

LLD 2 : OOP Terminology

[Q: DPM]

AND

SOLID Design Principles Intro

Pillars of OOP

Ab

Re

TG / 5
B

Java :
The Complete
Reference
(Part I)

Effective Java

Agenda



QUESTION

ANSWER

→ ① Key terms related to UOP

→ Abstraction

→ Encapsulation

→ Inheritance

→ Polymorphism

→ ② Interfaces

+ Use interfaces vs inheritance

isA vs HasA

③ Design Principles

→ SOLID

TSP

→ C-I-D

Sat
9 - 11 PM

Design Patterns

1 Principle of OOP
3 Pillars of OOP

4 Pillars of OOP

Support → language constructs that allow
to follow principle of OOP

Principle of OOP

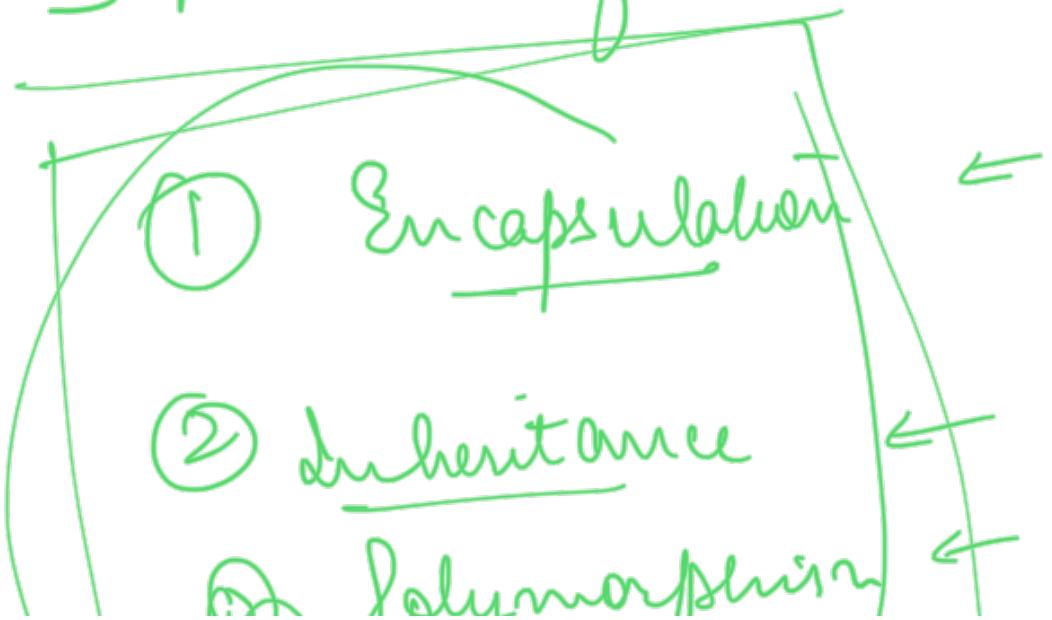
Abstraction → Idea Representation of Ideas

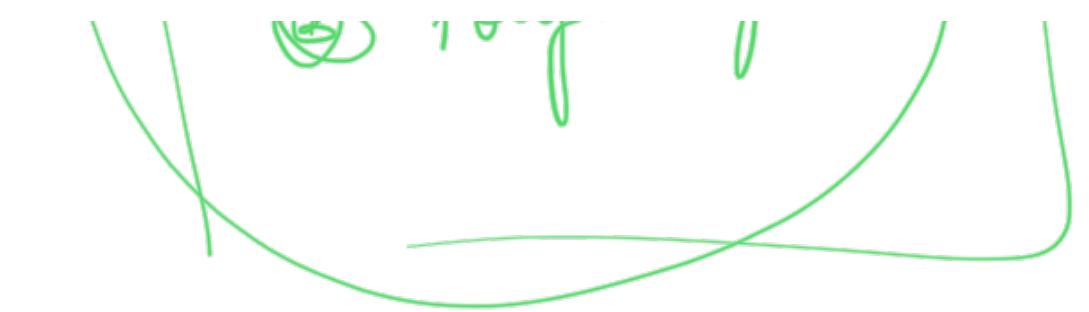
Principle of up all the ideas that exist
within a software system

