НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХТЕХНОЛОГИЙ

**ЛАБОРАТОРНАЯ РАБОТА №2**

Дисциплина «ПРОГРАММИРОВАНИЕ»

Лабораторная работа № 2

Выполнил студент

Здор Матвей Максимович

Группа № P3124

Преподаватель: Пименов Данила Дмитриевич

Санкт-Петербург

2023

Оглавление

[Задание: 3](#_Toc148602245)

[Описание работы: 4](#_Toc148602246)

[Отчет: 5](#_Toc148602247)

[Код программы 5](#_Toc148602248)

[UML: 5](#_Toc148602249)

[Вывод: 5](#_Toc148602250)

# 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 StatusMoveреализовать 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://pokemondb.net>,<http://veekun.com/dex/pokemon>



# Описание работы:

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

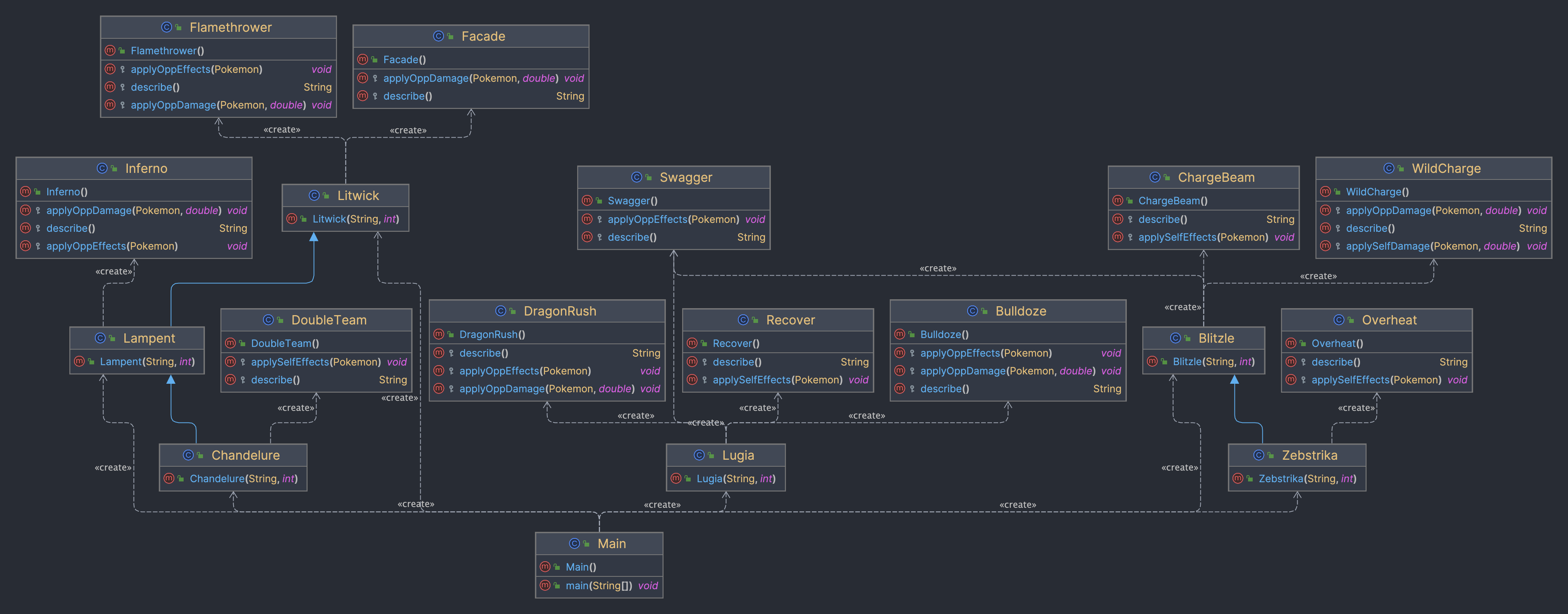
Что надо сделать (краткое описание)

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.

# Отчет:

## [Код программы](https://github.com/matviikk/itmo/tree/1b0ae4e9c91ef09558afb6a1f7ca1af9f3ee414e/term_1/programming/lab2)

## UML:



# Вывод:

В процессе выполнения данной лабораторной работы я ознакомился с основными концепциями объектно-ориентированного программирования на практическом примере симуляции боев между покемонами. Я научился создавать базовые классы, реализовывать наследование и переопределение методов. Создание различных классов покемонов и их атак дало понимание полиморфизма и показало, как можно использовать один и тот же метод в разных классах с разными реализациями.