Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по программированию №2

Вариант №31064

Выполнил:  
Студент группы P3106  
Мельник Фёдор Александрович

Проверил:

Вербовой А.А.,

Преподаватель практик ФПиКТ

Санкт-Петербург, 2024

Оглавление

[Задание 3](#_Toc179720864)

[UML 5](#_Toc179720865)

[Исходный код программы 7](#_Toc179720866)

[Main Class 7](#_Toc179720867)

[Pokemons 7](#_Toc179720868)

[Latias 7](#_Toc179720869)

[Sandyghast 8](#_Toc179720870)

[Palossand 8](#_Toc179720871)

[Pichu 8](#_Toc179720872)

[Pikachu 9](#_Toc179720873)

[Raichu 9](#_Toc179720874)

[Moves 9](#_Toc179720875)

[Physical 9](#_Toc179720876)

[Special 11](#_Toc179720877)

[Status 13](#_Toc179720878)

[Результат выполнения программы 15](#_Toc179720879)

[Вывод 17](#_Toc179720880)

# 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.

Battle b = new Battle();

Pokemon p1 = new Pokemon("Чужой", 1);

Pokemon p2 = new Pokemon("Хищник", 1);

b.addAlly(p1);

b.addFoe(p2);

b.go();

1. 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.
2. 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.
3. 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.
4. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

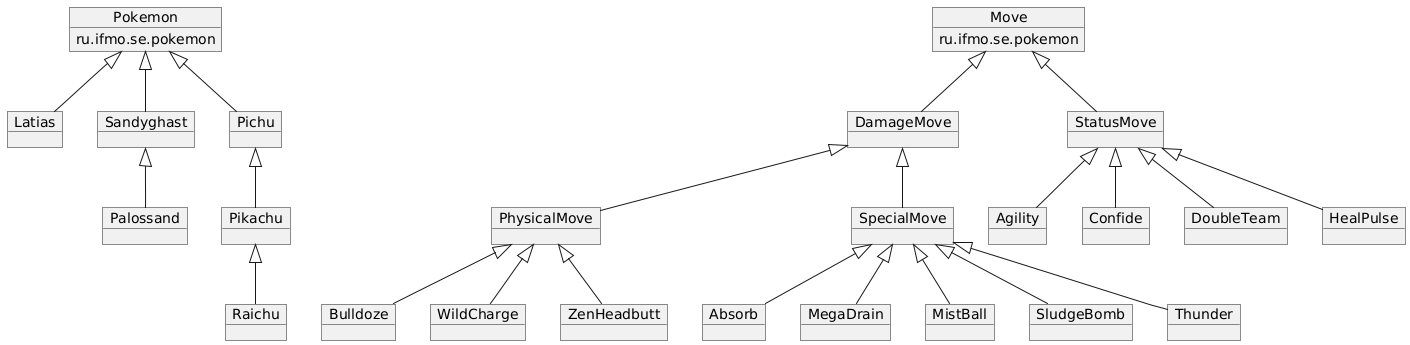
Изображение выглядит как текст, мультфильм

Автоматически созданное описание

# UML

Репозиторий, содержащий UML и код к ней: <https://github.com/ldpst/itmo/tree/main/labs/sem1/prog/lab2/UML>

Диаграмма классов реализованной объектной модели была создана с помощью инструментов PlantUML



Код для диаграммы:

@startuml

object Pokemon

Pokemon : ru.ifmo.se.pokemon

object Latias

object Sandyghast

object Palossand

object Pichu

object Pikachu

object Raichu

object Move

Move : ru.ifmo.se.pokemon

object DamageMove

object StatusMove

object PhysicalMove

object SpecialMove

object Bulldoze

object WildCharge

object ZenHeadbutt

object Agility

object Confide

object DoubleTeam

object HealPulse

object Absorb

object MegaDrain

object MistBall

object SludgeBomb

object Thunder

PhysicalMove <|-- Bulldoze

PhysicalMove <|-- WildCharge

PhysicalMove <|-- ZenHeadbutt

SpecialMove <|-- Absorb

SpecialMove <|-- MegaDrain

SpecialMove <|-- MistBall

SpecialMove <|-- SludgeBomb

SpecialMove <|-- Thunder

StatusMove <|-- Agility

StatusMove <|-- Confide

StatusMove <|-- DoubleTeam

StatusMove <|-- HealPulse

Move <|-- DamageMove

Move <|-- StatusMove

DamageMove <|-- PhysicalMove

DamageMove <|-- SpecialMove

Pokemon <|-- Latias

Pokemon <|-- Sandyghast

Pokemon <|-- Pichu

Sandyghast <|-- Palossand

Pichu <|-- Pikachu

Pikachu <|-- Raichu

@enduml

# Исходный код программы

Репозиторий с программной можно найти по ссылке: <https://github.com/ldpst/itmo/tree/main/labs/sem1/prog/lab2>

## Main Class

import pokemons.\*;  
import ru.ifmo.se.pokemon.Battle;  
  
public class PokeLab {  
 public static void main(String[] args) {  
 Latias latias = new Latias("Латиас", 1);  
 Palossand palossand = new Palossand("Палоссэнд", 1);  
 Pichu pichu = new Pichu("Пичу", 1);  
 Pikachu pikachu = new Pikachu("Пикачу", 1);  
 Raichu raichu = new Raichu("Раичу", 1);  
 Sandygast sandygast = new Sandygast("Сэндигаст", 1);  
 Battle battle = new Battle();  
 battle.addAlly(latias);  
 battle.addAlly(pichu);  
 battle.addAlly(raichu);  
 battle.addFoe(palossand);  
 battle.addFoe(pikachu);  
 battle.addFoe(sandygast);  
 battle.go();  
 }  
}

## Pokemons

### Latias

package pokemons;  
  
import moves.physical.Bulldoze;  
import moves.physical.ZenHeadbutt;  
import moves.special.MistBall;  
import moves.status.HealPulse;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Latias extends Pokemon {  
 public Latias(String name, int lvl) {  
 super(name, lvl);  
 setType(Type.*DRAGON*, Type.*PSYCHIC*);  
 setStats(80, 80, 90, 110, 130, 110);  
 addMove(new Bulldoze());  
 addMove(new HealPulse());  
 addMove(new ZenHeadbutt());  
 addMove(new MistBall());  
 }  
}

### Sandyghast

package pokemons;  
  
import moves.special.Absorb;  
import moves.special.MegaDrain;  
import moves.status.DoubleTeam;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
import java.util.Arrays;  
  
public class Sandygast extends Pokemon {  
 public Sandygast(String name, int lvl) {  
 super(name, lvl);  
 setType(Type.*GHOST*, Type.*GROUND*);  
 setStats(55, 55, 80, 70, 45, 15);  
 addMove(new Absorb());  
 addMove(new MegaDrain());  
 addMove(new DoubleTeam());  
 }  
}

### Palossand

package pokemons;  
  
import moves.special.SludgeBomb;  
  
public class Palossand extends Sandygast {  
 public Palossand(String name, int lvl) {  
 super(name, lvl);  
 setStats(85, 75, 110, 100, 75, 35);  
 addMove(new SludgeBomb());  
 }  
}

### Pichu

package pokemons;  
  
import moves.physical.WildCharge;  
import moves.special.Thunder;  
import ru.ifmo.se.pokemon.\*;  
  
public class Pichu extends Pokemon {  
 public Pichu(String name, int lvl) {  
 super(name, lvl);  
 setType(Type.*ELECTRIC*);  
 setStats(20, 40, 15, 35, 35, 60);  
 addMove(new WildCharge());  
 addMove(new Thunder());  
 }  
}