Zoo

Animal, Gatekeep, Payment Service

3 Pillars of OOP





Encapsulation



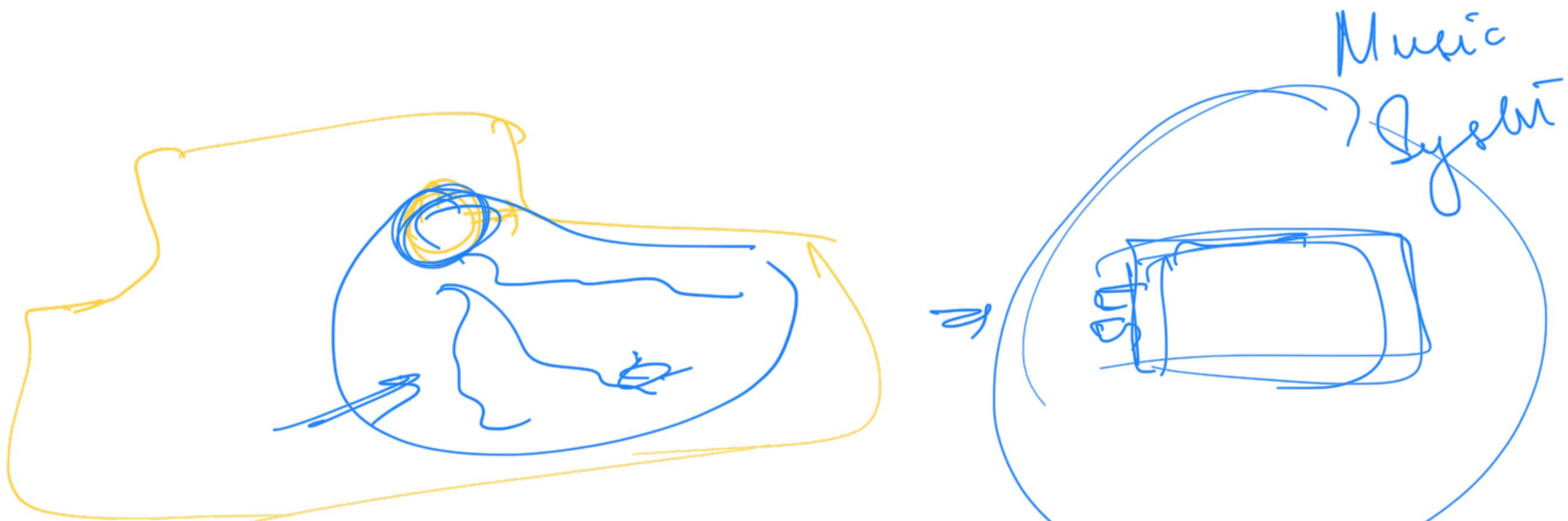
Container | wrapped

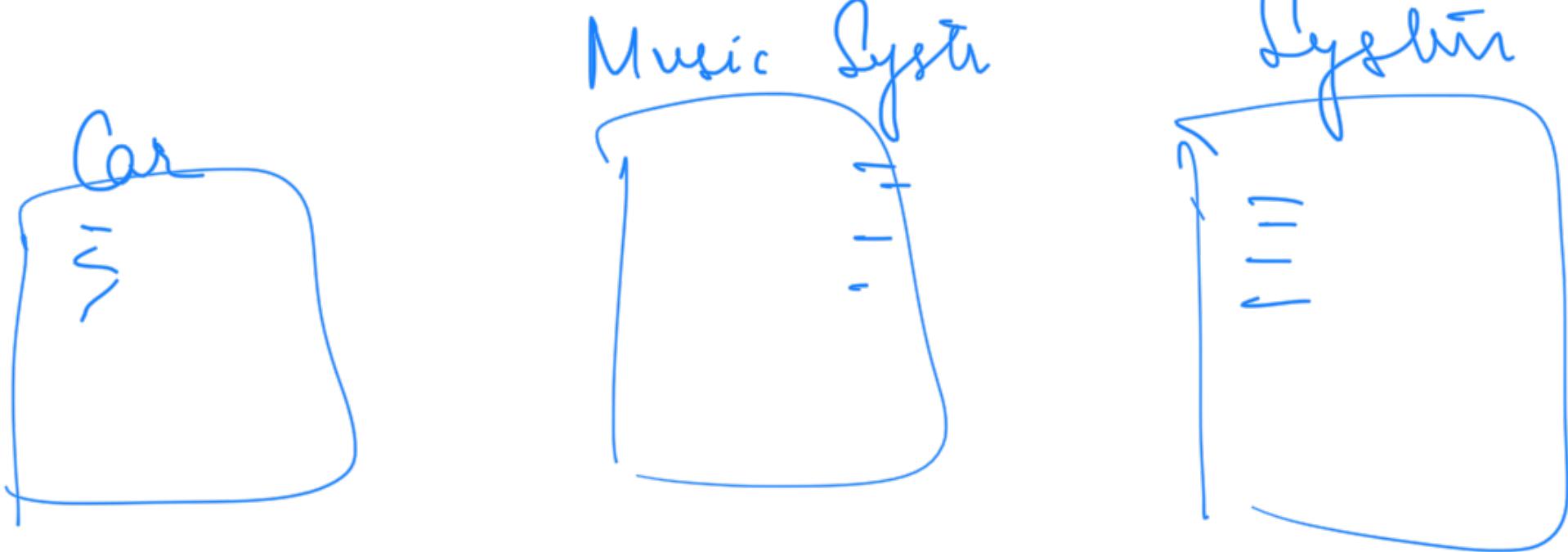


- Containing data and behaviour of an entity together
- Prevent data and behaviour from illegitimate access → Access Maxim



\rightarrow Only expose the methods / properties that you want to allow others to use





① Hide details of the system

② Expose only what you are fine with
others using

③ Holding data and behavior together

Restricting yourself to support that behavior

- with Only In Java

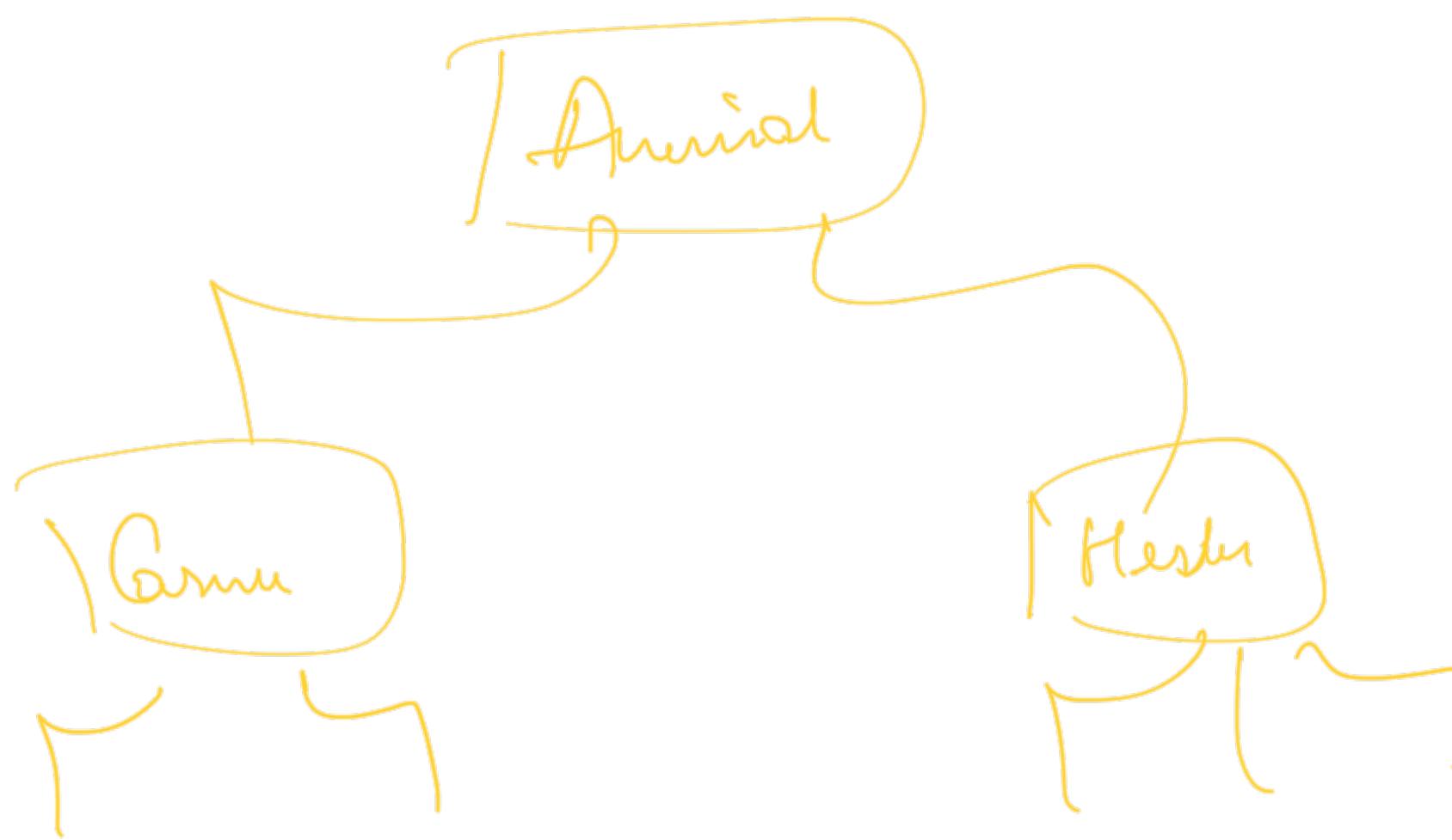
Java

② Inheritance

→ to represent hierarchy in the ideas (abstraction)

