CPSC-224 Software Development

UML ①

Yu Wang

wangy2@gonzaga.edu

Feb 21, 2025

Announcement

- Homework2 due day: Feb 28th
- Exam Day: Feb 28 (Announcement in Canvas)
- Homework0 due day: March 3rd

Daily Attendance (01)

- Scan the QR Code

Daily Attendance (02)

- Scan the QR Code

Review - Last Class

- ✓ We learned – Why do we need a report for a project?
- ✓ We learned – How to write a (project) report?
- ✓ We learned – A report should include
  - Cover Page
  - Abstract
  - Introduction
  - Problem Statement
  - System Design & Implementation
  - Test & Results
  - Conclusion & Future Enhancements

UML- Unified Modeling Language

GONZAGA UNIVERSITY

What is Unified Modeling Language(UML)?

- UML, short for Unified Modeling Language, is a standardized modeling language used in software engineering to visualize, specify, construct, and document the artifacts of a system.

UML- Unified Modeling Language

GONZAGA UNIVERSITY

Why Use UML?

- Helps in planning software design before coding.
- Makes it easier to understand complex systems.
- Improves communication between team members.
- Provides documentation for future reference

UML- Categories

GONZAGA UNIVERSITY

UML is divided into two main categories:

Structural Diagrams (Static)

Behavioral Diagrams (Dynamic)

UML Diagrams

GONZAGA UNIVERSITY

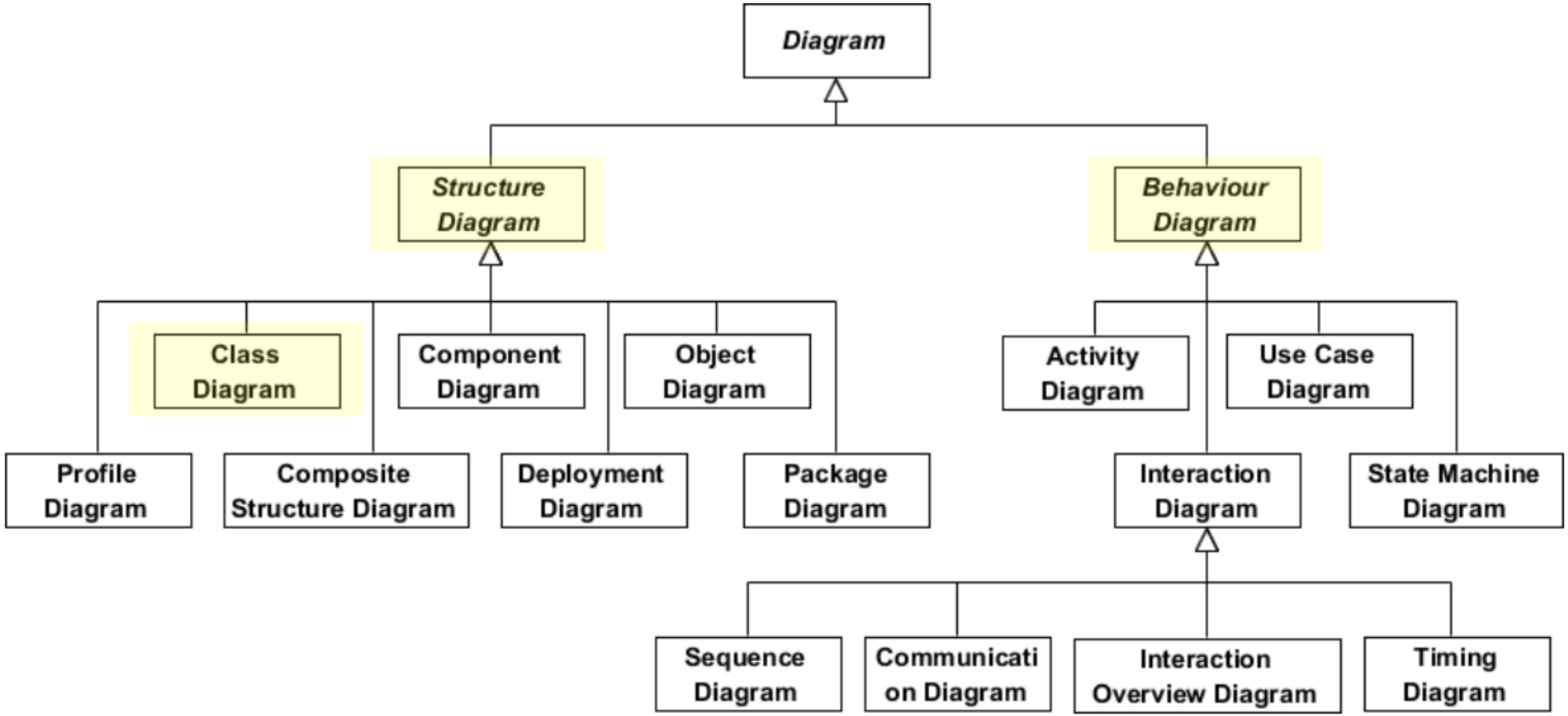
Structural Diagrams (Static)

These show the organization of a system

Behavioral Diagrams (Dynamic)

