Университет ИТМО

Факультет ПИиКТ

Студент Птицын Максим Евгеньевич

Группа P3130

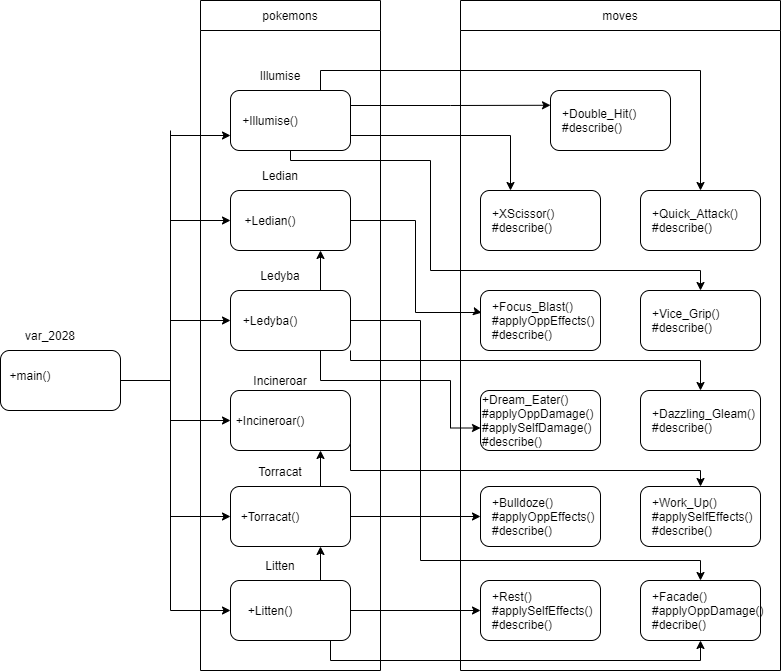
Преподаватель Блохина Елена Николаевна

Лабораторная работа №2.

Вариант № 2028.

г. Санкт-Петербург

2021 г.



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](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

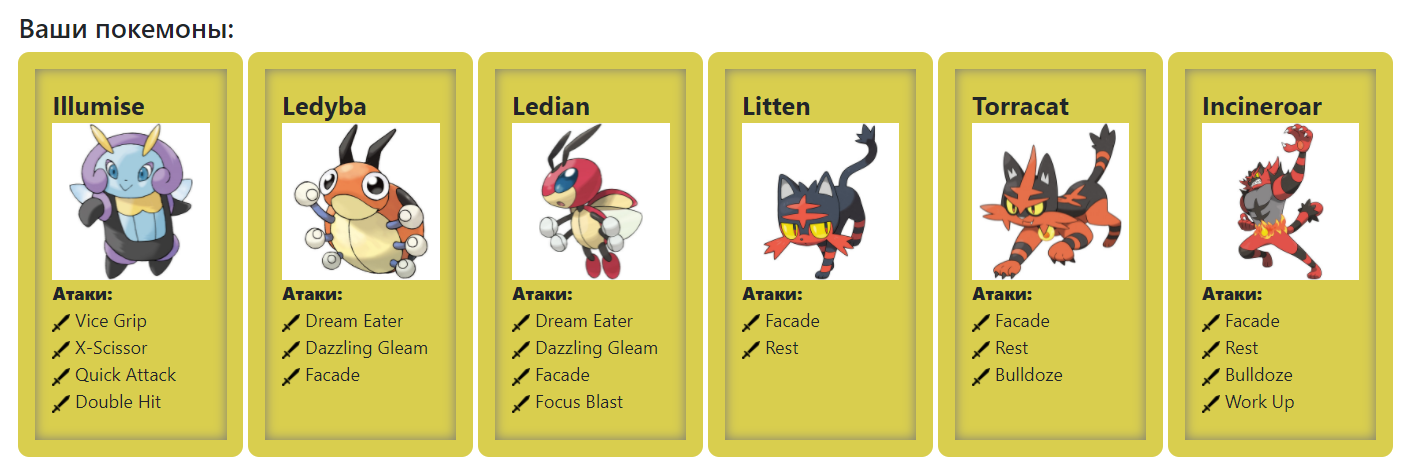
Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>

#### 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](https://se.ifmo.ru/~tony/doc/), 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.
4. Battle b = new Battle();
5. Pokemon p1 = new Pokemon("Чужой", 1);
6. Pokemon p2 = new Pokemon("Хищник", 1);
7. b.addAlly(p1);
8. b.addFoe(p2);
9. b.go();
10. 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.
11. 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.
12. 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.
13. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.



