Санкт-Петербургский Национальный Исследовательский Университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники

Вариант №478090
Лабораторная работа №2
По дисциплине
Программирование

Выполнил студент группы P3110:
Ахроров Карроматуллохон Фирдавсович

Преподаватель:
Гаврилов Антон Владимирович
Мустафаева Айнур Вугар Кызы

Санкт-Петербург 2024 г.

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. Все разработанные классы, не имеющие наследников, должны быть реализованы таким образом, чтобы от них нельзя было наследоваться.

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 (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - здесь.

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах http://poke-universe.ru, http://pokemondb.net, http://veekun.com/dex/pokemon

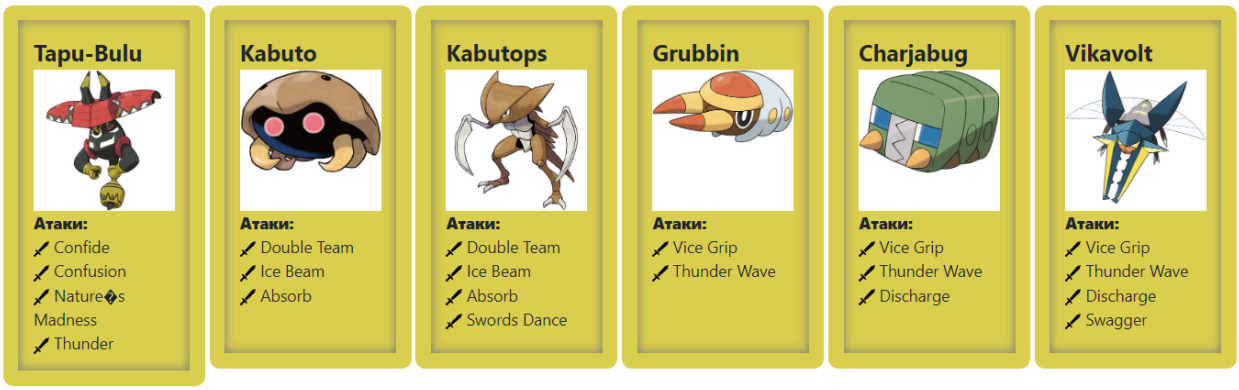

Рисунок 1. Покемоны

2. Исходный код программы.

Репозиторий:
https://github.com/Ahrorovk/itmo_programming_2

3. Диаграмма классов реализованной объектной модели.

Вывод в UML-формате см. в репозитории.

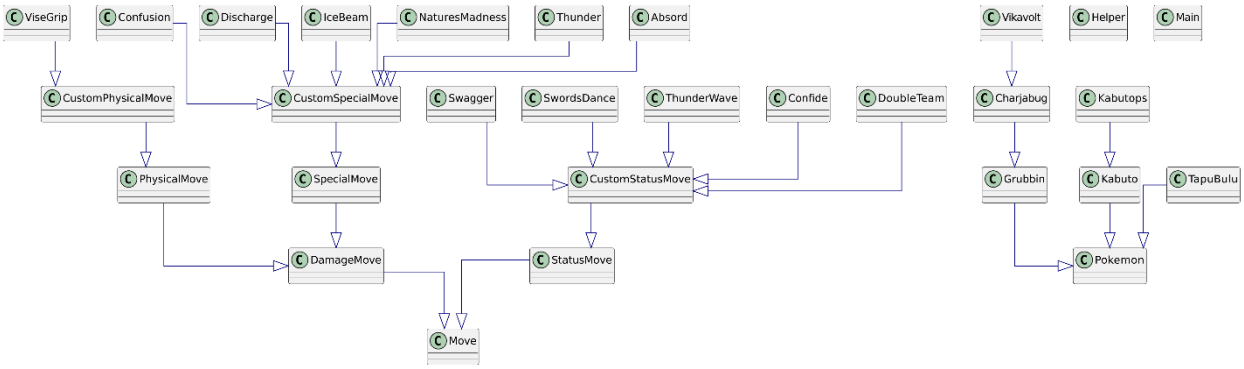

Рисунок 2. Диаграмма

4. Результат работы программы:

См. в репозитории:
https://github.com/Ahrorovk/itmo_programming_2/blob/main/logs/results.log

5. Вывод

Во время выполнения данной лабораторной работы я улучшил и укрепил свои знания языка программирования Java, использовал систему сборки Gradle Kts, научился пользоваться инструментом для генерации UML-диаграмм PlantUML, закрепил свои знания об основах ООП и повторил их на практике(использовал более удобную конструкцию, которая подходит в качестве использования ООП, решив задачу с изменением функции describe оптимальным способом), научился подключать внешнюю jar-зависимость в проект и собирать его в fat jar с помощью Gradle Kts.