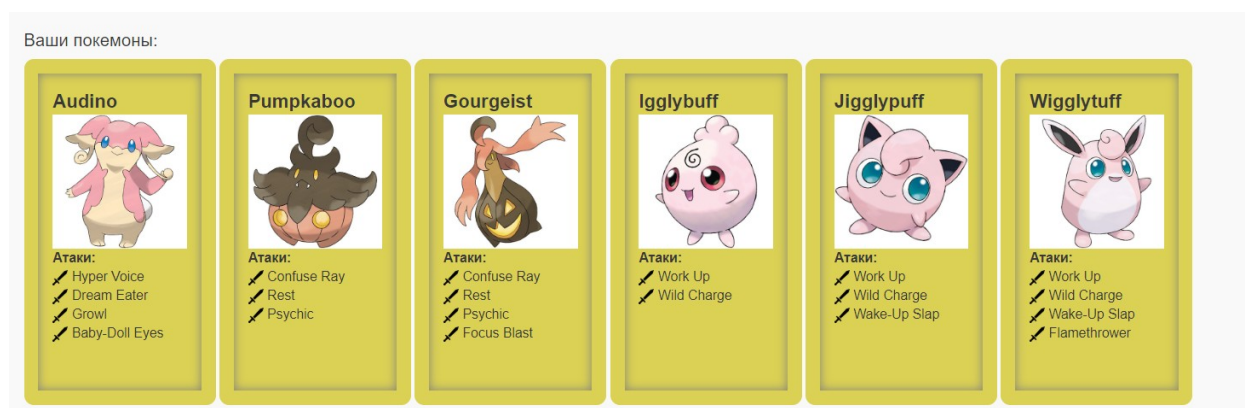
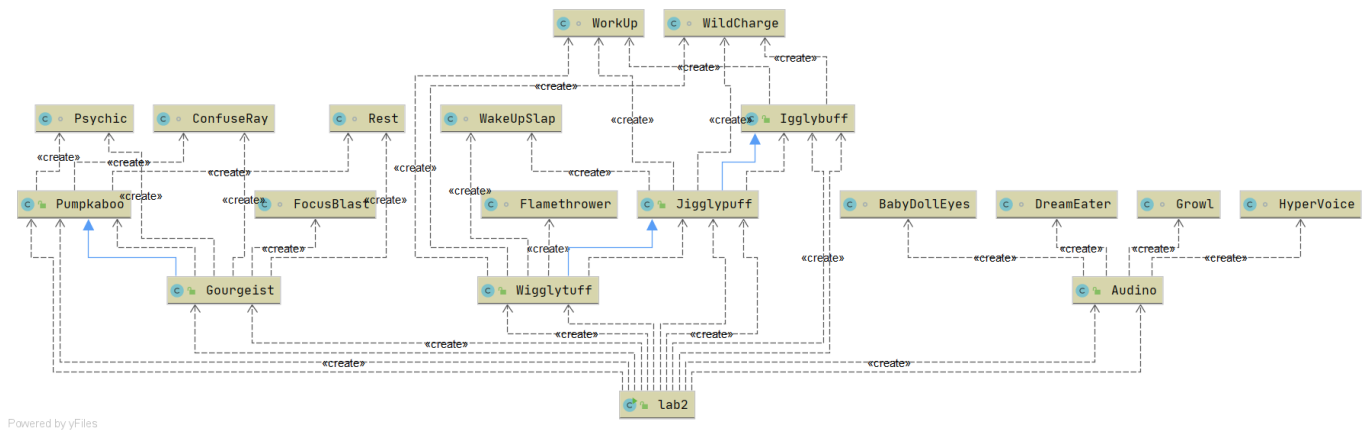


Диаграмма классов реализованной объектной модели:



Исходный код программы:

lab2.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class lab2 {

    public static void main(String[] args) {
        Battle b = new Battle();
        Audino p1 = new Audino("Au", 10);
        Pumpkaboo p2 = new Pumpkaboo("Pu", 10);
        Gourgeist p3 = new Gourgeist("Go", 10);
        Igglybuff p4 = new Igglybuff("Ig", 10);
        Jigglypuff p5 = new Jigglypuff("Ji", 11);
        Wigglytuff p6 = new Wigglytuff("Wi", 9);

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

        b.go();
    }
}

```

Audino.java

```

package lab2;
import ru.ifmo.se.pokemon.*;

public class Audino extends Pokemon {
    public Audino(String name, int lvl) {
        super(name, lvl);
        setStats(103, 60, 86, 60, 86, 50);
        setType(Type.NORMAL);
        setMove(new HyperVoice(), new DreamEater(), new Growl(), new
BabyDollEyes());
    }
}

```

```

class HyperVoice extends SpecialMove {
    public HyperVoice(){
        super(Type.NORMAL, 90, 1.0);
    }
}

class DreamEater extends SpecialMove{
    public DreamEater(){
        super(Type.PSYCHIC, 100, 1.0);
    }