**Исходный код:**

var2028.java

import pokemons.\*;

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

public class var2028 {

public static void main(String args[]) {

Battle b = new Battle();

Incineroar p1 = new Incineroar("", 3);

Illumise p2 = new Illumise("", 1);

Ledian p3 = new Ledian("", 2);

Ledyba p4 = new Ledyba("", 1);

Litten p5 = new Litten("", 1);

Torracat p6 = new Torracat("", 2);

b.addAlly(p1);

b.addAlly(p3);

b.addAlly(p5);

b.addFoe(p2);

b.addFoe(p4);

b.addFoe(p6);

b.go();

}

}

Illumise.java

package pokemons;

import moves.\*;

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

public class Illumise extends Pokemon {

public Illumise(String name, int level) {

super(name,level);

setStats(65, 47, 75, 73, 85, 85);

setType(Type.BUG);

setMove(new Vice\_Grip(), new XScissor(), new Quick\_Attack(), new Double\_Hit());

}

}

Incineroar.java

package pokemons;

import moves.\*;

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

public class Incineroar extends Torracat {

public Incineroar(String name, int level) {

super(name,level);

setStats(

95, 115, 90,

80, 90, 90);

addType(Type.DARK);

setMove(new Work\_Up());

}

}

Ledian.java

package pokemons;

import moves.Focus\_Blast;

import ru.ifmo.se.pokemon.Type;

public class Ledian extends Ledyba {

public Ledian(String name, int level) {

super(name,level);

setType(Type.BUG, Type.FLYING);

setStats(55, 35, 50, 55, 110, 85);

addMove(new Focus\_Blast());

}

}

Ledyba.java

package pokemons;

import moves.\*;

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

public class Ledyba extends Pokemon {

public Ledyba(String name, int level) {

super(name,level);

setStats(40, 20, 30, 40, 80, 55);

setMove(new Dream\_Eater(), new Dazzling\_Gleam(), new Facade());

}

}

Litten.java

package pokemons;

import moves.Facade;

import moves.Rest;

import ru.ifmo.se.pokemon.Pokemon;

import ru.ifmo.se.pokemon.Type;

public class Litten extends Pokemon {

public Litten(String name, int level) {

super(name,level);

setStats(45, 65, 40, 60, 40, 70);

setType((Type.FIRE));

setMove(new Rest(), new Facade());

}

}

Torracat.java

package pokemons;

import moves.Bulldoze;

public class Torracat extends Litten {

public Torracat(String name, int level) {

super(name, level);

setStats(65, 85, 50, 80, 50, 90);

addMove(new Bulldoze());

}

}

Bulldoze.java

package moves;

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

public class Bulldoze extends PhysicalMove {

public Bulldoze() {

super(Type.GROUND, 60, 100);

}

@Override

protected void applyOppEffects(Pokemon pokemon) {

pokemon.setMod(Stat.SPEED, -1);

}

@Override

protected java.lang.String describe() {

return "использует Bulldoze";

}

}

Dazzling\_Gleam.java

package moves;

import ru.ifmo.se.pokemon.SpecialMove;

import ru.ifmo.se.pokemon.Type;

public class Dazzling\_Gleam extends SpecialMove {

public Dazzling\_Gleam() {

super(Type.FAIRY, 80, 100);

}

@Override

protected String describe () {

return "использует Dazzling Gleam";

}

}

Double\_Hit.java

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Type;

public class Double\_Hit extends PhysicalMove {

public Double\_Hit() {

super(Type.NORMAL, 35, 90, 0, 2);

}

@Override

protected String describe() {

return "использует Double Hit";

}

}

Dream\_Eater.java

package moves;

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

public class Dream\_Eater extends SpecialMove {

double last\_damage;

public Dream\_Eater() {

super(Type.PSYCHIC, 100, 100);

}

@Override

protected void applyOppDamage(Pokemon def, double damage) {

last\_damage = damage;

Status def\_poke\_stat = def.getCondition();

if (def\_poke\_stat.equals(Status.SLEEP)) {

super.applyOppDamage(def, damage);

}

else {

super.applyOppDamage(def, 0);

}

}

@Override

protected void applySelfDamage(Pokemon p, double damage) {

double damage1=this.last\_damage;

super.applySelfDamage(p, damage1/(-2));

}

@Override

protected String describe() {

return "использует Dream Eater";

}

}

Facade.java

package moves;

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

public class Facade extends PhysicalMove {

public Facade() {

super(Type.NORMAL, 70, 100);

}

@Override

protected java.lang.String describe() {

return "использует Facade";

}

@Override

protected void applyOppDamage(Pokemon def, double damage) {

Status def\_pok\_status = def.getCondition();

if (def\_pok\_status.equals(Status.BURN) | def\_pok\_status.equals(Status.PARALYZE) | def\_pok\_status.equals(Status.POISON)) {

super.applyOppDamage(def, damage\*2);

}

else {

super.applyOppDamage(def, damage);

}

}

}

Focus\_Blast.java

package moves;

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

public class Focus\_Blast extends SpecialMove {

public Focus\_Blast() {

super(Type.FIGHTING, 120, 70);