These show system interactions over time

UML Diagrams



Structural Diagrams

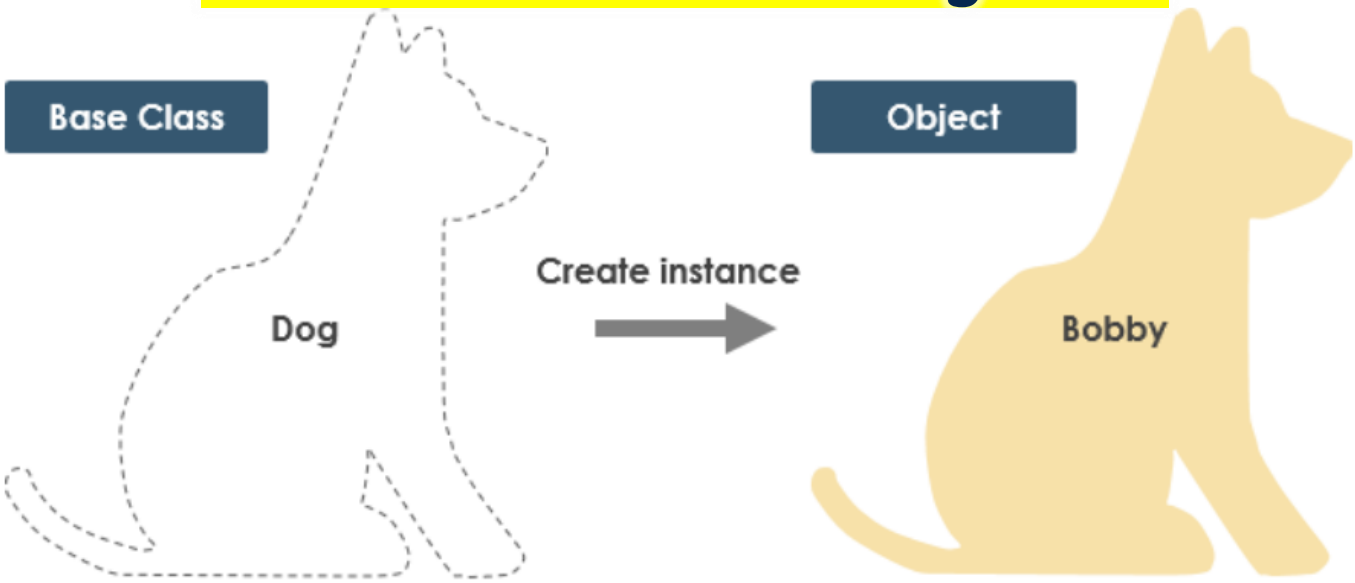
Category	Diagram Type	Purpose
Structural Diagrams (Static View)	Class Diagram	Defines **classes, attributes, methods**, and their **relationships**.
	Component Diagram	Represents **high-level software components** (modules, APIs, microservices) and their dependencies.
	Deployment Diagram	Represents **hardware and software deployment** (servers, nodes, devices, etc.).
	Object Diagram	Shows **specific instances of objects** at a given moment in time.
	Package Diagram	Groups related **classes and components** into **packages** for organization.
	Composite Structure Diagram	Shows the **internal structure of a class or component** (useful for embedded systems).
	Profile Diagram	Defines **custom stereotypes** to extend UML for **domain-specific modeling**.

Behavioral Diagrams

Behavioral Diagrams (Dynamic View)	Use Case Diagram	Represents **interactions between users (actors) and the system**.
	Activity Diagram	Models **workflows, business processes**, and **decision flows**.
	State Machine Diagram	Shows **states and transitions** of an object (e.g., a traffic light system).
	Sequence Diagram	Describes **object interactions over time** (message passing).
	Communication Diagram	Similar to Sequence Diagram but focuses on **message flow between objects**.
	Interaction Overview Diagram	A **high-level view** combining multiple sequence diagrams to show system flow.
	Timing Diagram	Shows **state changes over time**, used in **real-time systems**.

UML – Class Diagram Example

Class and Object

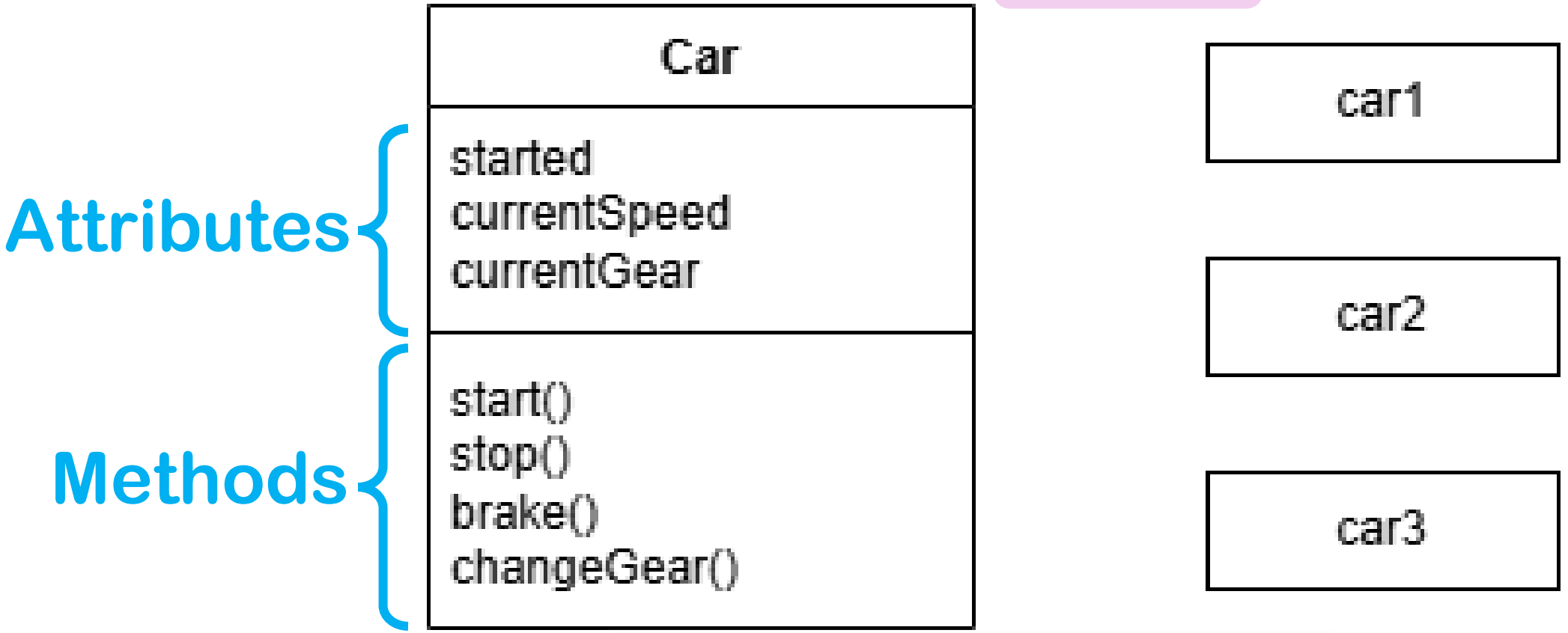


Properties	Methods
Color	Sit
Eye Color	Lay Down
Height	Shake
Length	Come
Weight	

Property Values	Methods
Color: Yellow	Sit
Eye Color: Brown	Lay Down
Height: 17 in	Shake
Length: 35 in	Come
Weight: 24 pounds	

