

## WEEK 1 REPORT

TTS: Nguyễn Khoa Đoàn

### 1.1. Yêu cầu (1): oop

- Mô tả: xây dựng chương trình java bất kỳ có sử dụng đầy đủ 4 tính chất của oop
- Điều kiện hoàn thành: Từ chương trình đã xây dựng trình bày về đã áp dụng oop như nào, (2) hiểu các khái niệm interface, static, ....., viết các biểu đồ UML liên quan

### 1.2. Yêu cầu (2) Đọc ghi file

- Viết chương trình java đọc ghi file theo 2 dạng binary và text
- Viết chương trình java thao tác với file và thư mục: list các file, đọc nội dung file

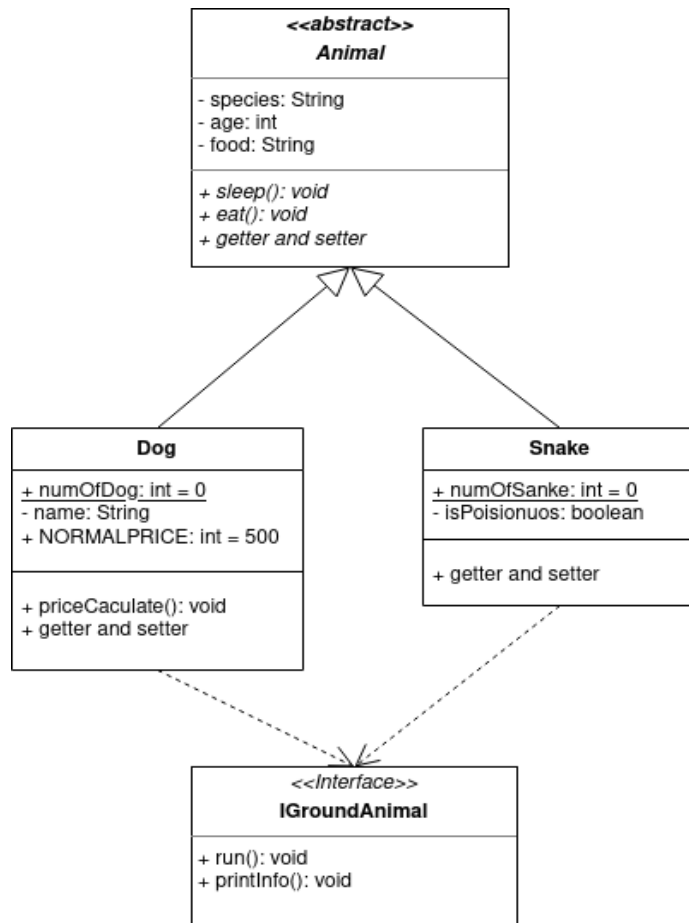
#####  
##

Source code: [LINK GITHUB](#)

### 1.1. Yêu cầu (1): oop

4 tính chất OOP:

- Trừu tượng (Abstract)
- Kế thừa (Inheritance)
- Đa hình (Polymorphism)
- Đóng gói (Encapsulation)



Mô tả đã áp dụng OOP như thế nào:

- Trừu tượng (Abstract): Thể hiện ở việc lấy những đặc điểm và hành động đặc trưng và đưa vào trong các Class. (Species, age, food, sleep, eat, ...)
- Kế thừa (Inheritance): Các lớp con kế thừa các thuộc tính và phương thức của lớp cha. Cụ thể lớp Dog và Snake kế thừa từ lớp Animal.
- Đa hình (Polymorphism): Thể hiện qua việc lớp Dog và Snake override lại các phương thức của lớp cha
- Đóng gói (Encapsulation): Thể hiện việc che giấu thông tin, sử dụng từ khóa private, package, toàn bộ các lớp được đóng gói trong package MyZoo
- Khái niệm Interface: Do class hoặc abstract class không hỗ trợ đa kế thừa, nên interface giải quyết vấn đề trên, mọi thuộc tính và phương thức của interface đều là abstract, dùng từ khóa implement để cài đặt, và phải cài đặt toàn bộ các phương thức.
- Static: thuộc tính và phương thức static được chia sẻ, cấp phát bộ nhớ dùng chung, có thể overload nhưng không thể override, static không thể được các phương thức không phải là static sử dụng. Trong UML sử dụng thuộc tính static numOfDog và numOfSnake để đếm số lượng Dog và Snake được khởi tạo

- Final: Final không thể ghi đè, sử dụng từ khóa final,

## 1.2. Yêu cầu (2) Đọc ghi file

### - Đọc ghi file text

Khai báo đường dẫn file cần đọc|ghi

Tạo Buffer để tạo luồng đọc|ghi file. `BufferedWriter` và `BufferedReader` kế thừa từ lớp `Writer`, `Reader`, lớp `Writer`, `Reader` kế thừa lớp `Object`. `FileWriter` và `FileReader` được sử dụng để ghi|đọc vào tệp tin

Tạo khối try-catch để xử lý ngoại lệ vào ra `IOException`. Dùng `write()` để ghi và `readLine()` để đọc. Đọc và ghi mặc định là 8192 ký tự.

### - Đọc ghi file nhị phân

Ghi sử dụng `FileOutputStream` và `DataOutputStream`. `DataOutputStream` tạo luồng đầu ra dữ liệu mới để ghi dữ liệu vào luồng đầu ra cơ bản được chỉ định. `OutputStream` ở đây là `FileOutputStream`

Đọc sử dụng `DataInputStream` và `FileInputStream`. `DataInputStream` là một lớp trong Java được sử dụng để đọc các dữ liệu nguyên thể và các kiểu dữ liệu cơ bản từ một `InputStream`. Ở trường hợp này `InputStream` là `FileInputStream`.

Chương trình sẽ ghi số nguyên 100 sử dụng `writeInt()` và ghi số thực 9.5 sử dụng `writeDouble()`. Sau ghi được ghi 100 và 9.5 sẽ được lưu vào file `mydata.bin` và không thể đọc được

Chương trình sẽ đọc nhị phân 2 số vừa ghi thông qua `readInt()` và `readDouble()`. Và sẽ đọc được cụ thể 2 số 100 và 9.5

### - Thao tác với thư mục

Khai báo tên thư mục. Tạo mảng `File[]` để lưu các file cần kiểm tra, dùng phương thức `listFiles` để liệt kê các file

Sử dụng vòng lặp for duyệt qua toàn bộ phần tử mảng files, dùng phương thức `isFile` kiểm tra có phải file không và in ra tên file sử dụng phương thức `getName()`

Sử dụng vòng lặp for để duyệt qua toàn bộ phần tử mảng files, dùng phương thức `isFile` kiểm tra có phải file không và sử dụng khối try catch xử lý ngoại lệ `IOException`. Dùng `BufferedReader` và `FileReader` để đọc nội dung file