    @Override

    protected boolean checkAccuracy(Pokemon att, Pokemon def){
        if (def.getCondition() != Status.SLEEP){
            return false;
        }
        else return true;
    }

    protected void applySelfEffects(Pokemon p){
        p.setMod(Stat.HP, (int) ((103 - p.getHP())/2));
    }
}

class Growl extends StatusMove{
    public Growl(){
        super(Type.NORMAL, 0, 1.0);
    }

    @Override

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

class BabyDollEyes extends StatusMove{
    public BabyDollEyes(){
        super(Type.FAIRY, 0, 1.0);
    }

    @Override

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

```

Gourgeist.java

```

package lab2;
import ru.ifmo.se.pokemon.*;

public class Gourgeist extends Pumpkaboo {
    public Gourgeist(String name, int lvl){
        super(name, lvl);
        setStats(65,90,122,58,75,84);
        setType(Type.GHOST, Type.GRASS);
        setMove(new ConfuseRay(), new Rest(), new Psychic(), new FocusBlast());
    }
}

```

```

    }
}

class FocusBlast extends SpecialMove{
    public FocusBlast(){
        super(Type.FIGHTING, 120, 0.7);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        if(Math.random() < 0.1){
            p.setMod(Stat.SPECIAL_DEFENSE, -1);
        }
    }
}

```

Igglybuff.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class Igglybuff extends Pokemon{
    public Igglybuff(String name, int lvl){
        super(name, lvl);
        setStats(90,30,15, 40,20,15);
        setType(Type.NORMAL, Type.FAIRY);
        setMove(new WorkUp(), new WildCharge());
    }
}

class WorkUp extends StatusMove {
    public WorkUp() {
        super(Type.NORMAL, 0, 1.0);
    }

    @Override

    protected boolean checkAccuracy(Pokemon att, Pokemon def) {
        return true;
    }

    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.ATTACK, 1);
        p.setMod(Stat.SPECIAL_ATTACK, 1);
    }
}

class WildCharge extends PhysicalMove {
    public WildCharge() {
        super(Type.NORMAL, 90, 1.0);
    }

    @Override

    protected void applySelfDamage(Pokemon att, double damage) {
        att.setMod(Stat.HP, -(int) (0.25 * damage));
    }
}

```

Jigglypuff.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class Jigglypuff extends Iggybuff {
    public Jigglypuff(String name, int lvl){
        super(name, lvl);
        setStats(115, 45, 20, 45, 25, 20);
        setType(Type.NORMAL, Type.FAIRY);
        setMove(new WorkUp(), new WildCharge(), new WakeUpSlap());
    }
}

class WakeUpSlap extends PhysicalMove {
    public WakeUpSlap(){
        super(Type.FIGHTING, 70, 1.0);
    }

    @Override
    protected void applyOppEffects(Pokemon p){
        if (p.getCondition() == Status.SLEEP){
            applyOppDamage(p, 70 * 2);
            Effect e = new Effect().condition(Status.NORMAL);
            p.setCondition(e);
        }
    }
}

```

Lab2.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class lab2 {

    public static void main(String[] args) {
        Battle b = new Battle();
        Audino p1 = new Audino("Au", 10);
        Pumpkaboo p2 = new Pumpkaboo("Pu", 10);
        Gourgeist p3 = new Gourgeist("Go", 10);
        Iggybuff p4 = new Iggybuff("Ig", 10);
        Jigglypuff p5 = new Jigglypuff("Ji", 11);
        Wigglytuff p6 = new Wigglytuff("Wi", 9);

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

        b.go();
    }
}

```

Wigglytuff.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class Wigglytuff extends Jigglypuff {
    public Wigglytuff(String name, int lvl){
        super(name, lvl);
        setStats(140, 70, 45, 85, 50, 45);
        setType(Type.NORMAL, Type.FAIRY);
        setMove(new WorkUp(), new WildCharge(), new WakeUpSlap(), new
Flamethrower());
    }
}

class Flamethrower extends SpecialMove{
    public Flamethrower(){
        super(Type.FIRE, 90, 1.0);
    }

    @Override

    protected void applyOppEffects(Pokemon p){
        if(Math.random() < 0.1){
            Effect.burn(p);
        }
    }
}

```

Pumpkaboo.java

```

package lab2;

import ru.ifmo.se.pokemon.*;

public class Pumpkaboo extends Pokemon {
    public Pumpkaboo(String name, int lvl){
        super(name, lvl);
        setStats(49, 66, 70, 44, 55, 51);
        setType(Type.GHOST, Type.GRASS);
        setMove(new ConfuseRay(), new Rest(), new Psychic());
    }
}

class ConfuseRay extends StatusMove {
    ConfuseRay(){
        super(Type.GHOST, 0, 1.0);
    }

    @Override

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

class Rest extends StatusMove {
    Rest(){
        super(Type.PSYCHIC, 0, 1.0);
    }

    @Override

    protected void applySelfEffects(Pokemon p){
        Effect e = new Effect().turns(2).condition(Status.SLEEP);
    }
}

```

```

        p.setCondition(e);
        p.setMod(Stat.HP, (int)(49 - p.getHP()));
    }
}

class Psychic extends SpecialMove {
    Psychic() {
        super(Type.PSYCHIC, 90, 1.0);
    }