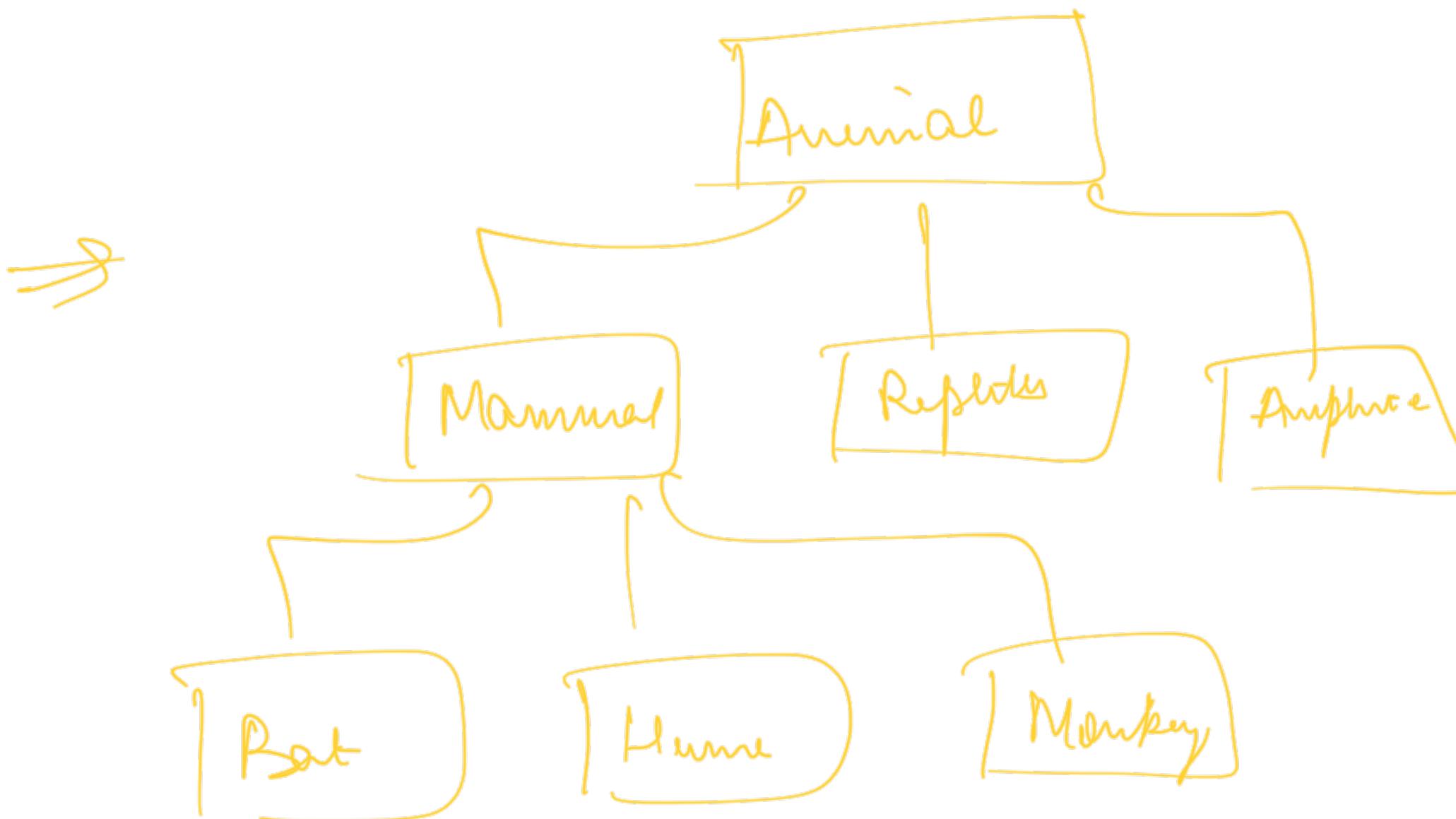
HRMS ⇒ Org Chart





② Dokumentation

(3) Long wavy





Count Animal (Animal) {

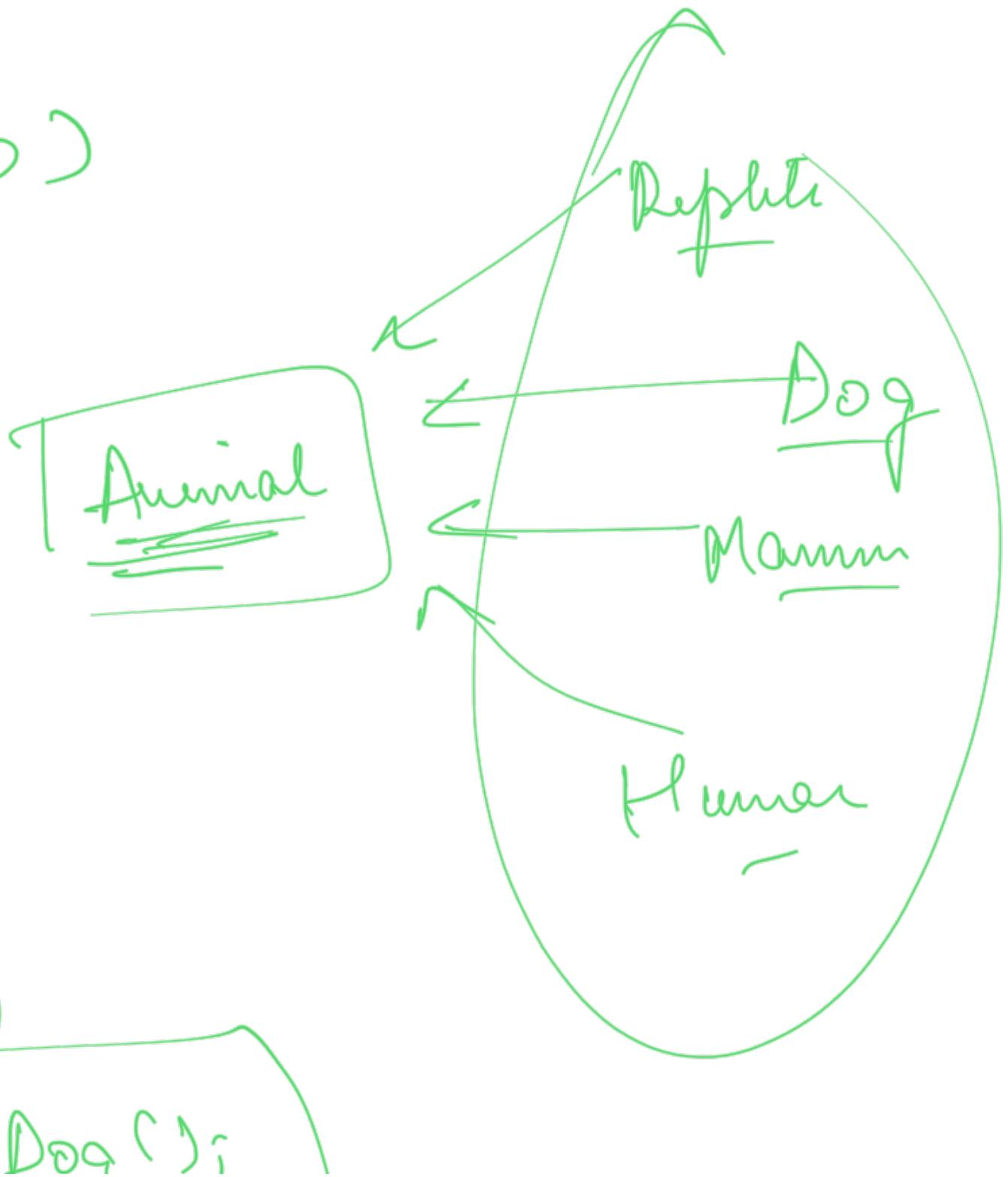
}

$\max(\text{int } a; \text{int } b)$