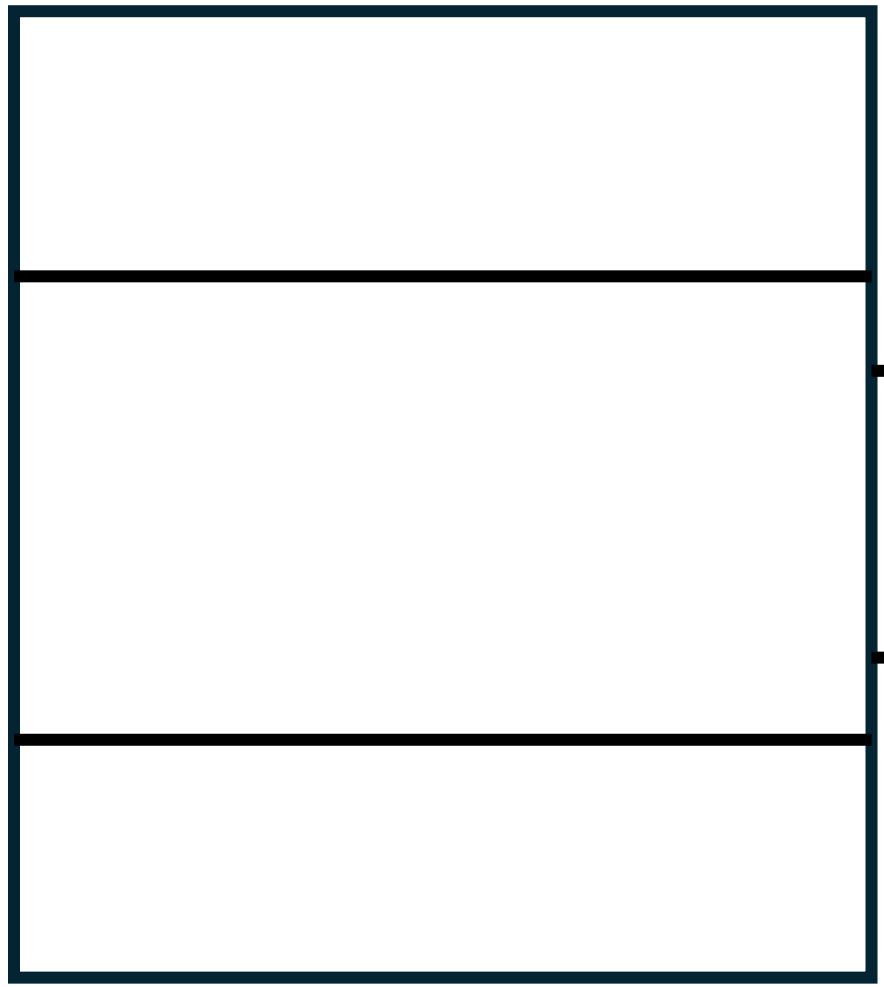
UML – Class Diagram Example

UML



Class and Object

UML – Class Diagram Example



Class

Attributes

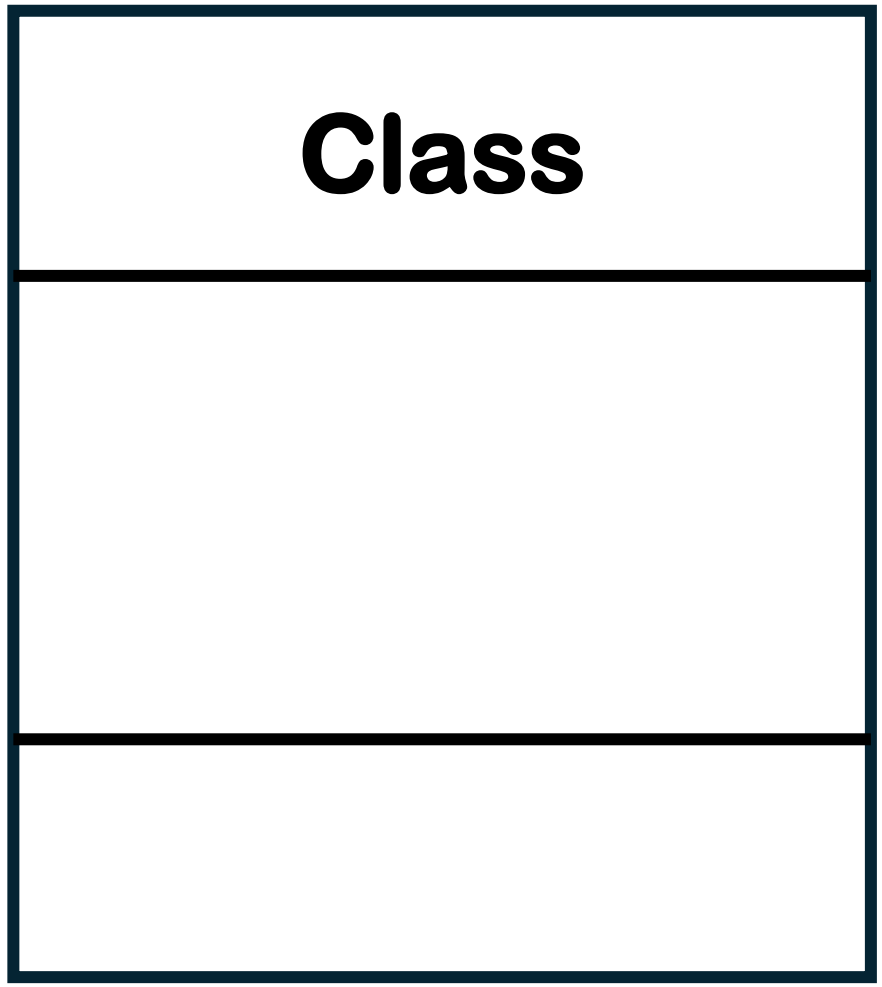
Methods

