Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

**Лабораторная работа #2**

**«Объектно-ориентированное программирование на Java»**

**по курсу «Программирование»**

Вариант 311434

Выполнил:

Студент 1-го курса, группы P3114

Ермаков Т. С., 311733

Проверил:

Каюков Иван Алексеевич

Санкт-Петербург, 2020

**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);

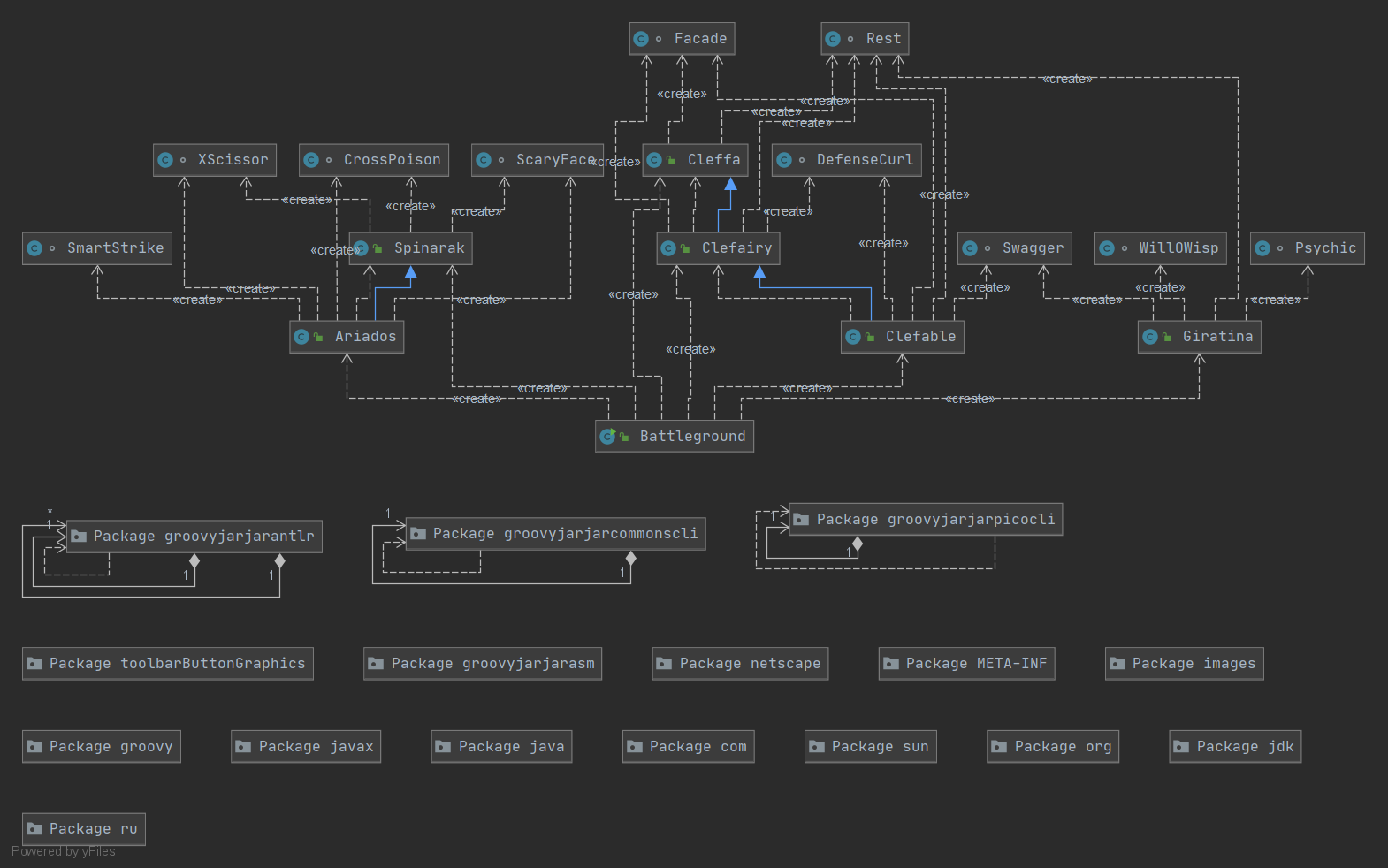
b.go();

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.

