

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

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №2 по дисциплине «Программирование»

Вариант 93055

Выполнил: Снагин Станислав Максимович
Проверил: Кулинич Ярослав Вадимович
Группа: Р3115

Санкт Петербург 2024 г.

Содержание

| | |
|---|---|
| Задания и цели работы..... | 3 |
| Исходный код программы..... | 4 |
| Диаграмма классов реализованной объектной модели..... | 5 |
| Результат работы программы..... | 6 |
| Вывод..... | 7 |

Задания и цели работы

Задание взято с сайта se.ifmo.ru:

Лабораторная работа #2

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. Все разработанные классы, не имеющие наследников, должны быть реализованы таким образом, чтобы от них нельзя было наследоваться.

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. Додумать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Введите вариант:

Ваши покемоны:

| Celebi | Bidoof | Bibarel | Deino | Zweilous | Hydreigon |
|---|---|---|--|---|--|
|  |  |  |  |  |  |
| Атаки: <ul style="list-style-type: none">✓ Confusion✓ Magical Leaf✓ Swagger✓ Dazzling Gleam | Атаки: <ul style="list-style-type: none">✓ Facade✓ Amnesia✓ Defense Curl | Атаки: <ul style="list-style-type: none">✓ Facade✓ Amnesia✓ Defense Curl✓ Waterfall | Атаки: <ul style="list-style-type: none">✓ Swagger✓ Focus Energy | Атаки: <ul style="list-style-type: none">✓ Swagger✓ Focus Energy✓ Double Hit | Атаки: <ul style="list-style-type: none">✓ Swagger✓ Focus Energy✓ Double Hit✓ Steel Wing |

Отчёт по работе должен содержать:

1. Текст задания.
2. Диаграмма классов реализованной объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Объектно-ориентированное программирование. Основные понятия: объекты, наследование, полиморфизм, инкапсуляция.
2. Понятие класса. Классы и объекты в Java.
3. Члены класса. Поля, методы, конструкторы. Модификаторы доступа.
4. Создание и инициализация объектов. Вызов методов.
5. Области видимости переменных.
6. Модификаторы **final** и **static**.
7. Пакеты, инструкция **import**.

Исходный код программы

Доступен на репозитории github:

https://github.com/ssnagin/ITMO_SPPO/tree/main/Programming/sem1/lab_2/pokemons93055

UML файл доступен в репозитории



Результат работы программы

Доступен по ссылке --

[https://raw.githubusercontent.com/ssnagin/ITMO_SPPO/refs/heads/main/Programming/sem1/
lab_2/pokemons93055/llogs/battle.llog](https://raw.githubusercontent.com/ssnagin/ITMO_SPPO/refs/heads/main/Programming/sem1/lab_2/pokemons93055/llogs/battle.llog)

Вывод

Благодаря данной лабораторной работе я разобрался в ООП на java, научился правильно генерировать UML диаграммы через PlantUML, подключать внешние библиотеки (Pokemon.jar) в проект, да и в целом как правильно организовывать код. Все эти навыки понадобятся в дальнейшей профессиональной деятельности.