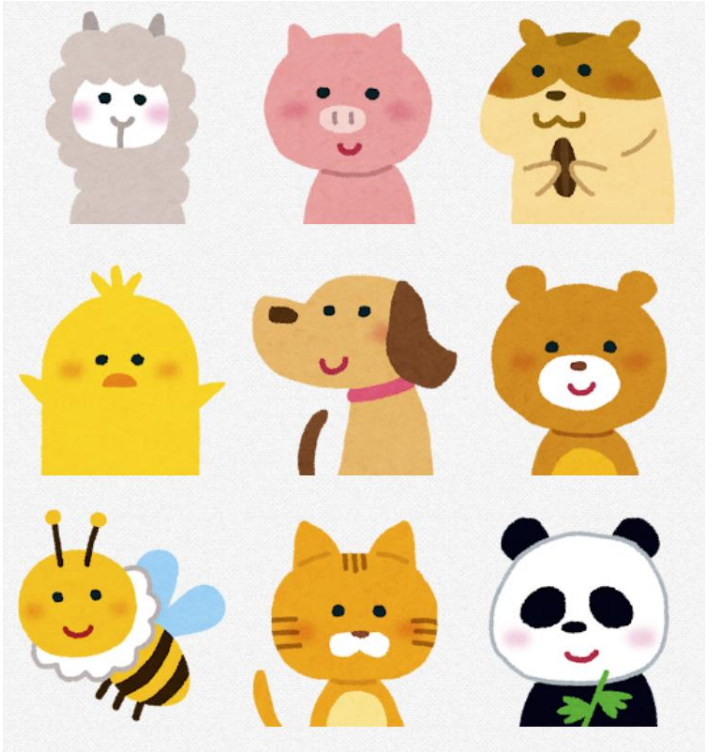
Relationships

UML – Class Diagram Example

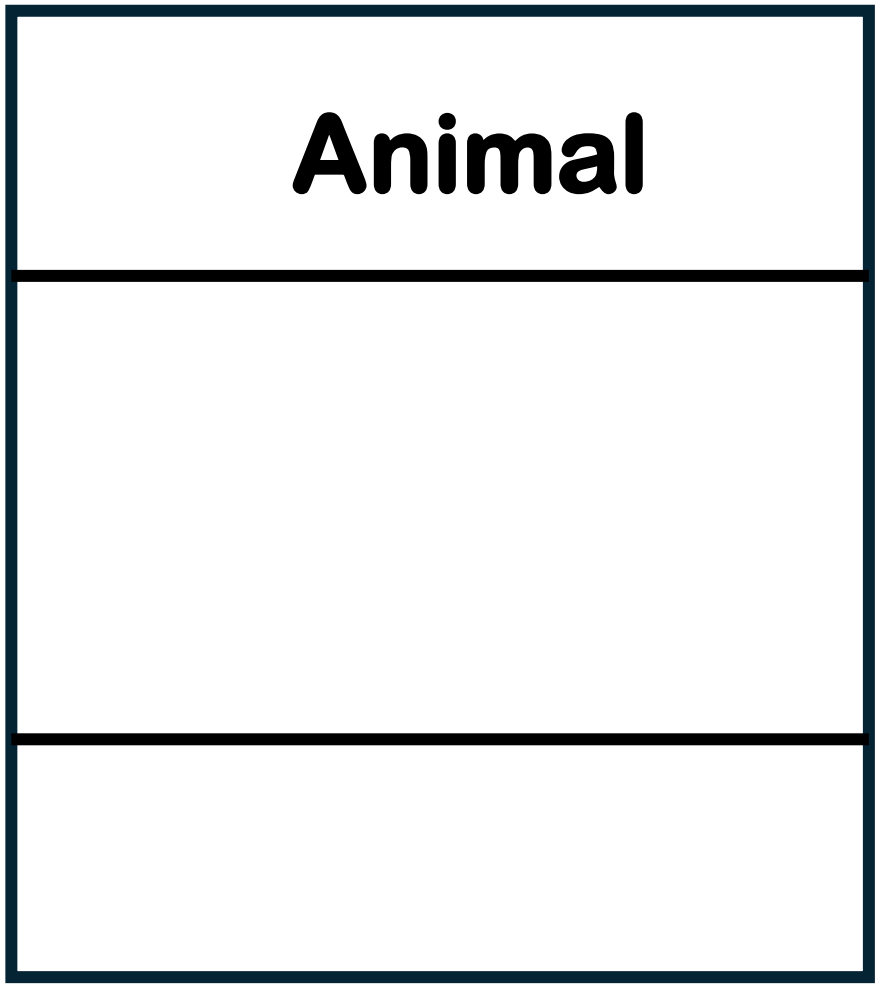
Class

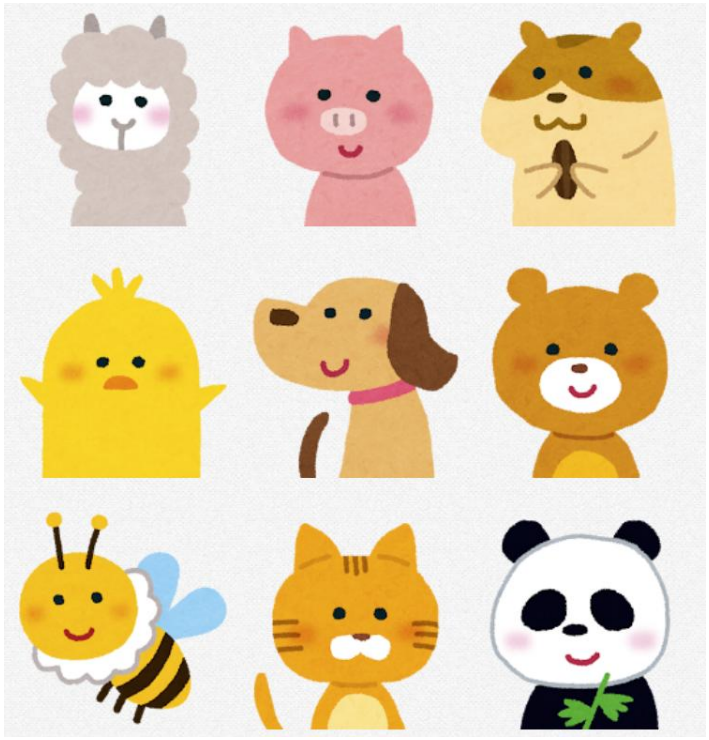
UML – Class Diagram Example



