

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»

Лабораторная работа №2
по дисциплине «Программирование»

Студент:

Дядев Владислав Александрович

Преподаватель:

Наумова Надежда Александровна

Санкт-Петербург

2024

Задание

Вариант №313164

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](#) (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>

Комментарии






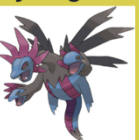
Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с [документацией](#), 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();
```
4. 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.
5. 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.
6. 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.
7. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Ваши покемоны:

<div>Dedenne</div>  <div>Атаки:<ul style="list-style-type: none">✓ Eerie Impulse✓ Rest✓ Charge Beam✓ Play Rough</div>	<div>Onix</div>  <div>Атаки:<ul style="list-style-type: none">✓ Brutal Swing✓ Slam✓ Rest</div>	<div>Steelix</div>  <div>Атаки:<ul style="list-style-type: none">✓ Brutal Swing✓ Slam✓ Rest✓ Ice Fang</div>	<div>Deino</div>  <div>Атаки:<ul style="list-style-type: none">✓ Dark Pulse✓ Facade</div>	<div>Zweilous</div>  <div>Атаки:<ul style="list-style-type: none">✓ Dark Pulse✓ Facade✓ Double Hit</div>	<div>Hydreigon</div>  <div>Атаки:<ul style="list-style-type: none">✓ Dark Pulse✓ Facade✓ Double Hit✓ Swagger</div>
---	---	---	--	---	--

Исходный код программы

Ссылка на github - <https://github.com/Alvas07/ITMO/tree/main/1-2%20Programming/Lab2/src>

Результат работы программы

Результат работы программы представлен на рисунках ниже.

Dedenne НедоПикачу из команды полосатых вступает в бой!
Onix Глыба из команды зеленых вступает в бой!
Dedenne НедоПикачу использует Rest.
Dedenne НедоПикачу засыпает

Onix Глыба использует Slam.
Dedenne НедоПикачу теряет 5 здоровья.

Onix Глыба использует Rest.
Onix Глыба засыпает

Dedenne НедоПикачу использует Play Rough.
Onix Глыба теряет 6 здоровья.

Dedenne НедоПикачу использует Eerie Impulse.
Onix Глыба не замечает воздействие типа ELECTRIC

Dedenne НедоПикачу использует Rest.
Dedenne НедоПикачу засыпает

Onix Глыба использует Slam.
Dedenne НедоПикачу теряет 6 здоровья.

Onix Глыба использует Rest.
Onix Глыба засыпает

Dedenne НедоПикачу использует Eerie Impulse.
Onix Глыба не замечает воздействие типа ELECTRIC

