

Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

Лабораторная работа по программированию №2

Вариант №2666

Группа: Р3107

Студент: Чусовлянов Максим Сергеевич

Преподаватель: Письмак Алексей Евгеньевич

г. Санкт-Петербург

2023

Оглавление

Текст задания	3
Диаграмма классов	4
Исходный код программы	5
Результат работы программы	6
Вывод	7

Текст задания

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.

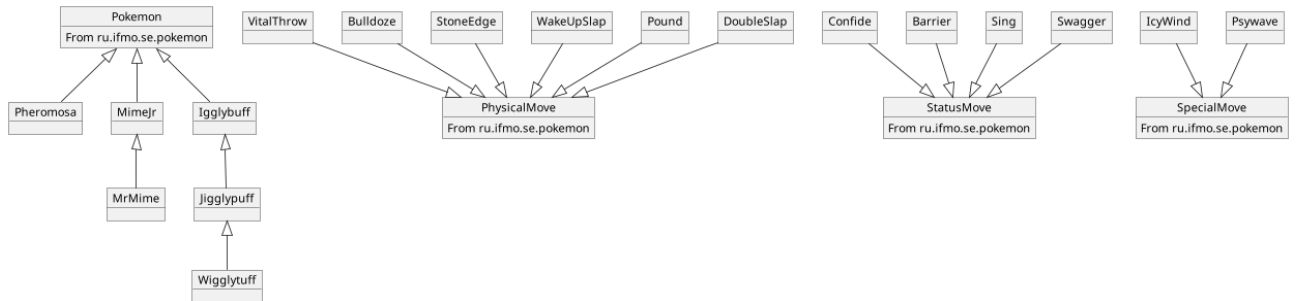
Базовые классы и симулятор сражения находятся в jar-архиве. Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах.

Pheromosa  Атаки: <ul style="list-style-type: none">✓ Vital Throw✓ Confide✓ Bulldoze✓ Stone Edge	Mime Jr.  Атаки: <ul style="list-style-type: none">✓ Icy Wind✓ Wake-Up Slap✓ Barrier	Mr. Mime  Атаки: <ul style="list-style-type: none">✓ Icy Wind✓ Wake-Up Slap✓ Barrier✓ Psywave	Igglybuff  Атаки: <ul style="list-style-type: none">✓ Sing✓ Swagger	Jigglypuff  Атаки: <ul style="list-style-type: none">✓ Sing✓ Swagger✓ Pound	Wigglytuff  Атаки: <ul style="list-style-type: none">✓ Sing✓ Swagger✓ Pound✓ Double Slap
--	---	---	--	--	--

Диаграмма классов

Вывод в UML-формате см. в репозитори:

<https://github.com/Vaneshik/VT-Labs/blob/main/java/lab2/UML/uml.plantuml>



Исходный код программы

Репозиторий: <https://github.com/Vaneshik/VT-Labs/tree/main/java/lab2>

Результат работы программы

См. в репозитории:

<https://raw.githubusercontent.com/Vaneshik/VT-Labs/main/java/lab2/battle.log>

Вывод

Во время выполнения данной лабораторной работы, я изучил синтаксис PlantUML для создания UML-диаграмм, узнал об основах ООП и научился применять их на практике, научился подключать внешнюю jar-зависимость в проект и собирать такой проект при помощи средств JDK.