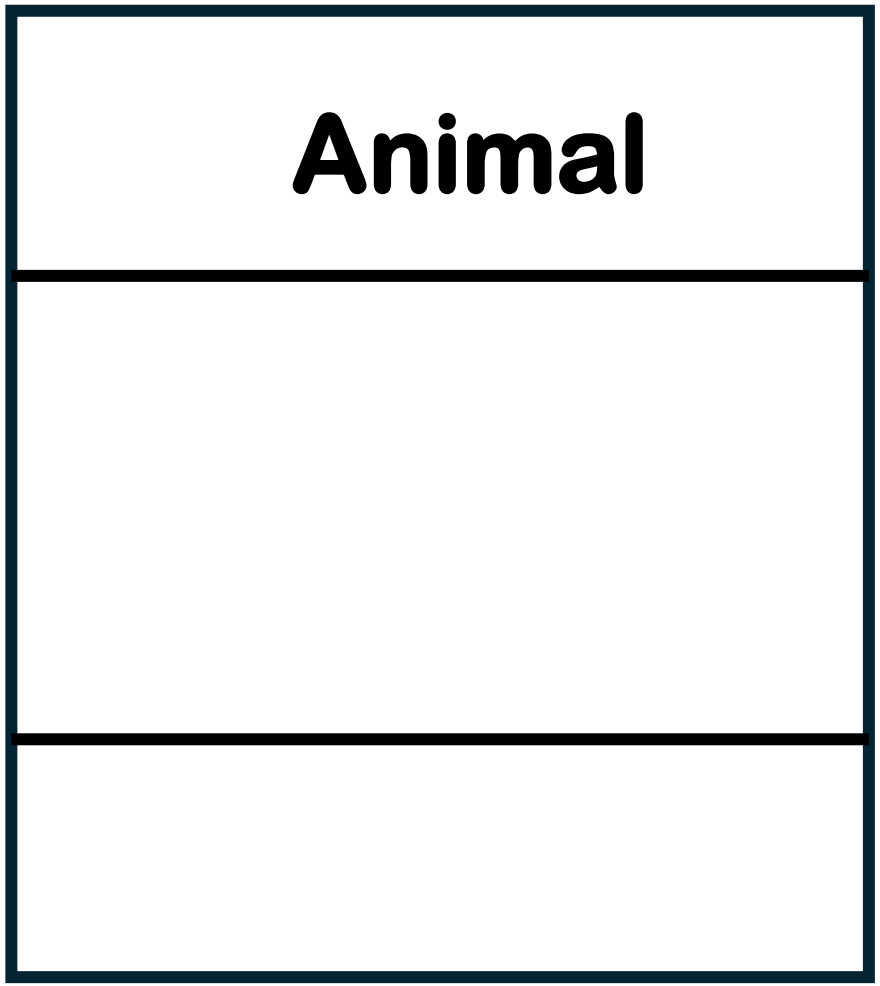


UML – Class Diagram Example

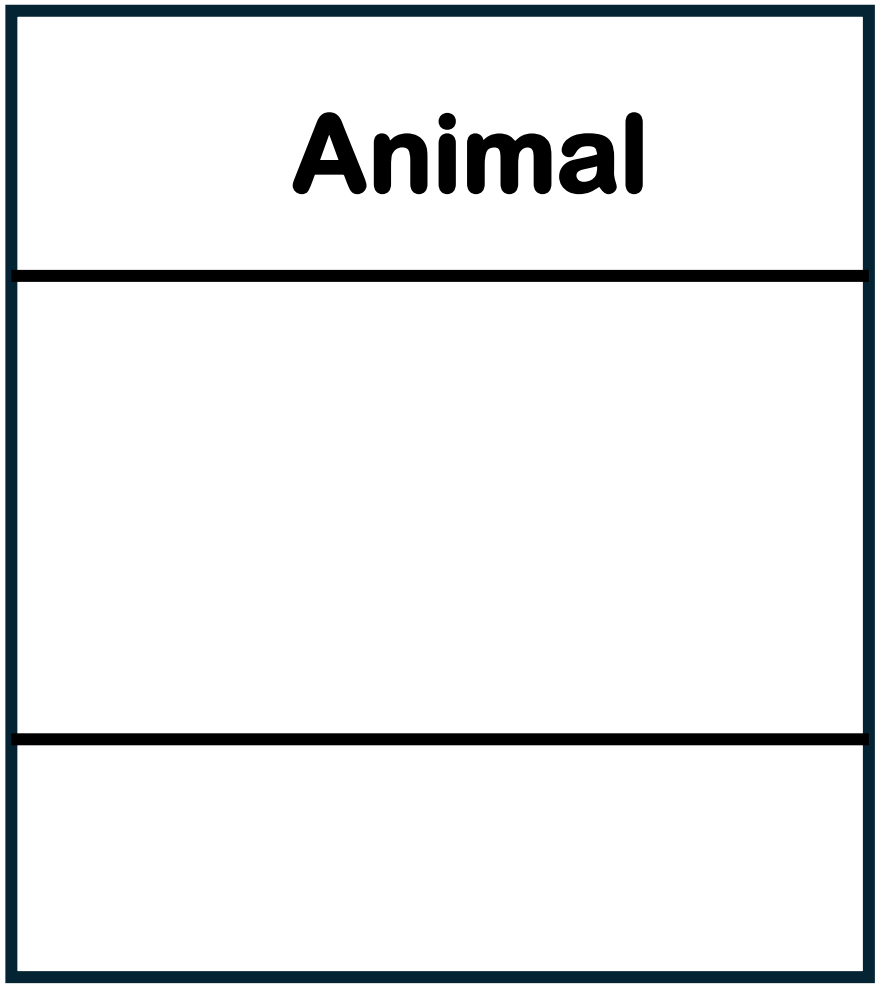




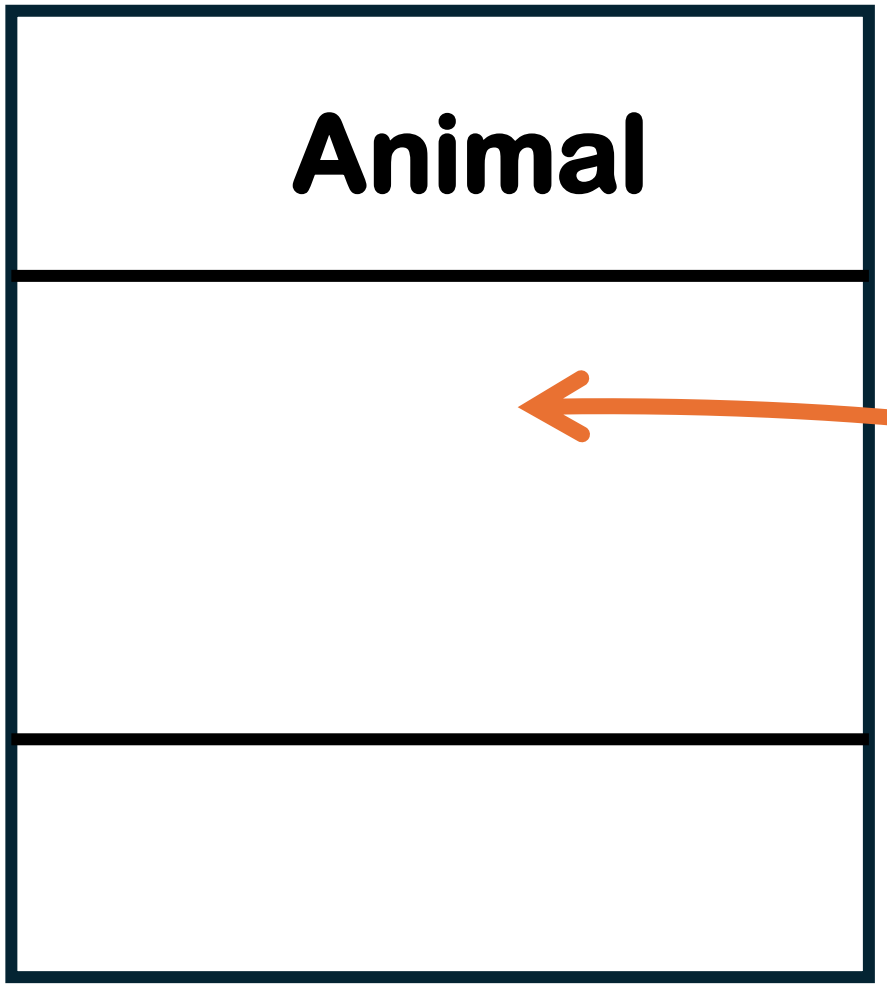
UML – Class Diagram Example



UML – Class Diagram Example



