Министерство образования и науки РФ

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

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

по дисциплине ‘ПРОГРАММИРОВАНИЕ’

**Вариант 215621**

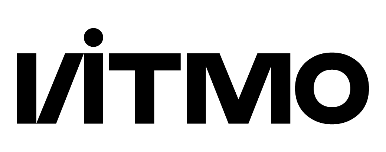
Выполнил:

Студент группы P3130

Суф Шараф Ахдаф

Проверил:

Райла Мартин



Санкт-Петербург  
2022

Оглавление:

[Задание 02](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917643)

[Диаграмма 04](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917644)

[Исходный код 04](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917645)

[Program.java 04](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917646)

[Покемоны 05](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917647)

[Moves 08](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917648)

[Результат работы программы 14](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917649)

[Вывод 15](file:///G:\1.Studies\Labs\Прога\2%20лаба\Лаб%202.docx#_Toc97917650)

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. Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.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](https://se.ifmo.ru/~tony/doc/). Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](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.

Battle b = new Battle();

Pokemon p1 = new Pokemon("Чужой", 1);

Pokemon p2 = new Pokemon("Хищник", 1);

b.addAlly(p1);

b.addFoe(p2);

b.go();

1. 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.
2. 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.
3. 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.
4. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Graphical user interface, application

Description automatically generatedДиаграмма

Diagram

Description automatically generated Исходный код

Program.java

![Background pattern

Description automatically generated with low confidence](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAASsAAAA3CAYAAABUx2zgAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAADSSURBVHhe7dSxDcAwDMAwN///3HbIERZALvpAz/sbgOXOLcBqZgUkmBWQYFZAglkBCWYFJJgVkGBWQIJZAQlmBSSYFZBgVkCCWQEJZgUkmBWQYFZAglkBCWYFJJgVkGBWQIJZAQlmBSSYFZBgVkCCWQEJZgUkmBWQYFZAglkBCWYFJJgVkGBWQIJZAQlmBSSYFZBgVkCCWQEJZgUkmBWQYFZAglkBCWYFJJgVkGBWQIJZAQlmBSSYFZBgVkCCWQEJZgUkmBWQYFZAglkBCWYFBMx8Hh0Eahno0fEAAAAASUVORK5CYII=)Text

Description automatically generated

## Покемоны

Genesect

Text

Description automatically generated

Octillery

Text

Description automatically generated

Remoraid

Text, letter

Description automatically generated

Porygon-Z

Graphical user interface, text, application

Description automatically generated

Porygon2

Text

Description automatically generated

Porygon

Text

Description automatically generated

Moves

Thunderbolt

Graphical user interface, text, application

Description automatically generated

Flash Cannon

Text

Description automatically generated

Dark Pulse

Graphical user interface, text, application

Description automatically generated

Zap Cannon

Graphical user interface, text, application

Description automatically generated

Thunder Wave

Graphical user interface, text, application

Description automatically generated

Charge Beam

Graphical user interface, text, application

Description automatically generated

Double Team

Graphical user interface, text, application

Description automatically generated

Sludge Wave

Graphical user interface, application

Description automatically generated

Facade

Graphical user interface, text, application, email

Description automatically generated

Ice Beam

Text

Description automatically generated

Defense Curl

Graphical user interface, text, application

Description automatically generated

Psybeam

Graphical user interface, text, application

Description automatically generated

# Результат работы программы в Гелиосе

|  |
| --- |
| Genesect Dark from the team Blue enters the battle!  Porygon Stone from the team Black enters the battle!  Genesect Dark does ZapCannon.  Porygon Stone loses 7 hit points.  Porygon Stone does IceBeam.  Genesect Dark loses 3 hit points.  Genesect Dark does Thunderbolt.  Porygon Stone loses 7 hit points.  Porygon Stone hits.  Genesect Dark does FlashCannon.  Porygon Stone loses 11 hit points.  Porygon Stone faints.  Porygon2 Rock from the team Black enters the battle!  Genesect Dark does FlashCannon.  Porygon2 Rock loses 12 hit points.  Porygon2 Rock does IceBeam.  Genesect Dark loses 4 hit points.  Genesect Dark does ZapCannon.  Porygon2 Rock loses 6 hit points.  Porygon2 Rock misses  Genesect Dark does FlashCannon.  Porygon2 Rock loses 7 hit points.  Porygon2 Rock faints.  PorygonZ Diamond from the team Black enters the battle!  Genesect Dark does Thunderbolt.  PorygonZ Diamond loses 8 hit points.  PorygonZ Diamond does IceBeam.  Genesect Dark loses 4 hit points.  Genesect Dark does DarkPulse.  PorygonZ Diamond loses 7 hit points.  PorygonZ Diamond misses  Genesect Dark does Thunderbolt.  PorygonZ Diamond loses 6 hit points.  PorygonZ Diamond does Psybeam.  Genesect Dark loses 3 hit points.  Genesect Dark does Thunderbolt.  PorygonZ Diamond loses 6 hit points.  PorygonZ Diamond faints.  Team Black loses its last Pokemon.  The team Blue wins the battle! |

Вывод

В этой лабораторной работе я узнал, как подключать внешние файлы jar к моей программе. я также научился справляться с проблемами объектно-ориентированного программирования.