$\max(1.0, 2.0)$

~~Poly~~
~~morphism~~

Many + forms

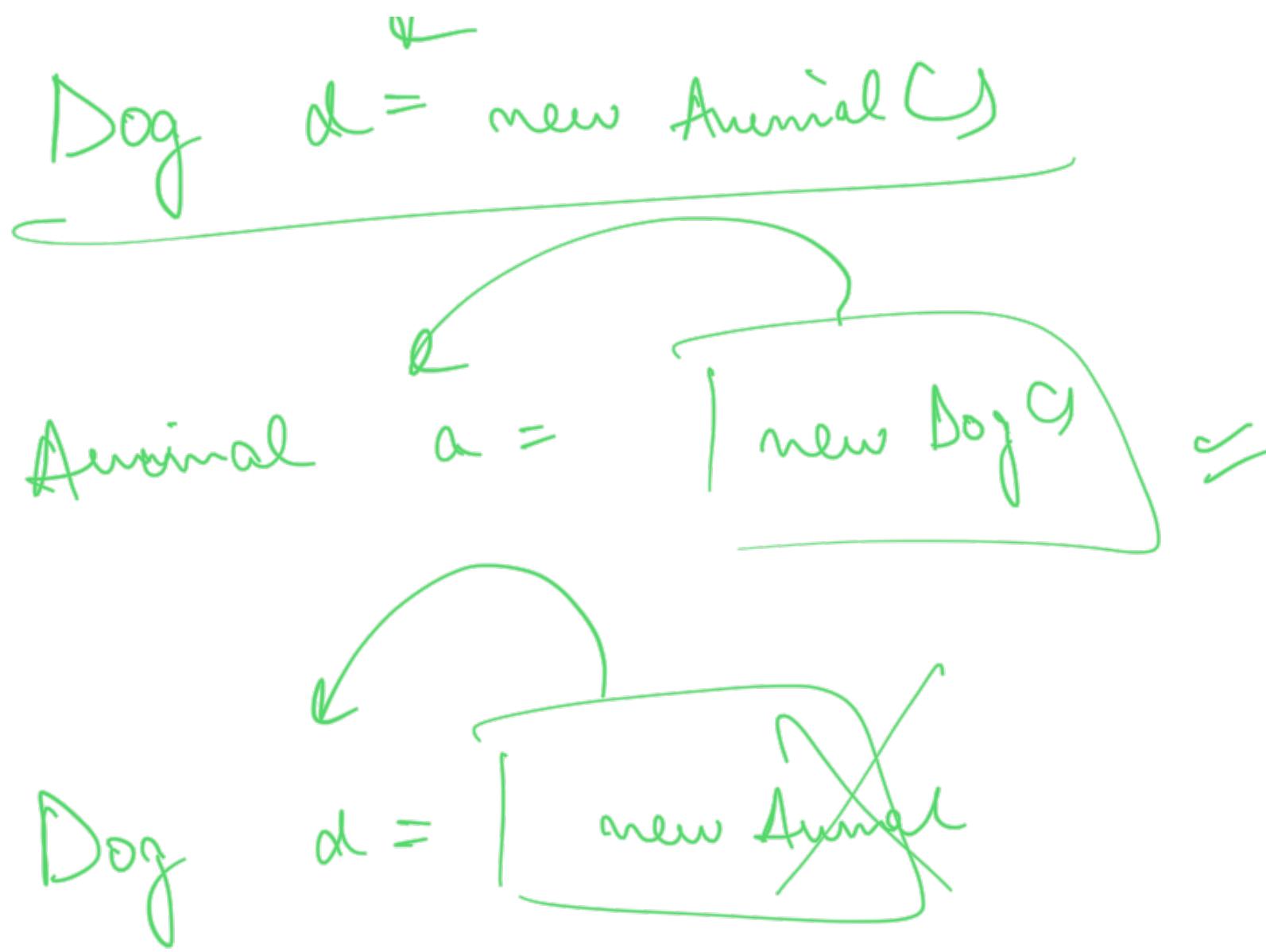


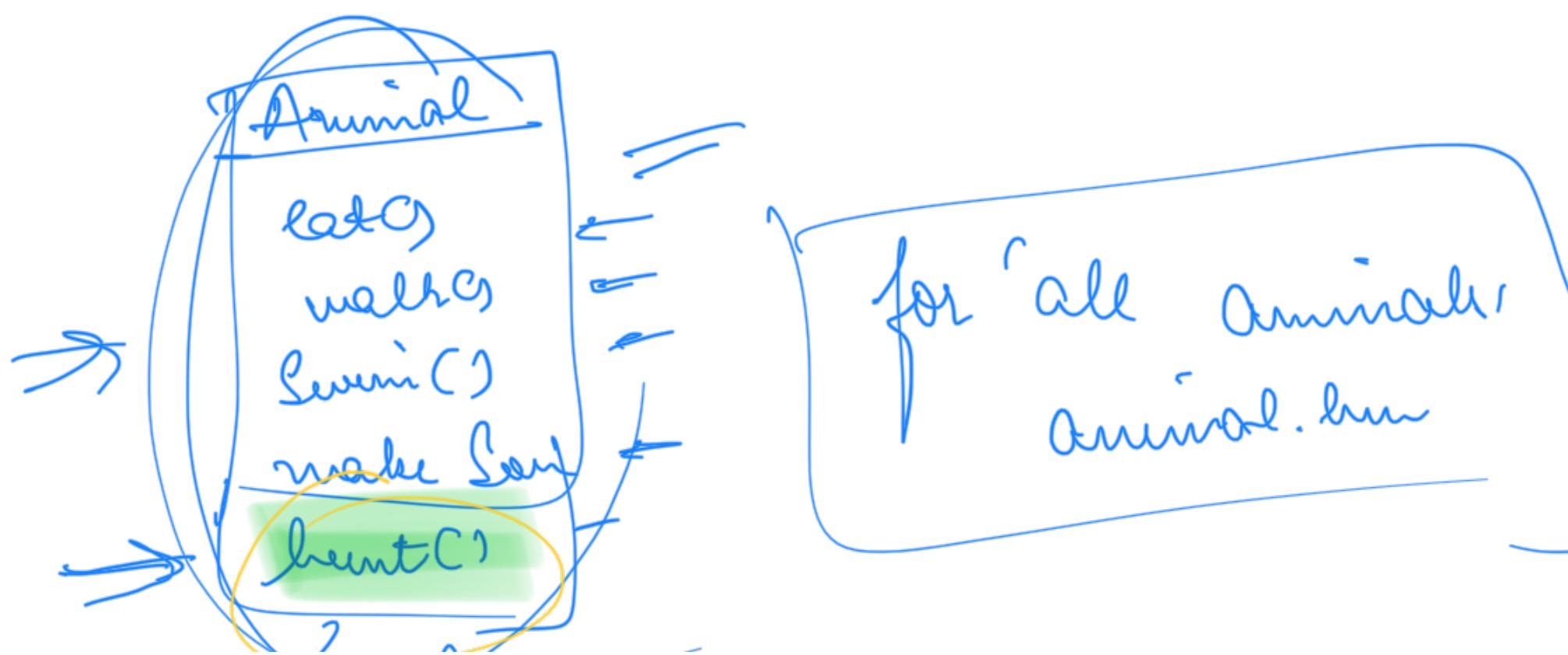
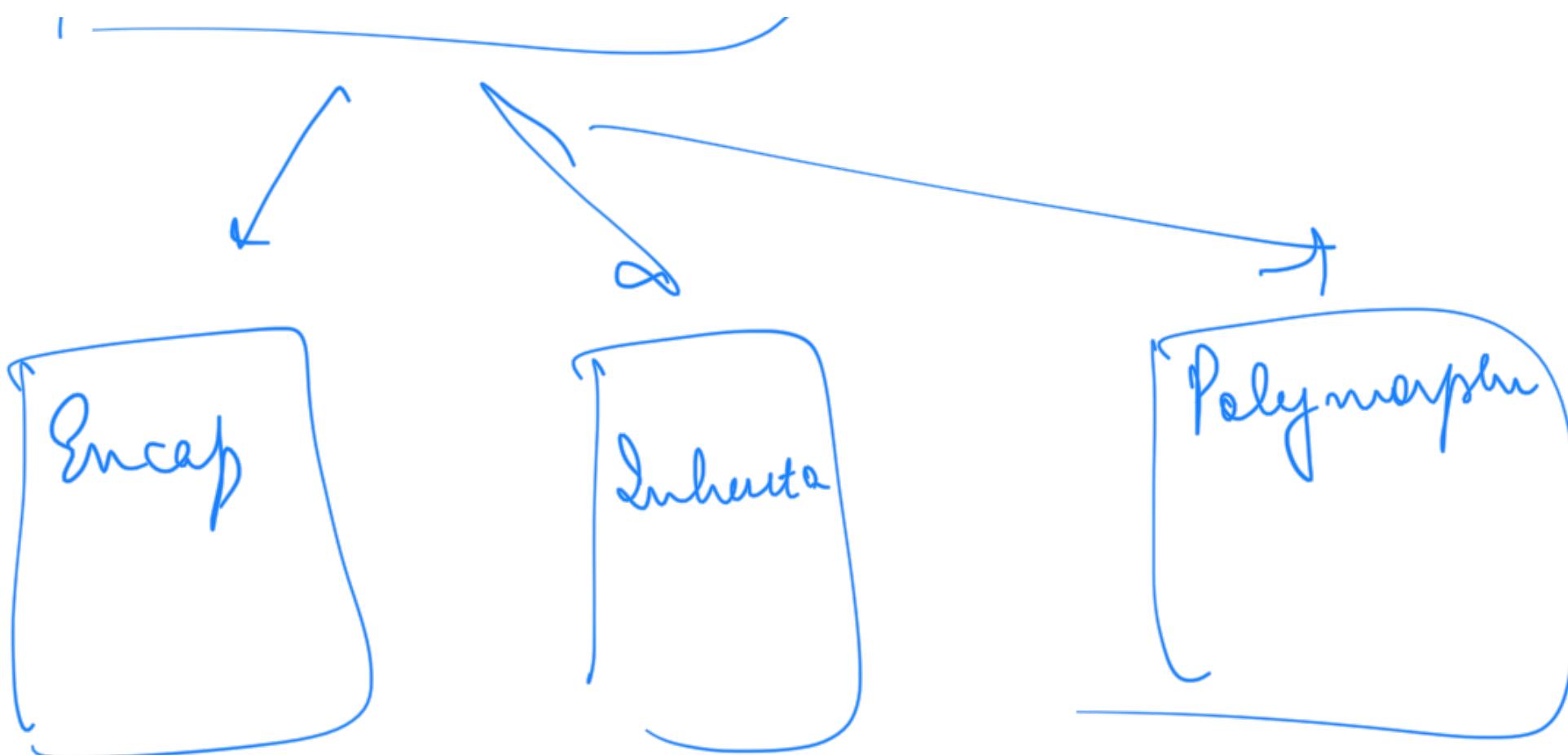
`T A n i m a l * a = new Dog();`