UML – Class Diagram Example



Attributes

A significant piece of data containing values that describe each instance of that class

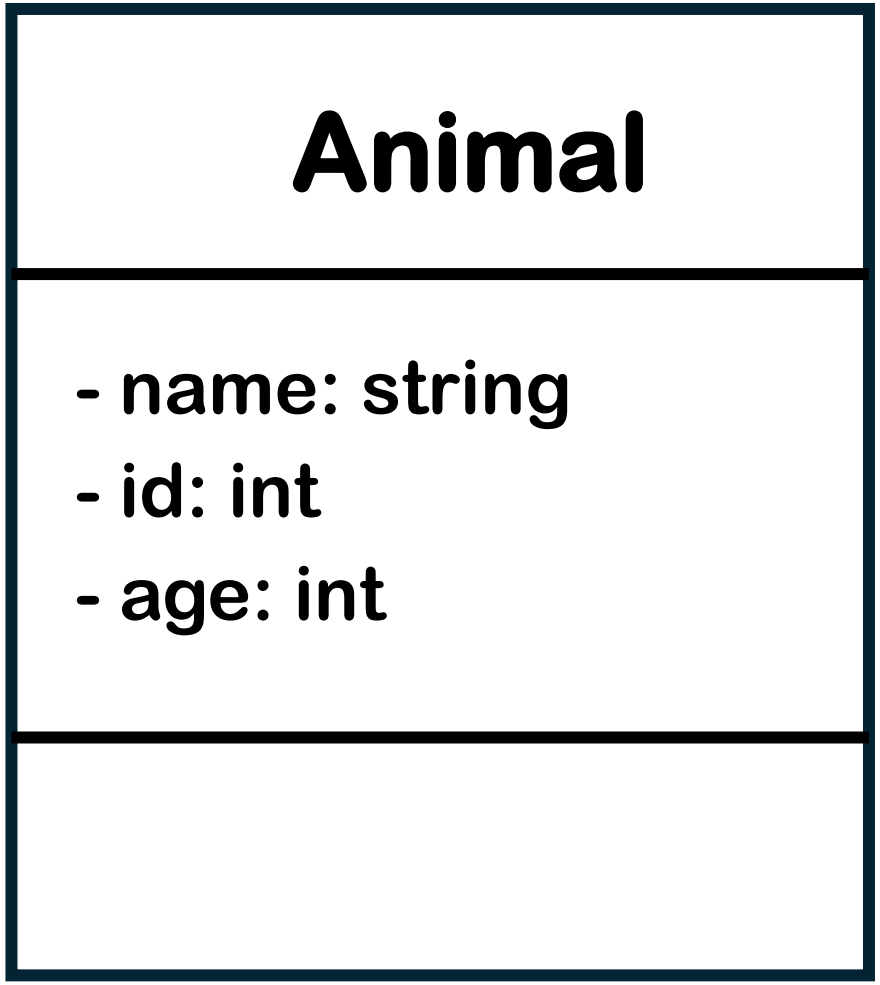
Also known as fields, variables, or properties