    @Override

    protected void applyOppEffects(Pokemon p){
        if(Math.random() < 0.1) {
            p.setMod(Stat.SPECIAL_DEFENSE, -1);
        }
    }
}

```

Вывод программы:

Audino Au из команды желтых вступает в бой!
 Igglybuff Ig из команды черных вступает в бой!
 Audino Au атакует.
 Igglybuff Ig уменьшает атаку.

Igglybuff Ig атакует.
 Igglybuff Ig увеличивает атаку.
 Igglybuff Ig увеличивает специальную атаку.

Audino Au промахивается

Igglybuff Ig атакует.
 Audino Au теряет 8 здоровья.
 Igglybuff Ig восстанавливает 2 здоровья.

Audino Au атакует.
 Igglybuff Ig уменьшает атаку.

Igglybuff Ig атакует.
 Audino Au теряет 13 здоровья.
 Igglybuff Ig восстанавливает 3 здоровья.

Audino Au атакует.
 Igglybuff Ig уменьшает атаку.
 Igglybuff Ig уменьшает специальную атаку.

Igglybuff Ig атакует.
 Audino Au восстанавливает 1 здоровья.
 Igglybuff Ig восстанавливает 2147483647 здоровья.
 Igglybuff Ig теряет сознание.
 Jigglypuff Ji из команды черных вступает в бой!
 Audino Au промахивается

Jigglypuff Ji атакует.
 Jigglypuff Ji увеличивает атаку.

Jigglypuff Ji увеличивает специальную атаку.

Audino Au промахивается

Jigglypuff Ji атакует.

Audino Au теряет 15 здоровья.

Jigglypuff Ji восстанавливает 3 здоровья.

Audino Au атакует.

Jigglypuff Ji уменьшает атаку.

Jigglypuff Ji атакует.

Jigglypuff Ji увеличивает атаку.

Jigglypuff Ji увеличивает специальную атаку.

Audino Au промахивается

Jigglypuff Ji атакует.

Jigglypuff Ji увеличивает атаку.

Jigglypuff Ji увеличивает специальную атаку.

Audino Au атакует.

Jigglypuff Ji уменьшает атаку.

Jigglypuff Ji атакует.

Audino Au теряет 11 здоровья.

Jigglypuff Ji восстанавливает 2 здоровья.

Audino Au теряет сознание.

Pumpkaboo Pu из команды желтых вступает в бой!

Pumpkaboo Pu атакует.

Jigglypuff Ji не замечает воздействие типа GHOST

Jigglypuff Ji атакует.

Pumpkaboo Pu теряет 1 здоровья.

Pumpkaboo Pu не замечает воздействие типа NORMAL

Pumpkaboo Pu атакует.

Jigglypuff Ji теряет 7 здоровья.

Jigglypuff Ji атакует.

Pumpkaboo Pu теряет 1 здоровья.

Pumpkaboo Pu не замечает воздействие типа FIGHTING

Pumpkaboo Pu атакует.

Jigglypuff Ji теряет 9 здоровья.

Jigglypuff Ji атакует.

Pumpkaboo Pu не замечает воздействие типа NORMAL

Jigglypuff Ji увеличивает атаку.

Jigglypuff Ji увеличивает специальную атаку.

Pumpkaboo Pu атакует.

Pumpkaboo Pu засыпает
Pumpkaboo Pu теряет 19 здоровья.

Jigglypuff Ji атакует.
Pumpkaboo Pu теряет 1 здоровья.
Pumpkaboo Pu не замечает воздействие типа FIGHTING

Pumpkaboo Pu атакует.
Pumpkaboo Pu теряет 39 здоровья.
Pumpkaboo Pu теряет сознание.
Gourgeist Go из команды желтых вступает в бой!
Gourgeist Go атакует.
Gourgeist Go засыпает
Gourgeist Go теряет 14 здоровья.

Jigglypuff Ji атакует.
Gourgeist Go теряет 1 здоровья.
Gourgeist Go не замечает воздействие типа FIGHTING

Gourgeist Go промахивается

Jigglypuff Ji атакует.
Gourgeist Go теряет 1 здоровья.
Gourgeist Go не замечает воздействие типа FIGHTING

Gourgeist Go атакует.
Jigglypuff Ji теряет 12 здоровья.

Jigglypuff Ji атакует.
Gourgeist Go теряет 1 здоровья.
Gourgeist Go не замечает воздействие типа NORMAL

Gourgeist Go атакует.
Jigglypuff Ji теряет 8 здоровья.

Jigglypuff Ji атакует.
Gourgeist Go теряет 1 здоровья.
Gourgeist Go не замечает воздействие типа NORMAL

Gourgeist Go атакует.
Gourgeist Go засыпает
Gourgeist Go теряет 32 здоровья.
Gourgeist Go теряет сознание.
В команде желтых не осталось покемонов.
Команда черных побеждает в этом бою!

Process finished with exit code 0

Вывод: в ходе выполнения работы я познакомился с основными принципами и возможностями ООП, научился применять механизмы

наследования в своих программах. Узнал о структуре Java проектов, научился создавать пакеты и подключать к своему проекту JAR файлы в качестве библиотек.