



Вариант №15256898
Лабораторная работа №2
По дисциплине
Программирование

Выполнил студент группы Р3115:
Барсуков Максим

Преподаватель:
Сорокин Роман Борисович
Письмак Алексей Евгеньевич

1. 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.



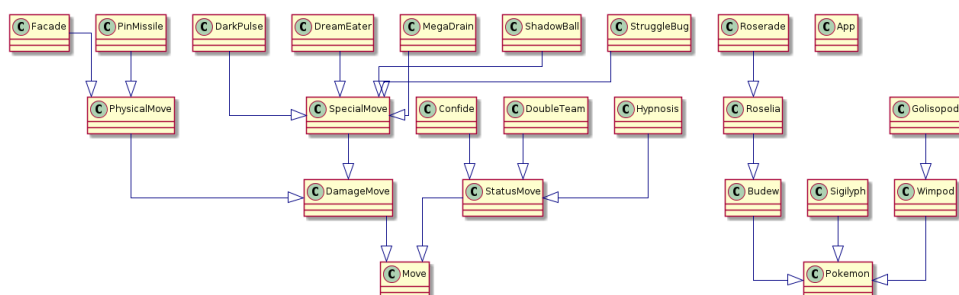
2. Исходный код программы.

Репозиторий:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab2>

3. Диаграмма классов реализованной объектной модели.

Вывод в UML-формате см. в репозитории.



4. Результат работы программы:

См. в репозитории:

<https://raw.githubusercontent.com/maxbarsukov/itmo/master/%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab2/logs/result.log>

5. Вывод

Во время выполнения данной лабораторной работы я познакомился с языком программирования Kotlin, системой сборки Gradle, инструментом для генерации UML-диаграмм PlantUML, узнал об основах ООП и научился применять их на практике, научился подключать внешнюю jar-зависимость в проект и собирать его в fat jar с помощью Gradle.