}

@Override

protected void applyOppEffects(Pokemon p){

Effect e = new Effect().chance(0.1).turns(-1).stat(Stat.SPECIAL\_DEFENSE, -1);

p.addEffect(e);

}

@Override

protected String describe() {

return "использует Focus Blast";

}

}

Quick\_Attack.java

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Type;

public class Quick\_Attack extends PhysicalMove {

public Quick\_Attack() {

super(Type.NORMAL, 40, 100, +1, 1);

}

@Override

protected java.lang.String describe() {

return "использует Quick Attack";

}

}

Rest.java

package moves;

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

public class Rest extends StatusMove {

public Rest() {

super(Type.PSYCHIC,0,0);

}

@Override

protected void applySelfEffects(Pokemon pokemon){

Effect e = new Effect().turns(2).condition(Status.SLEEP);

pokemon.addEffect(e); //или pokemon.setCondition(e); ??

pokemon.setMod(Stat.HP,999999999);

}

//protected void applySelfEffects(Pokemon pokemon){

// Effect e1 = new Effect().turns(2).condition(Status.SLEEP);

// Effect e2 = new Effect().turns(0).stat(Stat.HP, +9999999);

// pokemon.setCondition(e1);

// pokemon.addEffect(e2);

// }

//getStat(Stat stat) - Вернёт максимальное хп?

//Возвращает эффективное боевое значение характеристики покемона с учетом его уровня и воможных действующих на покемона эффектов.

@Override

protected java.lang.String describe() {

return "использует Rest";

}

}

Vice\_Grip.java

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Type;

public class Vice\_Grip extends PhysicalMove {

public Vice\_Grip() {

super(Type.NORMAL, 55, 100);

}

@Override

protected java.lang.String describe() {

return "использует Vice Grip";

}

}

Work\_Up.java

package moves;

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

public class Work\_Up extends StatusMove {

public Work\_Up() {

super(Type.NORMAL, 0, 0);

}

@Override

protected void applySelfEffects(Pokemon pokemon) {

pokemon.setMod(Stat.ATTACK, +1);

pokemon.setMod(Stat.SPECIAL\_ATTACK, +1);

}

@Override

protected java.lang.String describe() {

return "использует Work Up";

}

}

Xscissor.java

package moves;

import ru.ifmo.se.pokemon.PhysicalMove;

import ru.ifmo.se.pokemon.Type;

public class XScissor extends PhysicalMove {

public XScissor() {

super(Type.BUG, 80, 100);

}

@Override

protected java.lang.String describe() {

return "использует X-Scissor";

}

}

Вывод:

Incineroar из команды желтых вступает в бой!

Illumise из команды белых вступает в бой!

Incineroar промахивается

Illumise использует Vice Grip.

Incineroar теряет 3 здоровья.

Illumise использует Quick Attack.

Incineroar теряет 3 здоровья.

Incineroar промахивается

Illumise использует Quick Attack.

Incineroar теряет 3 здоровья.

Incineroar промахивается

Incineroar промахивается

Illumise использует Double Hit.

Incineroar теряет 3 здоровья.

Illumise использует Double Hit.

Incineroar теряет 3 здоровья.

Incineroar промахивается

Illumise использует Vice Grip.

Incineroar теряет 5 здоровья.

Incineroar теряет сознание.

Ledian из команды желтых вступает в бой!

Ledian использует Facade.

Illumise теряет 4 здоровья.

Illumise использует Vice Grip.

Ledian теряет 4 здоровья.

Ledian использует Dazzling Gleam.

Illumise теряет 6 здоровья.

Illumise использует X-Scissor.

Ledian теряет 4 здоровья.

Ledian использует Dream Eater.

Illumise использует Double Hit.

Ledian теряет 3 здоровья.

Illumise использует Double Hit.

Ledian теряет 4 здоровья.

Ledian теряет сознание.

Litten из команды желтых вступает в бой!

Illumise использует Vice Grip.

Litten теряет 5 здоровья.

Litten использует Facade.

Illumise теряет 5 здоровья.

Illumise теряет сознание.

Ledyba из команды белых вступает в бой!

Litten промахивается

Ledyba использует Dream Eater.

Litten использует Facade.

Ledyba теряет 4 здоровья.

Ledyba использует Facade.

Litten теряет 5 здоровья.

Litten использует Facade.

Ledyba теряет 5 здоровья.

Ledyba использует Dream Eater.

Litten промахивается

Ledyba использует Dream Eater.

Litten промахивается

Ledyba использует Dazzling Gleam.

Litten теряет 2 здоровья.

Litten использует Facade.

Ledyba теряет 4 здоровья.

Ledyba теряет сознание.

Torracat из команды белых вступает в бой!

Torracat использует Facade.

Litten теряет 6 здоровья.

Litten теряет сознание.

В команде желтых не осталось покемонов.

Команда белых побеждает в этом бою!