UML – Class Diagram Example

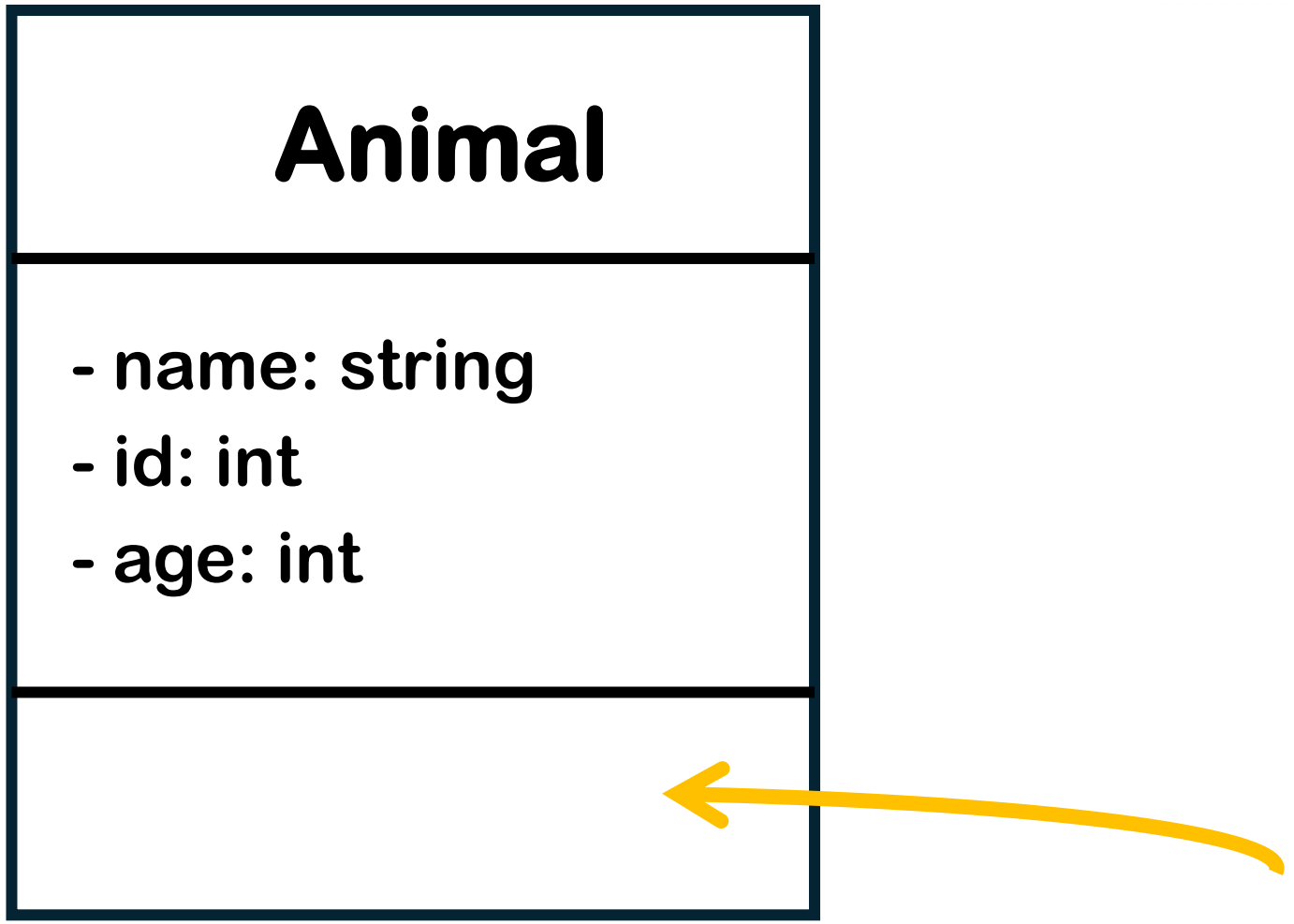
Animal
name id age

name: YuanYuan
id: 99
age: 20

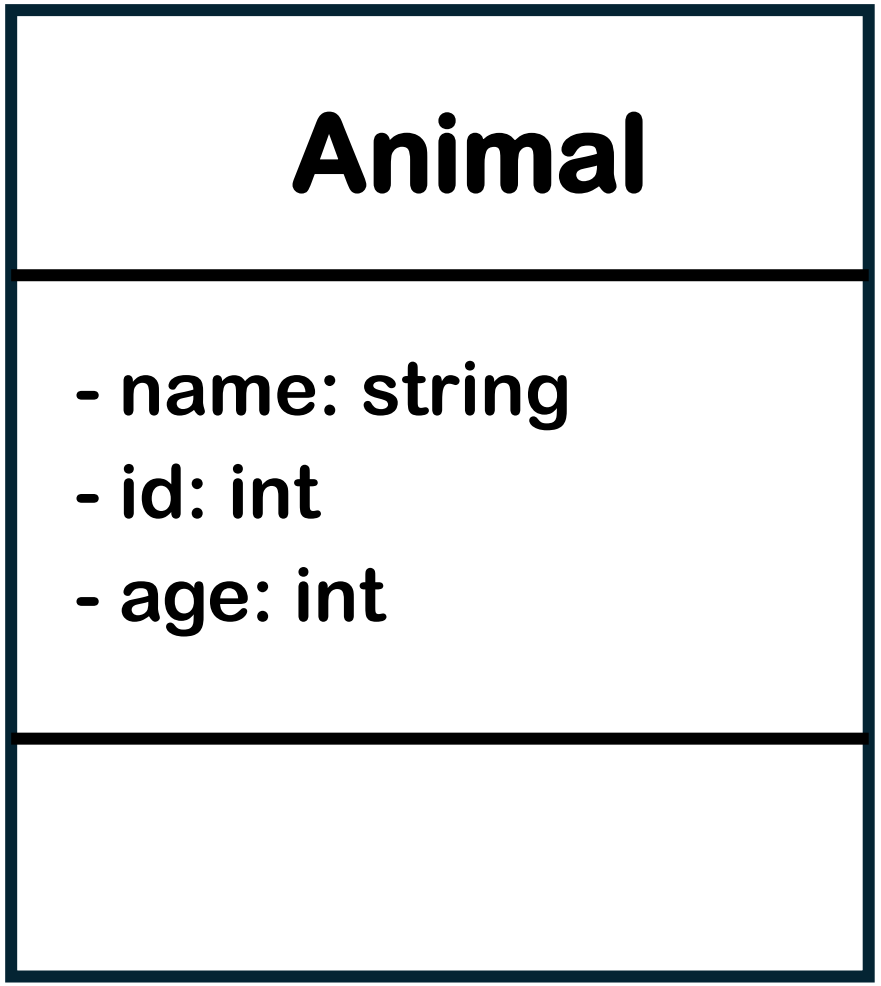
UML – Class Diagram Example



UML – Class Diagram Example



UML – Class Diagram Example



Methods