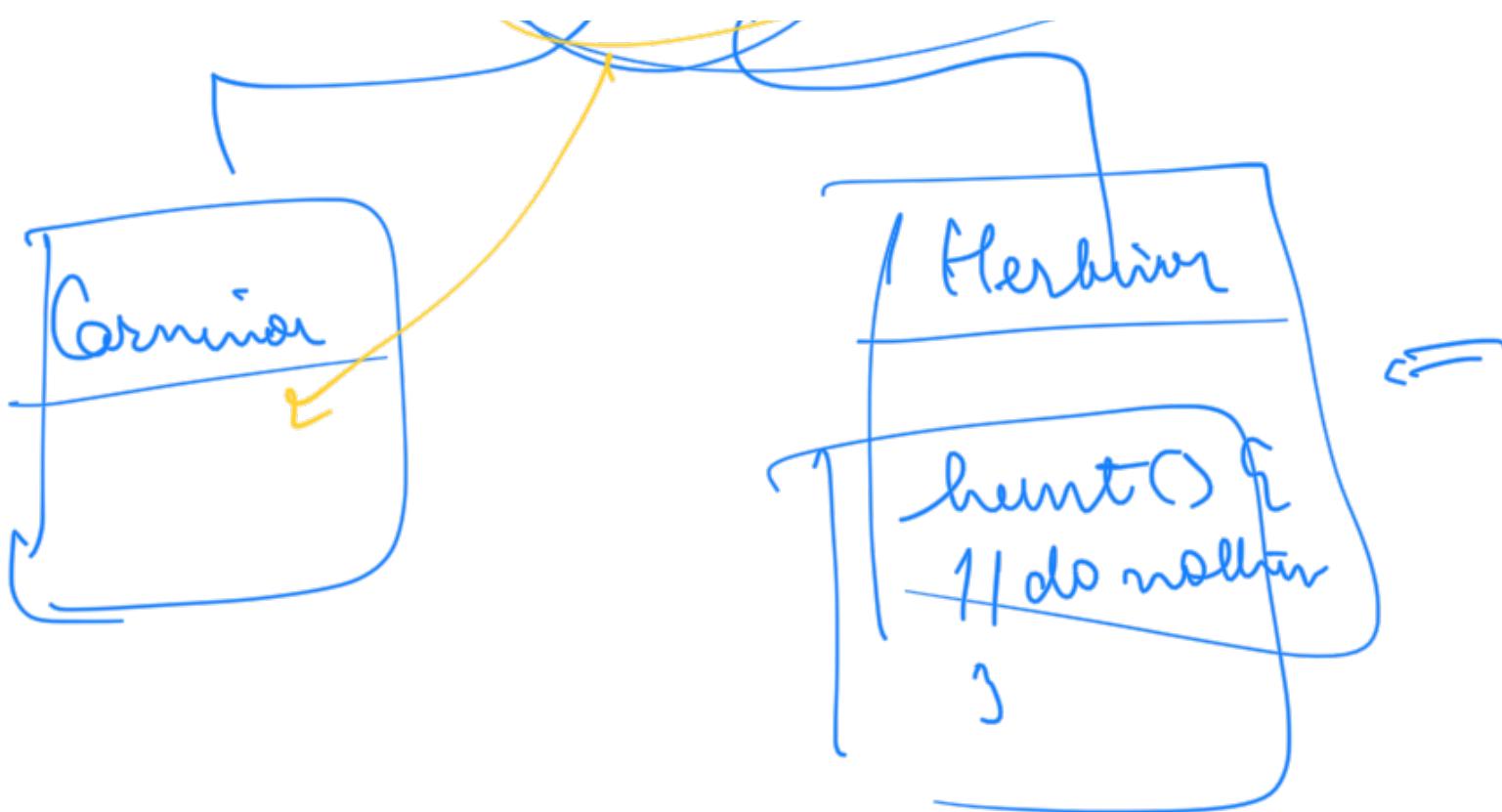
Human " - " ()