**Покемоны:**

****

**Выполнение работы**

**Диаграмма классов**

**Исходный код программы.**

**Battleground.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Battleground {  
  
 public static void main(String[] args) {  
 Battle field = new Battle();  
 field.addAlly(new Spinarak("Petrov", 2));  
 field.addAlly(new Giratina("Ivanova", 1));  
 field.addAlly(new Ariados("Sidorov", 2));  
 field.addFoe(new Cleffa("Smirnova", 3));  
 field.addFoe(new Clefairy("Popov", 3));  
 field.addFoe(new Clefable("Sokolova", 2));  
 field.go();  
 }  
}

**PokeAttacks.java**

import ru.ifmo.se.pokemon.\*;  
  
class WillOWisp extends PhysicalMove {  
 protected WillOWisp() {  
 super(Type.*FIRE*, 0, 75);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 Effect.*burn*(p);  
 }  
  
 @Override  
 protected String describe(){  
 return "Атака Огненного типа, которая поджигает противника";  
 }  
}  
  
class Psychic extends StatusMove {  
 protected Psychic() {  
 super(Type.*PSYCHIC*, 90, 100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.1) p.setMod(Stat.*SPECIAL\_DEFENSE*, -1);  
 }  
  
 @Override  
 protected String describe(){  
 return "Имеет 10% шанс понизить Спец. Защиту цели на одну ступень";  
 }  
}  
  
class Rest extends StatusMove {  
 protected Rest() {  
 super(Type.*PSYCHIC*, 0, 0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*HP*, 100);  
 Effect.*sleep*(p);  
 Effect effect = new Effect();  
 effect.turns(2);  
 p.addEffect(effect);  
 }  
  
 @Override  
 protected String describe(){  
 return "Покемон полностью вылечивает себя и засыпает на два хода";  
 }  
}  
  
class CrossPoison extends PhysicalMove {  
 protected CrossPoison() {  
 super(Type.*POISON*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 if (Math.*random*() <= 0.1) def.setMod(Stat.*HP*, 0);  
 }  
  
 @Override  
 protected double calcCriticalHit(Pokemon pokemon, Pokemon pokemon1) {  
 pokemon.setMod(Stat.*SPEED*, 1);  
 if (Math.*random*() < pokemon.getStat(Stat.*SPEED*) / 512) return 2 \* pokemon.getStat(Stat.*ATTACK*);  
 else return pokemon.getStat(Stat.*ATTACK*);  
 }  
  
 @Override  
 protected String describe(){  
 return "Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара";  
 }  
}  
  
  
class XScissor extends PhysicalMove {  
 protected XScissor() { super(Type.*BUG*, 80, 100); }  
  
 @Override  
 protected String describe(){  
 return "Наносит обычные повреждения без дополнительных эффектов";  
 }  
}  
  
class ScaryFace extends PhysicalMove {  
 protected ScaryFace() { super(Type.*NORMAL*, 0, 100); }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*SPEED*, -2);  
 }  
  
 @Override  
 protected String describe(){  
 return "Понижает Скорость цели на две ступени";  
 }  
}  
  
class SmartStrike extends PhysicalMove {  
 protected SmartStrike() { super(Type.*STEEL*, 70, 0); }  
  
 @Override  
 protected String describe(){  
 return "Покемон наносит цели удар в спину острым рогом. Никогда не промахивается";  
 }  
}  
  
class DefenseCurl extends PhysicalMove {  
 protected DefenseCurl() { super(Type.*NORMAL*, 0, 0); }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.setMod(Stat.*DEFENSE*, 1);  
 }  
  
 @Override  
 protected String describe(){  
 return "Покемон сворачивается, пряча уязвимые места и повышая свою Защиту";  
 }  
}  
  
class Swagger extends StatusMove{  
 protected 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 "Повышает Атаку цели на две ступени и сбивает её с толку";  
 }  
}  
  
class Facade extends PhysicalMove{  
 protected Facade(){  
 super(Type.*NORMAL*, 70, 100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage){  
 Status PokCon = def.getCondition();  
 if (PokCon.equals(Status.*BURN*) || PokCon.equals(Status.*POISON*) || PokCon.equals(Status.*PARALYZE*)) {  
 def.setMod(Stat.*HP*, (int) Math.*round*(damage) \* 2);  
 }  
 }  
  
 @Override  
 protected String describe(){  
 return "Сила удваивается, если использующий обожжён, парализован или отравлен";  
 }  
}

