Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант № 27869

Лабораторная работа №2

По дисциплине

Программирование

Выполнил студент группы P3109:

Поляков Алексей

Преподаватель:

Гаврилов А. В.

Мустафаева А. В.

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.

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.

Изображение выглядит как текст, мультфильм, млекопитающее

Автоматически созданное описание

*Рисунок 1 – Мои Покемоны*

1. **Исходный код программы.**

Исходный код программы можно посмотреть в репозитории на [GitHub](https://github.com/programeraleksey/ITMO/tree/8626cd7ecd77acbe4c72eb9a96028422cc199390/programming/labs/lab2/code).

1. **Диаграмма классов реализованной объектной модели.**

Диаграмма классов реализованной объектной модели представлена на рисунке 2

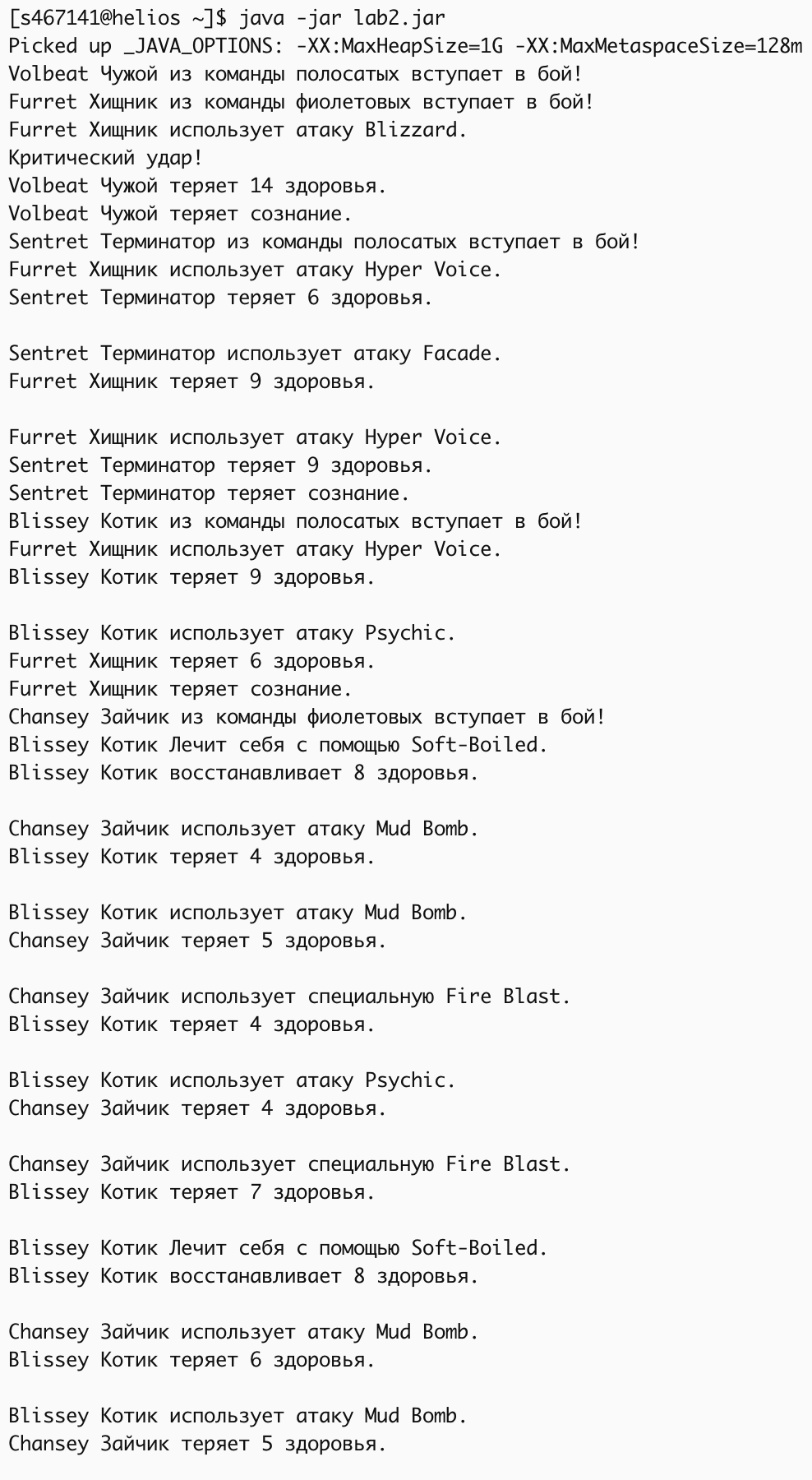
Изображение выглядит как линия, диаграмма, План, Параллельный

Автоматически созданное описание

*Рисунок 2 - Диаграмма классов.*

1. **Результат работы программы:**

Результат работы программы представлен на рисунках 3.1 и 3.2:



*Рисунок 3.1 – Вывод программы*

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

*Рисунок 3.2 – Вывод программы*

1. **Вывод**

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