Favour interface instead of
inheritance

- Allows us to up hierarchy
- Code Reuse
- Extend a parent class

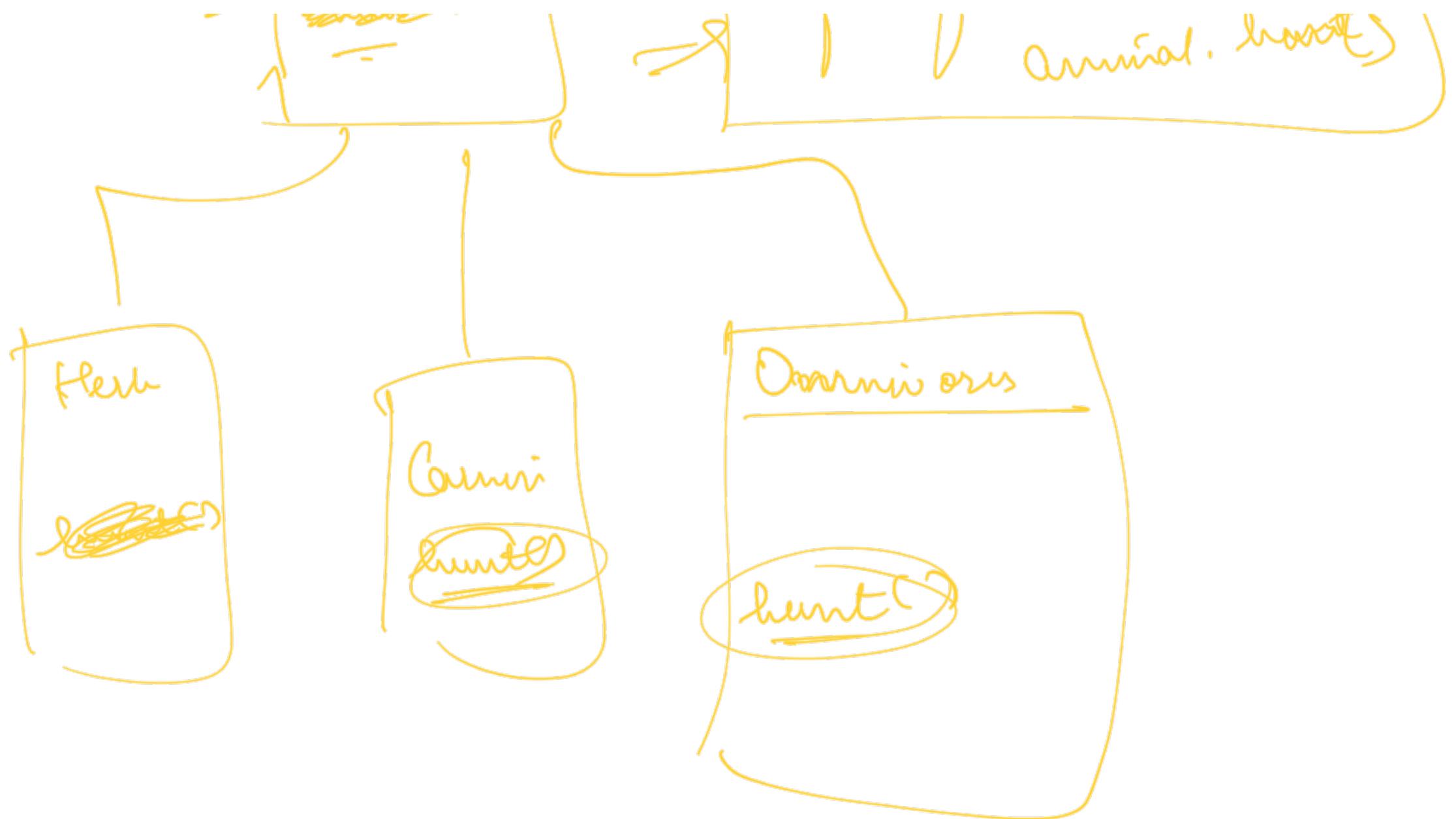




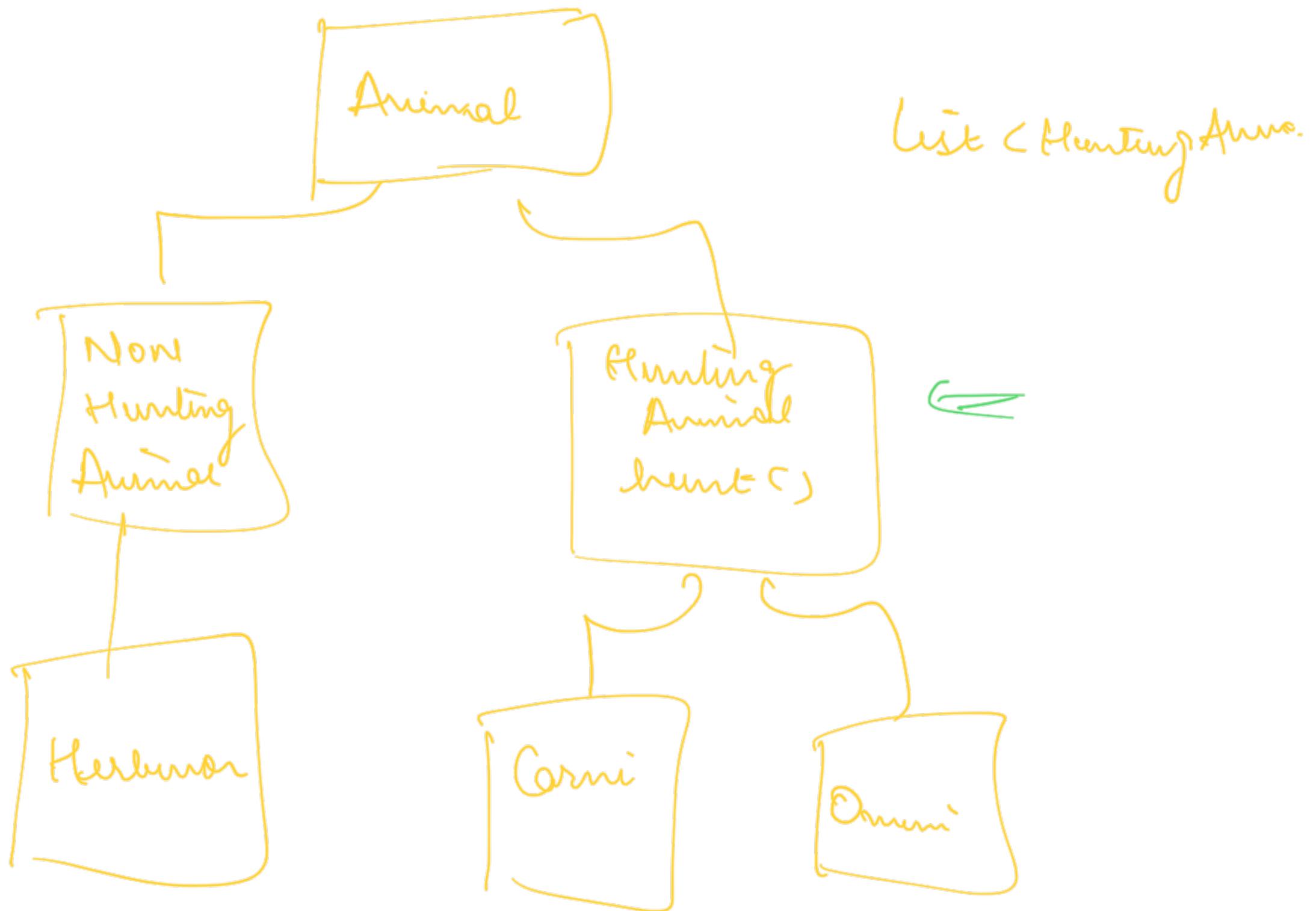


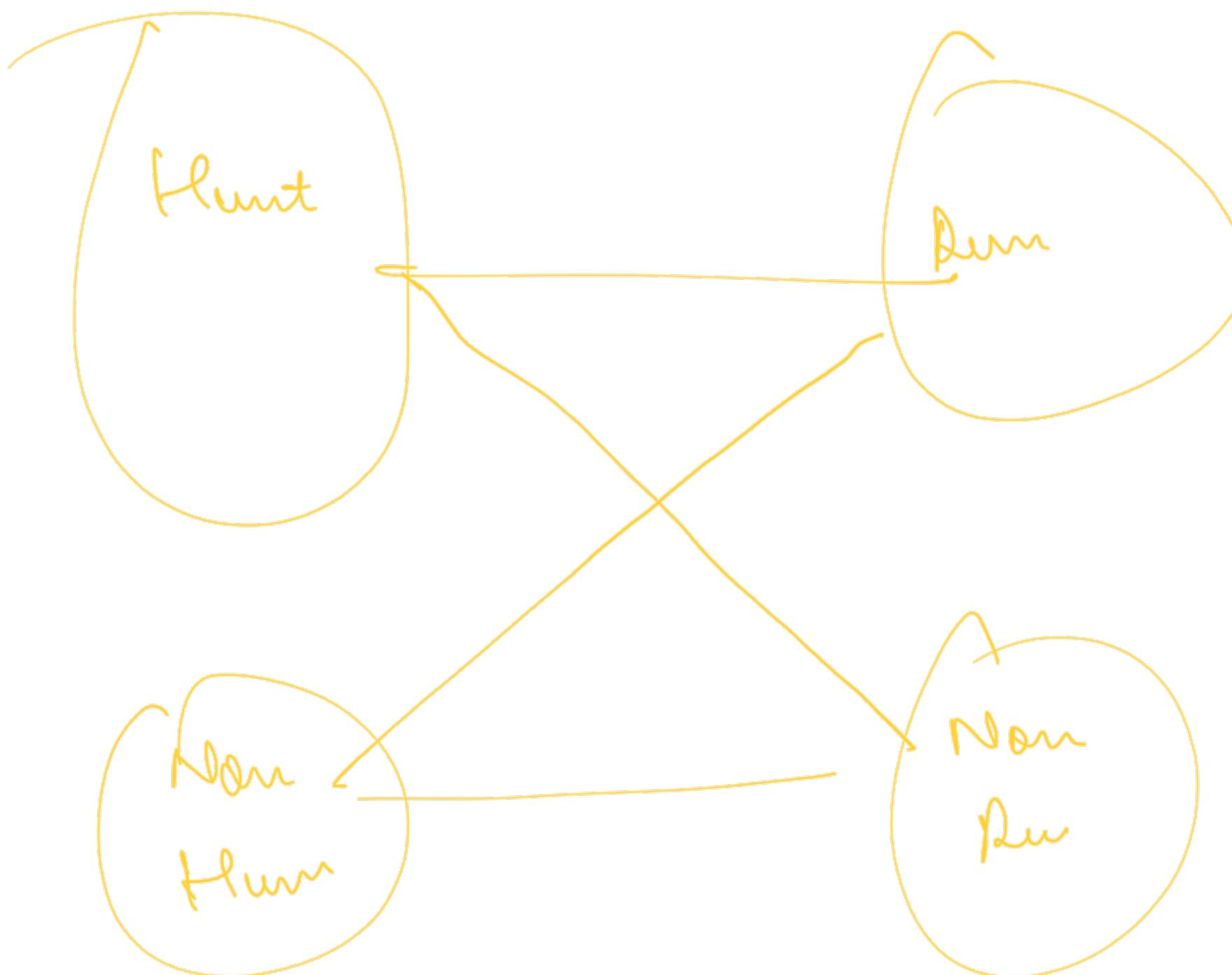
→ don't include a behaviour in a class unless it is a natural property of that class