**Giratina.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Giratina extends Pokemon {  
 public Giratina(String name, int level) {  
 super(name, level);  
 setStats(150, 100, 120, 100, 120,90);  
 setType(Type.*GHOST*, Type.*DRAGON*);  
 setMove(new WillOWisp(), new Psychic(), new Rest(), new Swagger());  
 }  
}

**Spinarak.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Spinarak extends Pokemon {  
 public Spinarak(String name, int level) {  
 super(name, level);  
 setStats(40,60,40,40,40,30);  
 setType(Type.BUG, Type.POISON);  
 setMove(new CrossPoison(), new XScissor(), new ScaryFace());  
 }  
}

**Ariados.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Ariados extends Spinarak {  
 public Ariados(String name, int level) {  
 super(name, level);  
 setStats(70, 90, 70, 60, 70, 40);  
 setType(Type.*BUG*, Type.*POISON*);  
 setMove(new CrossPoison(), new XScissor(), new ScaryFace(), new SmartStrike());  
 }  
}

**Cleffa.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Cleffa extends Pokemon {  
 public Cleffa(String name, int level) {  
 super(name, level);  
 setStats(50, 25,28, 45, 55, 15);  
 setType(Type.*FAIRY*);  
 setMove(new Rest(), new Facade());  
 }  
}

**Clefairy.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Clefairy extends Cleffa {  
 public Clefairy(String name, int level) {  
 super(name, level);  
 setStats(70, 45, 48, 60, 65,35);  
 setType(Type.*FAIRY*);  
 setMove(new Rest(), new Facade(), new DefenseCurl());  
 }  
}

**Clefable.java**

import ru.ifmo.se.pokemon.\*;  
  
public class Clefable extends Clefairy {  
 public Clefable(String name, int level) {  
 super(name, level);  
 setStats(95,70,73,95,90,60);  
 setType(Type.*GRASS*);  
 setMove(new Rest(), new Facade(), new DefenseCurl(), new Swagger());  
 }  
}

**Результат работы программы**

Spinarak Petrov из команды белых вступает в бой!

Cleffa Smirnova из команды зеленых вступает в бой!

Spinarak Petrov Понижает Скорость цели на две ступени.

Cleffa Smirnova теряет 4 здоровья.

Cleffa Smirnova уменьшает скорость.

Cleffa Smirnova Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Cleffa Smirnova Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Cleffa Smirnova промахивается

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Cleffa Smirnova теряет 5 здоровья.

Cleffa Smirnova промахивается

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Cleffa Smirnova Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Cleffa Smirnova теряет 4 здоровья.

Cleffa Smirnova Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Cleffa Smirnova Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Cleffa Smirnova теряет 5 здоровья.

Cleffa Smirnova теряет сознание.

Clefairy Popov из команды зеленых вступает в бой!

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Clefairy Popov теряет 3 здоровья.

Clefairy Popov промахивается

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Clefairy Popov теряет 4 здоровья.

Clefairy Popov промахивается

Spinarak Petrov Понижает Скорость цели на две ступени.

Clefairy Popov теряет 4 здоровья.

Clefairy Popov уменьшает скорость.

Clefairy Popov промахивается

Clefairy Popov Сила удваивается, если использующий обожжён, парализован или отравлен.

Критический удар!

Spinarak Petrov Понижает Скорость цели на две ступени.

Clefairy Popov теряет 4 здоровья.

Clefairy Popov уменьшает скорость.

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Clefairy Popov промахивается

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Clefairy Popov теряет 5 здоровья.

Clefairy Popov теряет сознание.

Clefable Sokolova из команды зеленых вступает в бой!

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Clefable Sokolova промахивается

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Clefable Sokolova теряет 16 здоровья.

Clefable Sokolova промахивается

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Clefable Sokolova Повышает Атаку цели на две ступени и сбивает её с толку.

Spinarak Petrov увеличивает атаку.

Spinarak Petrov Режущая атака ядовитым лезвием, также способная отравить цель. Имеет высокий шанс критического удара.

Spinarak Petrov увеличивает скорость.

Clefable Sokolova промахивается

Spinarak Petrov Наносит обычные повреждения без дополнительных эффектов.

Clefable Sokolova теряет 21 здоровья.

Clefable Sokolova теряет сознание.

В команде зеленых не осталось покемонов.

Команда белых побеждает в этом бою!

**Выводы по работе**

В результате выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования.

Я научился применять ООП на практике, разработав классы и методы покемонов по существующему API, создавать диаграммы классов для визуализации взаимосвязей в коде.