### Pikachu

package pokemons;  
  
import moves.status.Agility;  
  
public class Pikachu extends Pichu {  
 public Pikachu(String name, int lvl) {  
 super(name, lvl);  
 setStats(35, 55, 40, 50, 50, 90);  
 addMove(new Agility());  
 }  
}

### Raichu

package pokemons;  
  
import moves.status.Confide;  
  
public class Raichu extends Pikachu {  
 public Raichu(String name, int lvl) {  
 super(name, lvl);  
 setStats(60, 90, 55, 90, 80, 110);  
 addMove(new Confide());  
 }  
}

## Moves

### Physical

#### Bulldoze

package moves.physical;  
  
import ru.ifmo.se.pokemon.PhysicalMove;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class Bulldoze extends PhysicalMove {  
 public Bulldoze() {  
 super(Type.*GROUND*, 60, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 def.setMod(Stat.*HP*, (int) damage);  
 def.setMod(Stat.*SPEED*, -1);  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Bulldoze\"";  
 }  
}

#### Wild Charge

package moves.physical;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class WildCharge extends PhysicalMove {  
 public WildCharge() {  
 super(Type.*ELECTRIC*, 90, 100);  
 }  
  
 @Override  
 protected void applySelfDamage(Pokemon att, double damage) {  
 att.setMod(Stat.*HP*, (int) (damage \* (1.0 / 4)));  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Wild Charge\"";  
 }  
}

#### Zen Headbutt

package moves.physical;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class ZenHeadbutt extends PhysicalMove {  
 public ZenHeadbutt() {  
 super(Type.*PSYCHIC*, 80, 90);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 def.setMod(Stat.*HP*, (int) damage);  
 if (0.2 >= Math.*random*()) {  
 Effect.*flinch*(def);  
 }  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Zen Headbutt\"";  
 }  
}

### Special

#### Absorb

package moves.special;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class Absorb extends SpecialMove {  
 public Absorb() {  
 super(Type.*GRASS*, 20, 100);  
 }  
  
 @Override  
 protected void applySelfDamage(Pokemon atk, double damage) {  
 atk.setMod(Stat.*HP*, -1 \* (int) (damage \* 0.5));  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Absorb\"";  
 }  
}

#### Mega Drain

package moves.special;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class MegaDrain extends SpecialMove {  
 public MegaDrain() {  
 super(Type.*GRASS*, 40, 100);  
 }  
  
 @Override  
 protected void applySelfDamage(Pokemon att, double damage) {  
 att.setMod(Stat.*HP*, -1 \* (int) (damage \* 0.5));  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Mega Drain\"";  
 }  
}

#### Mist Ball

package moves.special;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.Type;  
  
public class MistBall extends SpecialMove {  
 public MistBall() {  
 super(Type.*PSYCHIC*, 95, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 def.setMod(Stat.*HP*, (int) damage);  
 if (0.5 >= Math.*random*()) {  
 def.setMod(Stat.*SPECIAL\_ATTACK*, -1);  
 }  
 }  
}

#### Sludge Bomb

package moves.special;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class SludgeBomb extends SpecialMove {  
 public SludgeBomb() {  
 super(Type.*POISON*, 90, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 def.setMod(Stat.*HP*, (int) damage);  
 if (0.3 >= Math.*random*()) {  
 Effect.*poison*(def);  
 }  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Sludge Bomb\"";  
 }  
}

#### Thunder

package moves.special;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Thunder extends SpecialMove {  
 public Thunder() {  
 super(Type.*ELECTRIC*, 110, 70);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage) {  
 if (0.3 >= Math.*random*()) {  
 Effect.*paralyze*(def);  
 }  
 def.setMod(Stat.*HP*, (int) damage);  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Thunder\"";  
 }  
}

### Status

#### Agility

package moves.status;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Agility extends StatusMove {  
 public Agility() {  
 super(Type.*PSYCHIC*, 0, 0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*SPEED*, +2);  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Agility\"";  
 }  
}

#### Confide

package moves.status;  
  
