

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО »

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине
«ПРОГРАММИРОВАНИЕ»

Вариант № 311901

Выполнил:

Студент группы Р3119

Таранов Кирилл

Викторович

Преподаватель:

Пашнин Александр

Денисович

Содержание

Задание.....	3
Исходный код.....	4
UML-диаграмма классов	4
Вывод:.....	4

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 `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.

Введите вариант:

Ваши покемоны:

Zygarde  Атаки: ✓ Bulldoze ✓ Haze ✓ Glare ✓ Crunch	Koffing  Атаки: ✓ Sludge ✓ Tackle ✓ Smog	Weezing  Атаки: ✓ Sludge ✓ Tackle ✓ Smog ✓ Double Hit	Cleffa  Атаки: ✓ Work Up ✓ Swagger	Clefairy  Атаки: ✓ Work Up ✓ Swagger ✓ Moonblast	Clefable  Атаки: ✓ Work Up ✓ Swagger ✓ Moonblast ✓ Sing
---	--	--	---	--	--

Исходный код

<https://github.com/k1rtar/LabWorksOnProgramming/tree/main/Lab2>

UML-диаграмма классов

<https://github.com/k1rtar/LabWorksOnProgramming/blob/main/Lab2/UML%20class%20diagram%20Lab2.png>

Вывод:

На простом примере разобрался с основными концепциями ООП и научился использовать их в программах.