Onix Глыба использует Rest.
Onix Глыба засыпает

Dedenne НедоПикачу использует Eerie Impulse.
Onix Глыба не замечает воздействие типа ELECTRIC

Dedenne НедоПикачу использует Rest.
Dedenne НедоПикачу засыпает

Onix Глыба использует Brutal Swing.
Dedenne НедоПикачу теряет 2 здоровья.
Dedenne НедоПикачу теряет сознание.
Steelix Глыбище из команды полосатых вступает в бой!
Steelix Глыбище использует Brutal Swing.
Onix Глыба теряет 3 здоровья.

Onix Глыба использует Rest.

Onix Глыба засыпает

Steelix Глыбище использует Rest.

Steelix Глыбище засыпает

Onix Глыба использует Rest.

Onix Глыба засыпает

Steelix Глыбище использует Brutal Swing.

Onix Глыба теряет 6 здоровья.

Onix Глыба теряет сознание.

Deino МиниДино из команды зеленых вступает в бой!

Deino МиниДино использует Facade.

Steelix Глыбище теряет 3 здоровья.

Steelix Глыбище использует Ice Fang.

Deino МиниДино теряет 7 здоровья.

Deino МиниДино использует Dark Pulse.

Steelix Глыбище теряет 8 здоровья.

Steelix Глыбище использует Brutal Swing.

Deino МиниДино теряет 3 здоровья.

Deino МиниДино использует Facade.

Steelix Глыбище теряет 2 здоровья.

Steelix Глыбище теряет сознание.

Zweilous НедоГидра из команды полосатых вступает в бой!

Zweilous НедоГидра использует Double Hit.

Deino МиниДино теряет 4 здоровья.

Deino МиниДино теряет 4 здоровья.

Deino МиниДино теряет сознание.

Hydreigon Горыныч из команды зеленых вступает в бой!

Hydreigon Горыныч использует Double Hit.

Zweilous НедоГидра теряет 3 здоровья.

Zweilous НедоГидра теряет 3 здоровья.

Zweilous НедоГидра использует Double Hit.

Hydreigon Горыныч теряет 5 здоровья.

Hydreigon Горыныч теряет 5 здоровья.

Hydreigon Горыныч использует Dark Pulse.

Zweilous НедоГидра теряет 4 здоровья.

Zweilous НедоГидра использует Facade.

Hydreigon Горыныч теряет 4 здоровья.

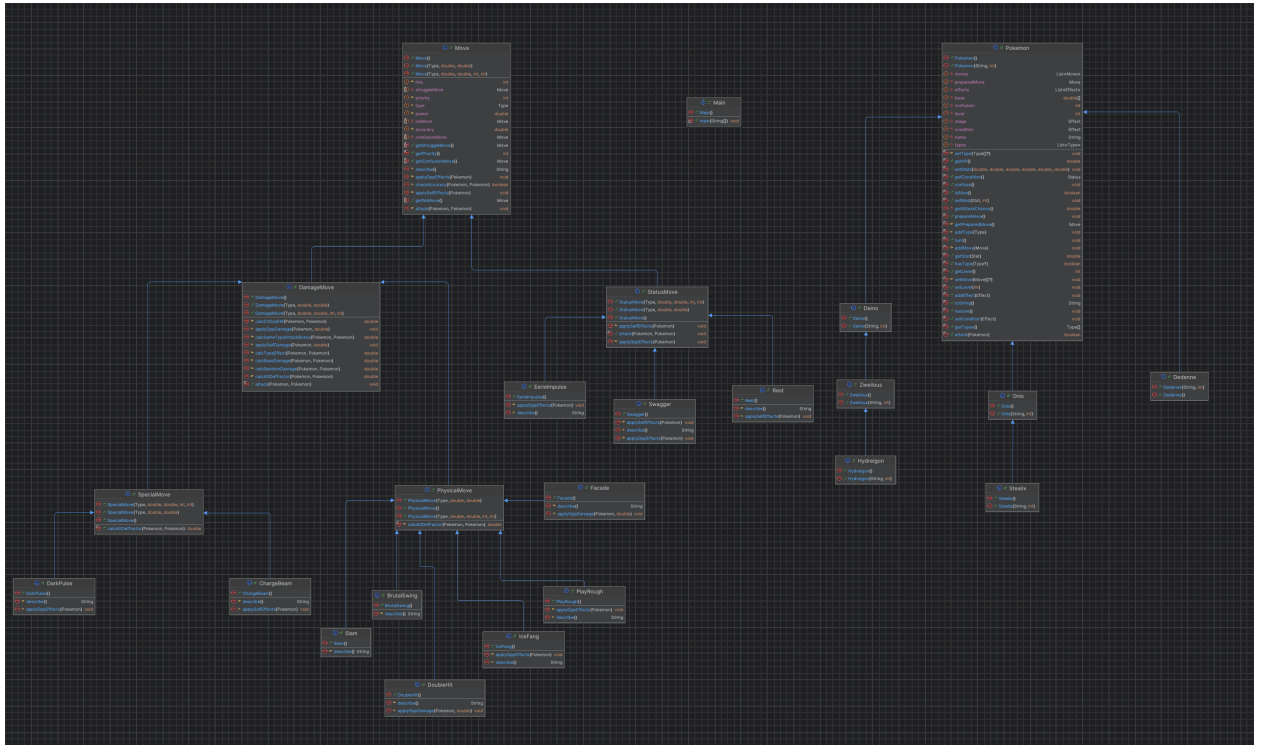
Hydreigon Горыныч теряет сознание.

В команде зеленых не осталось покемонов.

Команда полосатых побеждает в этом бою!

Диаграмма классов

Диаграмма реализованных классов представлена на рисунке.



Выводы по работе

В ходе лабораторной работы я познакомился с инкапсуляцией, наследованием и полиморфизмом в ООП, переопределением методов, модификаторами доступа. Также научился подключать к программе внешние jar-архивы и собирать jar-архив из нескольких файлов с зависимостями. Кроме того, мною была создана диаграмма классов программы.