МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Университет ИТМО

**Факультет программной инженерии и компьютерной техники**

Лабораторная работа №2

по дисциплине:

«***Программирование***»

Вариант №33103

Выполнил:

Студент группы № P3133

Аганин Егор Владимирович

Преподаватель:

Шешуков Дмитрий Михайлович



Санкт-Петербург, 2021

Оглавление

[Задание: 3](#_Toc85814537)

[Покемоны и их атаки: 4](#_Toc85814538)

[UML диаграмма 5](#_Toc85814539)

[Код программы 6](#_Toc85814540)

[Class main 6](#_Toc85814541)

[Покемоны 7](#_Toc85814542)

[Passimian 7](#_Toc85814543)

[Wingull 7](#_Toc85814544)

[Pelipper 8](#_Toc85814545)

[Ralts 8](#_Toc85814546)

[Kirlia 9](#_Toc85814547)

[**Gardevoir** 9](#_Toc85814548)

[Атаки 10](#_Toc85814549)

[AirSlash 10](#_Toc85814550)

[BraveBird 10](#_Toc85814551)

[Confide 11](#_Toc85814552)

[DoubleTeam 11](#_Toc85814553)

[EnergyBall 12](#_Toc85814554)

[NightSlash 12](#_Toc85814555)

[PoisonPowder 13](#_Toc85814556)

[Rest 13](#_Toc85814557)

[RockTomb 14](#_Toc85814558)

[Swagger 14](#_Toc85814559)

[SwordsDance 15](#_Toc85814560)

[WoodHammer 15](#_Toc85814561)

[Вывод 16](#_Toc85814562)

# 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 и их атаки:

#### 

#### 

**Отчёт по работе должен содержать:**

1. Текст задания.
2. Диаграмма классов реализованной объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

**Вопросы к защите лабораторной работы:**

1. Объектно-ориентированное программирование. Основные понятия: объекты, наследование, полиморфизм, инкапсуляция.
2. Понятие класса. Классы и объекты в Java.
3. Члены класса. Модификаторы доступа.
4. Создание и инициализация объектов. Вызов методов.
5. Области видимости переменных.
6. Модификаторы final и static.
7. Пакеты, инструкция import.

# UML диаграмма

Изображение выглядит как текст, внутренний, черный

Автоматически созданное описание

# Код программы

## Class main

Изображение выглядит как текст

Автоматически созданное описание

## Покемоны

### Passimian

Изображение выглядит как текст

Автоматически созданное описание

### Wingull

Изображение выглядит как текст

Автоматически созданное описание

### Pelipper

Изображение выглядит как текст

Автоматически созданное описание

### Ralts

Изображение выглядит как текст

Автоматически созданное описание

### Kirlia

Изображение выглядит как текст

Автоматически созданное описание

### **Gardevoir**

Изображение выглядит как текст

Автоматически созданное описание

## Атаки

### AirSlash

Изображение выглядит как текст

Автоматически созданное описание

### BraveBird

Изображение выглядит как текст

Автоматически созданное описание

### Confide

Изображение выглядит как текст

Автоматически созданное описание

### DoubleTeam

Изображение выглядит как текст

Автоматически созданное описание

### EnergyBall

Изображение выглядит как текст

Автоматически созданное описание

### NightSlash

Изображение выглядит как текст

Автоматически созданное описание

### PoisonPowder

Изображение выглядит как текст

Автоматически созданное описание

### Rest

Изображение выглядит как текст

Автоматически созданное описание

### RockTomb

Изображение выглядит как текст

Автоматически созданное описание

### Swagger

Изображение выглядит как текст

Автоматически созданное описание

### SwordsDance

Изображение выглядит как текст

Автоматически созданное описание

### WoodHammer

Изображение выглядит как текст

Автоматически созданное описание

# Вывод

Выполняя данную лабораторную работу я научился основам Объектно Ориентированного Программирования Java.

Я узнал что такое класс, метод, поле, геттеры, сеттеры, конструкторы, узнал для чего они нужны и как их применять.Я понял на конкретном примере как работают модификаторы доступа private, public, protected.