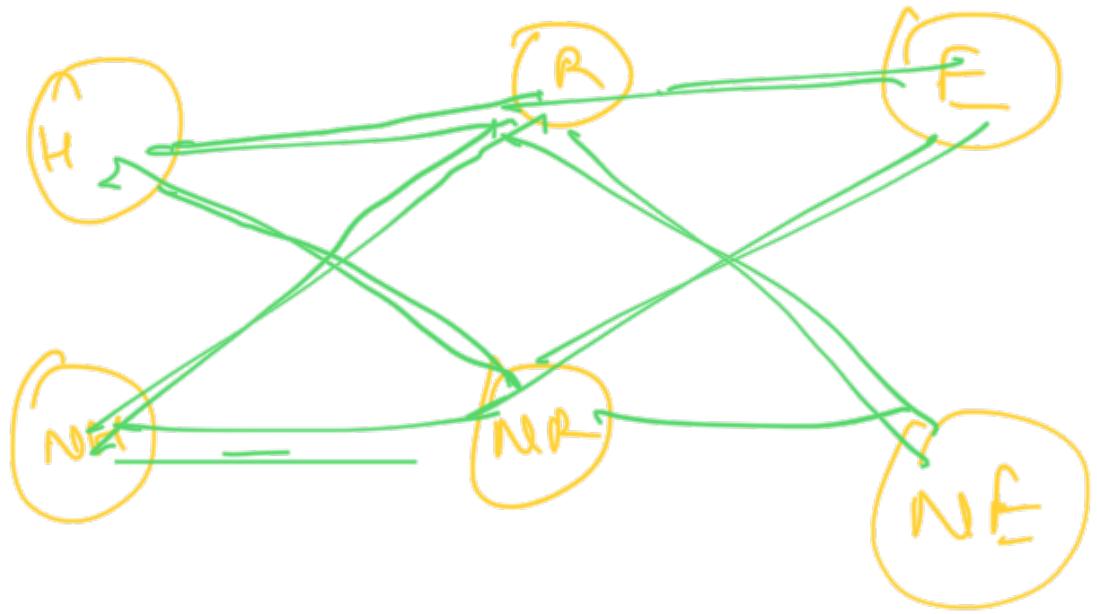
Client req: All the animals that can hunt



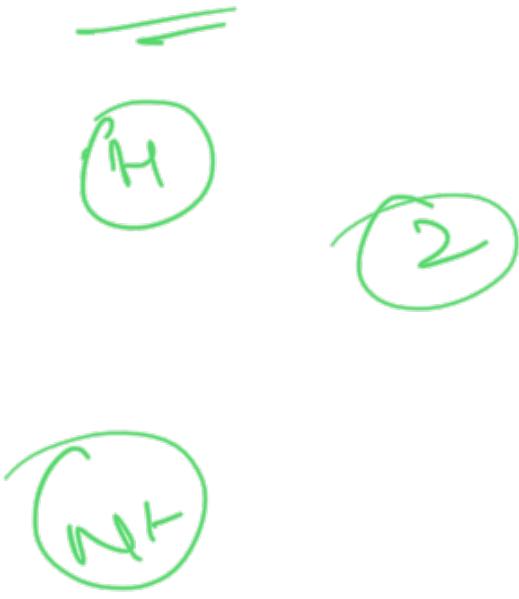
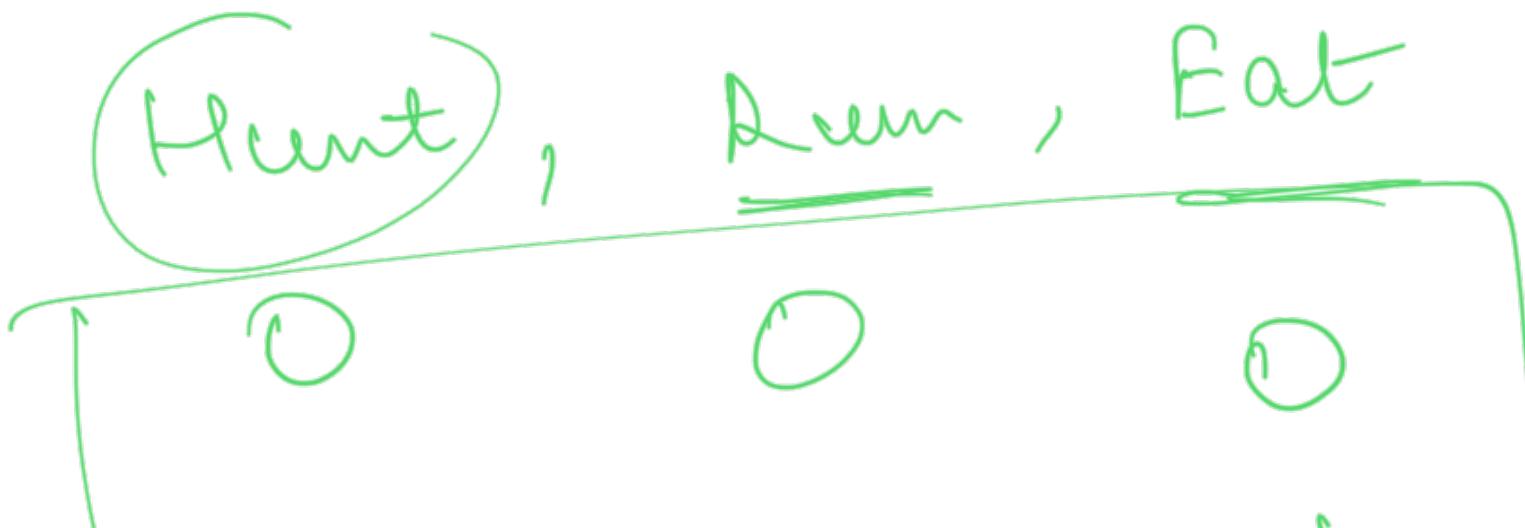