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.*ATTACK*, -1);  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Confide\"";  
 }  
}

#### Double Team

package moves.status;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
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 java.lang.String describe() {  
 return "использует \"Double Team\"";  
 }  
}

#### Heal Pulse

package moves.status;  
  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Stat;  
import ru.ifmo.se.pokemon.StatusMove;  
import ru.ifmo.se.pokemon.Type;  
  
public class HealPulse extends StatusMove {  
 public HealPulse() {  
 super(Type.*PSYCHIC*, 0, 0);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*HP*, -1 \* (int) (p.getHP() \* 0.5));  
 }  
  
 @Override  
 protected java.lang.String describe() {  
 return "использует \"Heal Pulse\"";  
 }  
}

# Результат выполнения программы

Файл .log можно найти по ссылке: <https://github.com/ldpst/itmo/blob/main/labs/sem1/prog/lab2/lab2.log>

Latias Латиас из команды зеленых вступает в бой!

Palossand Палоссэнд из команды синих вступает в бой!

Latias Латиас использует "Bulldoze".

Palossand Палоссэнд теряет 4 здоровья.

Palossand Палоссэнд уменьшает скорость.

Palossand Палоссэнд использует "Sludge Bomb".

Latias Латиас теряет 5 здоровья.

Latias Латиас использует "Zen Headbutt".

Palossand Палоссэнд теряет 7 здоровья.

Palossand Палоссэнд промахивается

Latias Латиас использует "Bulldoze".

Palossand Палоссэнд теряет 3 здоровья.

Palossand Палоссэнд уменьшает скорость.

Palossand Палоссэнд теряет сознание.

Pikachu Пикачу из команды синих вступает в бой!

Latias Латиас использует "Bulldoze".

Pikachu Пикачу теряет 6 здоровья.

Pikachu Пикачу уменьшает скорость.

Pikachu Пикачу промахивается

Pikachu Пикачу промахивается

Latias Латиас атакует.

Pikachu Пикачу теряет 10 здоровья.

Pikachu Пикачу теряет сознание.

Sandygast Сэндигаст из команды синих вступает в бой!

Latias Латиас атакует.

Sandygast Сэндигаст теряет 8 здоровья.

Sandygast Сэндигаст уменьшает специальную атаку.

Sandygast Сэндигаст использует "Absorb".

Latias Латиас теряет 2 здоровья.

Sandygast Сэндигаст восстанавливает 1 здоровья.

Latias Латиас промахивается

Sandygast Сэндигаст использует "Absorb".

Latias Латиас теряет 2 здоровья.

Sandygast Сэндигаст восстанавливает 1 здоровья.

Latias Латиас использует "Bulldoze".

Sandygast Сэндигаст теряет 5 здоровья.

Sandygast Сэндигаст уменьшает скорость.

Sandygast Сэндигаст использует "Mega Drain".

Latias Латиас теряет 2 здоровья.

Sandygast Сэндигаст восстанавливает 1 здоровья.

Latias Латиас промахивается

Sandygast Сэндигаст промахивается

Latias Латиас промахивается

Sandygast Сэндигаст борется с соперником.

Latias Латиас теряет 5 здоровья.

Sandygast Сэндигаст теряет 1 здоровья.

Latias Латиас теряет сознание.

Pichu Пичу из команды зеленых вступает в бой!

Pichu Пичу борется с соперником.

Sandygast Сэндигаст теряет 4 здоровья.

Pichu Пичу теряет 1 здоровья.

Sandygast Сэндигаст теряет сознание.

В команде синих не осталось покемонов.

Команда зеленых побеждает в этом бою!

# Вывод

Во время выполнения лабораторной работы я ближе познакомился с объектно-ориентированным программированием на языке программирования Java. Я узнал, как правильно наследовать классы, получать и использовать собственными типы данных. По итогу лабораторной работы я написал свою программу симулятор боя покемонов, использующую информацию с сайтов по покемонам.