

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
по дисциплине «Программирование»
Вариант 99

Выполнил
Разгоняев Максим Витальевич
Группа Р3131
Проверил
Бобрусь Александр Владимирович

Санкт-Петербург 2024

Оглавление

Задание.....	2
Диаграмма классов вычислений	4
Исходный код программы	4
Результат работы программы	4
Заключение	6

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. 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 `javados` - здесь.

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

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 (Рисунок 1)



Рисунок 1

Диаграмма классов вычислений

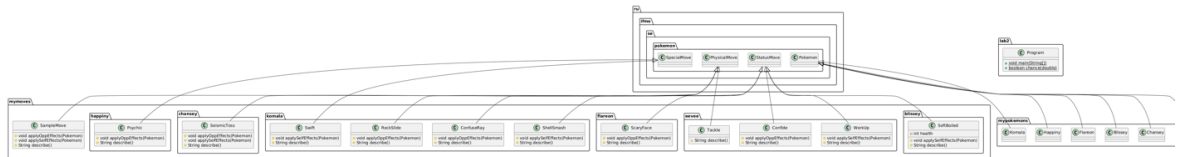


Рисунок 2

Также данная диаграмма доступна по [ССЫЛКЕ](#)

Исходный код программы

Исходный код программы доступен по [ССЫЛКЕ](#)

Результат работы программы

Komala Twix из команды красных вступает в бой!

Happiny Happiny из команды фиолетовых вступает в бой!

Komala Twix атакует методом ConfuseRay.

Happiny Happiny не замечает воздействие типа GHOST

Happiny Happiny промахивается

Komala Twix промахивается

Happiny Happiny атакует методом Psychic.

Komala Twix теряет 5 здоровья.

Komala Twix атакует методом RockSlide.

Happiny Happiny теряет 5 здоровья.

Happiny Happiny промахивается

Komala Twix атакует методом Swift.

Happiny Happiny теряет 7 здоровья.

Happiny Happiny промахивается

Komala Twix атакует методом Swift.

Happiny Happiny теряет 5 здоровья.

Happiny Happiny теряет сознание.

Chansey Chansey из команды фиолетовых вступает в бой!

Komala Twix промахивается

Chansey Chansey промахивается

Komala Twix атакует методом RockSlide.

Chansey Chansey теряет 5 здоровья.

Chansey Chansey атакует методом Psychic.

Komala Twix теряет 3 здоровья.

Komala Twix промахивается

Chansey Chansey промахивается

Komala Twix атакует методом RockSlide.

Chansey Chansey теряет 4 здоровья.

Chansey Chansey атакует методом Psychic.

Komala Twix теряет 5 здоровья.

Komala Twix теряет сознание.

Eevee Meow из команды красных вступает в бой!

Eevee Meow атакует методом Tackle.
Chansey Chansey теряет 8 здоровья.
Chansey Chansey теряет сознание.
Blissey Blissey из команды фиолетовых вступает в бой!
Eevee Meow атакует методом Tackle.
Blissey Blissey теряет 8 здоровья.

Blissey Blissey промахивается

Blissey Blissey промахивается

Eevee Meow промахивается

Blissey Blissey промахивается

Blissey Blissey промахивается

Blissey Blissey атакует методом Psychic.
Blissey Blissey теряет 5 здоровья.

Blissey Blissey атакует методом Psychic.
Blissey Blissey теряет 4 здоровья.
Оба покемона теряют сознание.
В команде фиолетовых не осталось покемонов.
Команда красных побеждает в этом бою!

Заключение

В ходе выполнения лабораторной работы я повторил основные принципы ООП и познакомился с его реализацией в Java, научился работать с классами, объектами, методами и конструкторами.