10 Such behaviour

Hunt

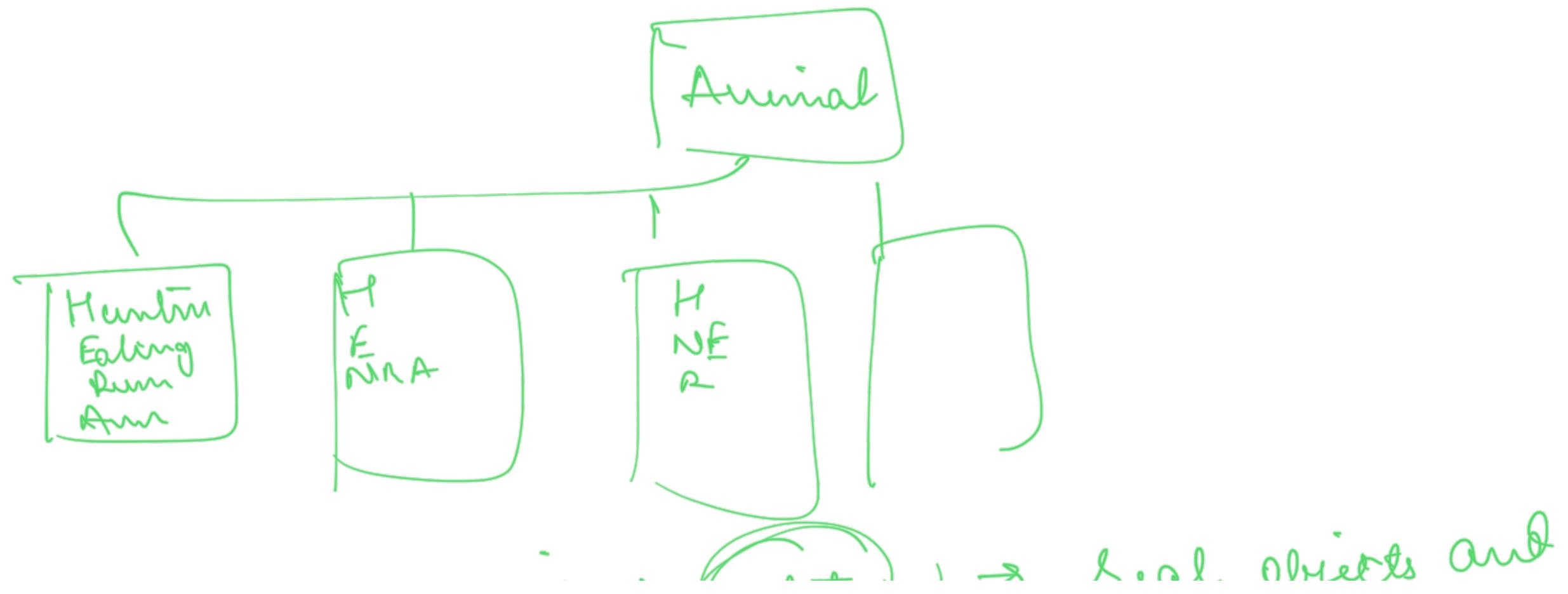


Hunt, Run



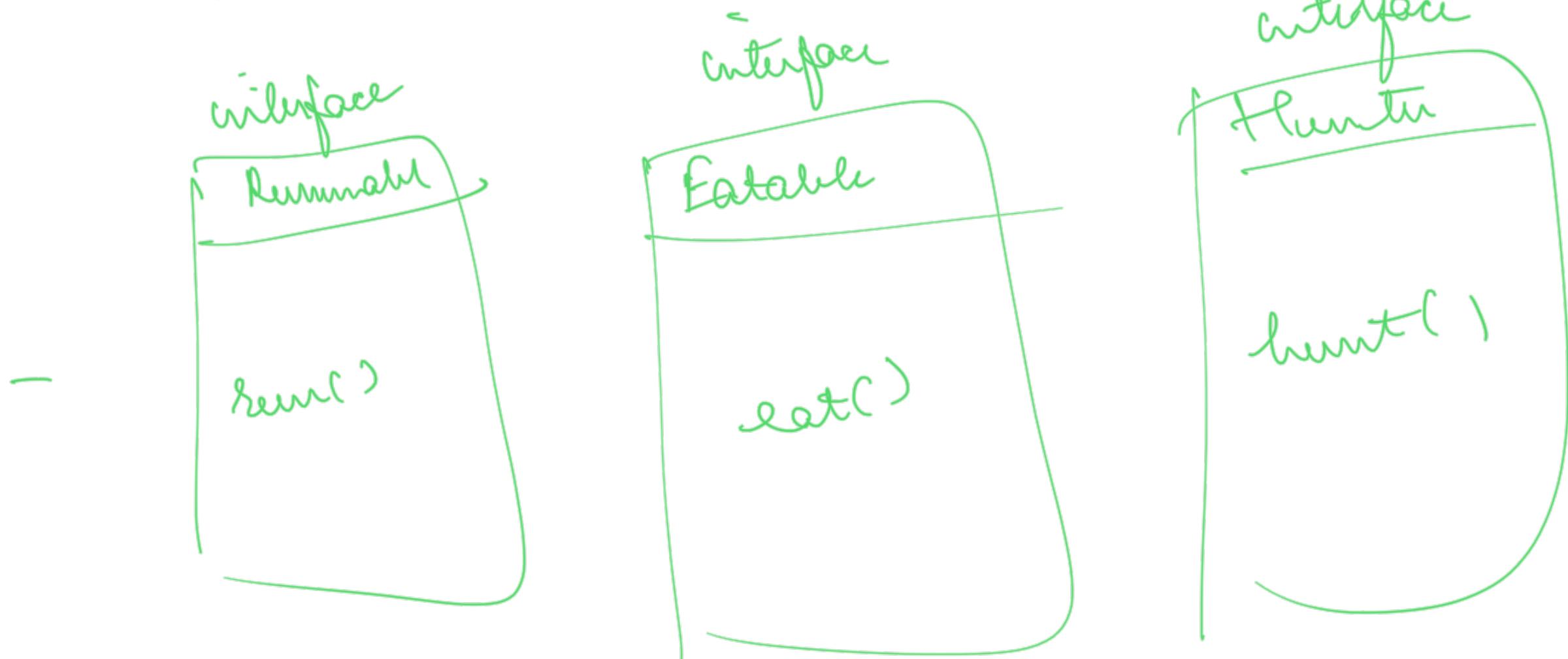
↑ | | ↓

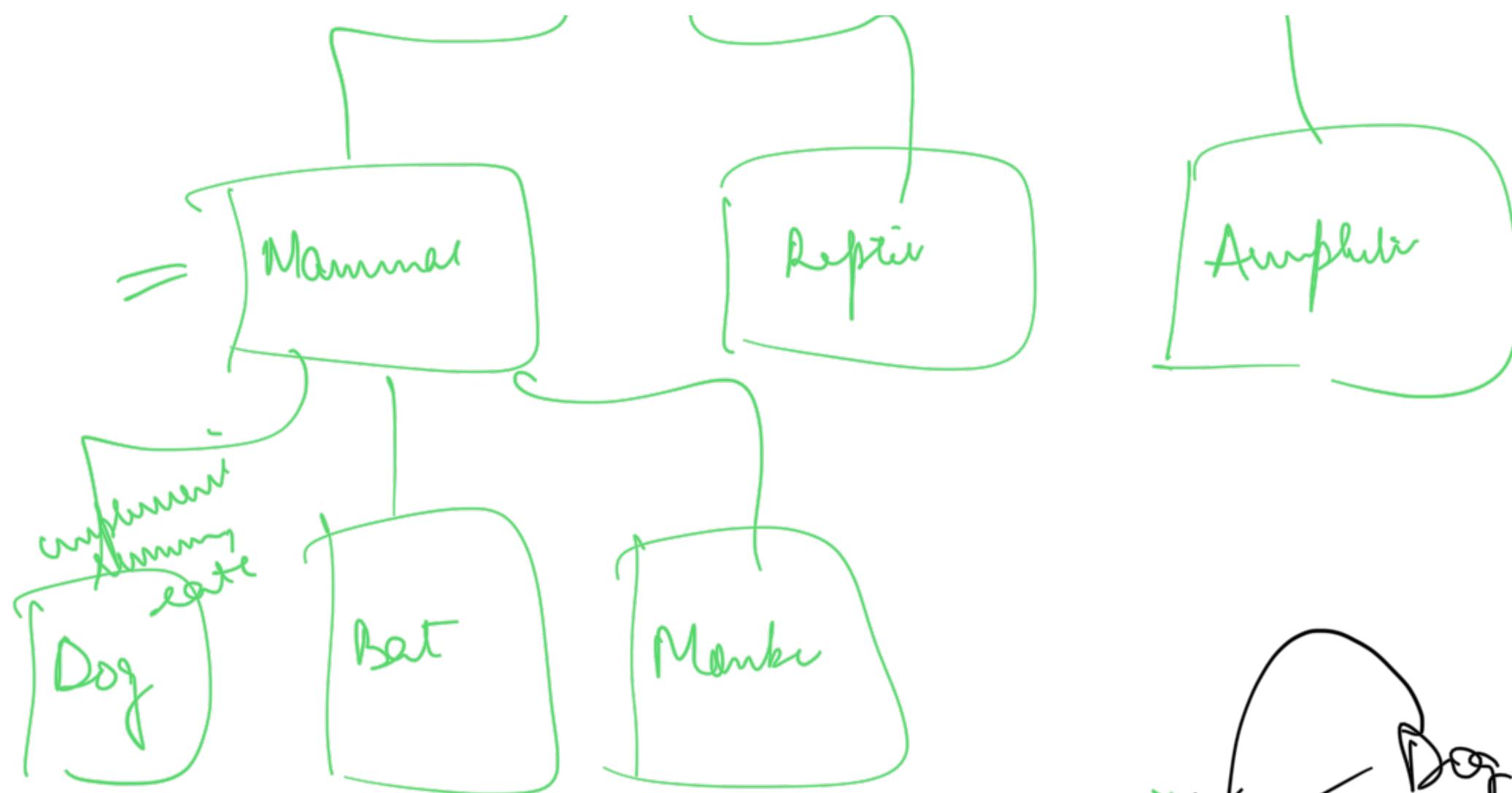
$2^{10} \approx 1024$



Class: Blueprint of Entity \Rightarrow their physical pr

Interfaces: Blueprint of behaviour





Polymorphism

Eatable a = new Dog();
