Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: «Программирование»

**Лабораторная работа по программированию №2**

­

Выполнил: Третьяков Илья Антонович

Группа: P3108

Вариант: 827

Преподаватель: Харитонова Анастасия Евгеньевна

Санкт-Петербург

2021

# 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 StatusMoveреализовать 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](https://helios.cs.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](http://se.ifmo.ru/~tony/doc/).

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

#### **Комментарии**

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с [документацией](https://se.ifmo.ru/~tony/doc/), обращая особое внимание на классы Pokemon и Move. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл Pokemon.jar. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть, как она работает.
4. Battle b = new Battle();
5. Pokemon p1 = new Pokemon("Чужой", 1);
6. Pokemon p2 = new Pokemon("Хищник", 1);
7. b.addAlly(p1);
8. b.addFoe(p2);
9. b.go();
10. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса Pokemon. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
11. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса PhysicalMove или SpecialMove. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод describe, чтобы выводилось нужное сообщение.
12. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники StatusMove), скорее всего придется разобраться с классом Effect. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
13. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Graphical user interface

Description automatically generated with medium confidence

# Диаграмма классов реализованной объектной модели

Graphical user interface, diagram

Description automatically generated

# Исходный код программы

Ссылка на исходный код на GitHub:

<https://github.com/bitte165/itmo/tree/main/Programming/Lab2/ru/bitte>

# Результат работы программы

**Mew 1 из команды фиолетовых вступает в бой!**

**Pichu 4 из команды желтых вступает в бой!**

**Mew 1 использует Low Sweep.**

**Pichu 4 теряет 7 здоровья.**

**Pichu 4 уменьшает скорость.**

**Pichu 4 использует Swagger - сбивает противника с толку.**

**Mew 1 увеличивает атаку.**

**Pichu 4 использует Swagger - сбивает противника с толку.**

**Mew 1 увеличивает атаку.**

**Mew 1 растерянно попадает по себе.**

**Mew 1 теряет 5 здоровья.**

**Pichu 4 использует Thunder.**

**Mew 1 теряет 8 здоровья.**

**Mew 1 использует Waterfall.**

**Pichu 4 теряет 5 здоровья.**

**Pichu 4 теряет сознание.**

**Pikachu 5 из команды желтых вступает в бой!**

**Mew 1 использует Low Sweep.**

**Pikachu 5 теряет 6 здоровья.**

**Pikachu 5 уменьшает скорость.**

**Pikachu 5 использует Thunder.**

**Mew 1 теряет 8 здоровья.**

**Mew 1 парализован**

**Mew 1 теряет сознание.**

**Ponyta 2 из команды фиолетовых вступает в бой!**

**Pikachu 5 промахивается**

**Ponyta 2 использует Growl.**

**Pikachu 5 уменьшает атаку.**

**Pikachu 5 использует Thunder.**

**Ponyta 2 теряет 9 здоровья.**

**Ponyta 2 парализован**

**Ponyta 2 использует Flamethrower.**

**Критический удар!**

**Pikachu 5 теряет 20 здоровья.**

**Pikachu 5 теряет сознание.**

**Raichu 6 из команды желтых вступает в бой!**

**Ponyta 2 использует Growl.**

**Raichu 6 уменьшает атаку.**

**Raichu 6 использует Swagger - сбивает противника с толку.**

**Ponyta 2 увеличивает атаку.**

**Raichu 6 использует Swagger - сбивает противника с толку.**

**Ponyta 2 увеличивает атаку.**

**Ponyta 2 использует Flamethrower.**

**Критический удар!**

**Raichu 6 теряет 14 здоровья.**

**Raichu 6 теряет сознание.**

**В команде желтых не осталось покемонов.**

**Команда фиолетовых побеждает в этом бою!**

# Вывод

Проделав лабораторную работу, я научился основам объектно-ориентированного программирования, а именно: научился создавать объекты из предопределённых классов и вызывать их методы, научился создавать свои классы, добавлять им методы и создавать объекты из них. Также я понял основные принципы ООП – инкапсуляцию, наследование и полиморфизм. Помимо этого, я научился переопределять методы в наследуемых классах и понял, как работать с пакетами в Java. Я научился подключать внешние JAR-файлы, импортировать чужие пакеты и создавать свои.