As knowns as functions or behaviors

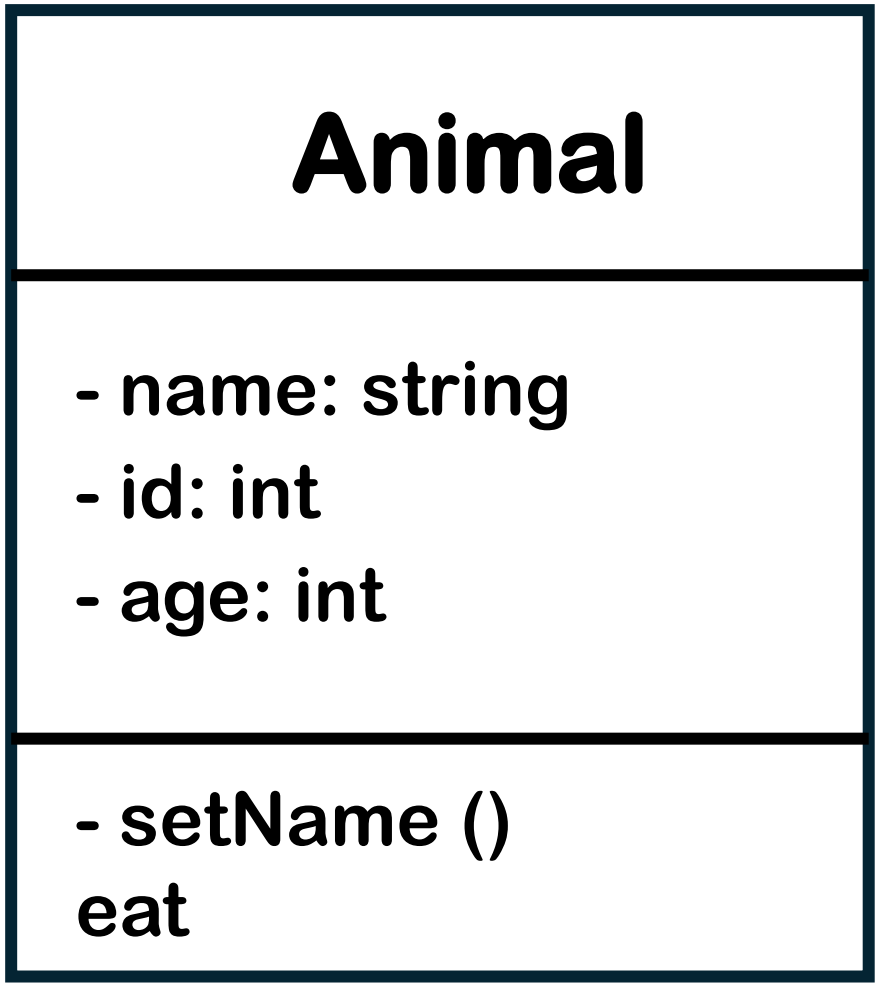
Allow you to specify any behavioral features of a class.

UML – Class Diagram Example

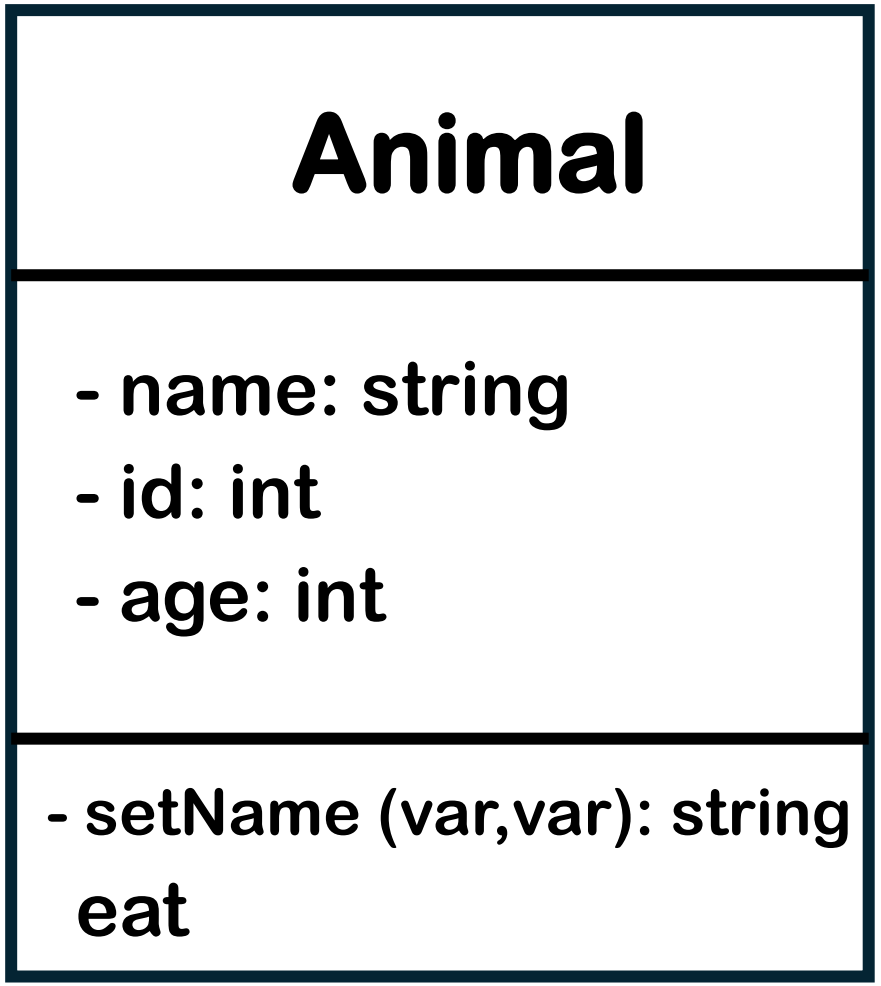
Animal
- name: string - id: int - age: int
setName eat

name: XiangXiang
id: 99
age: 20

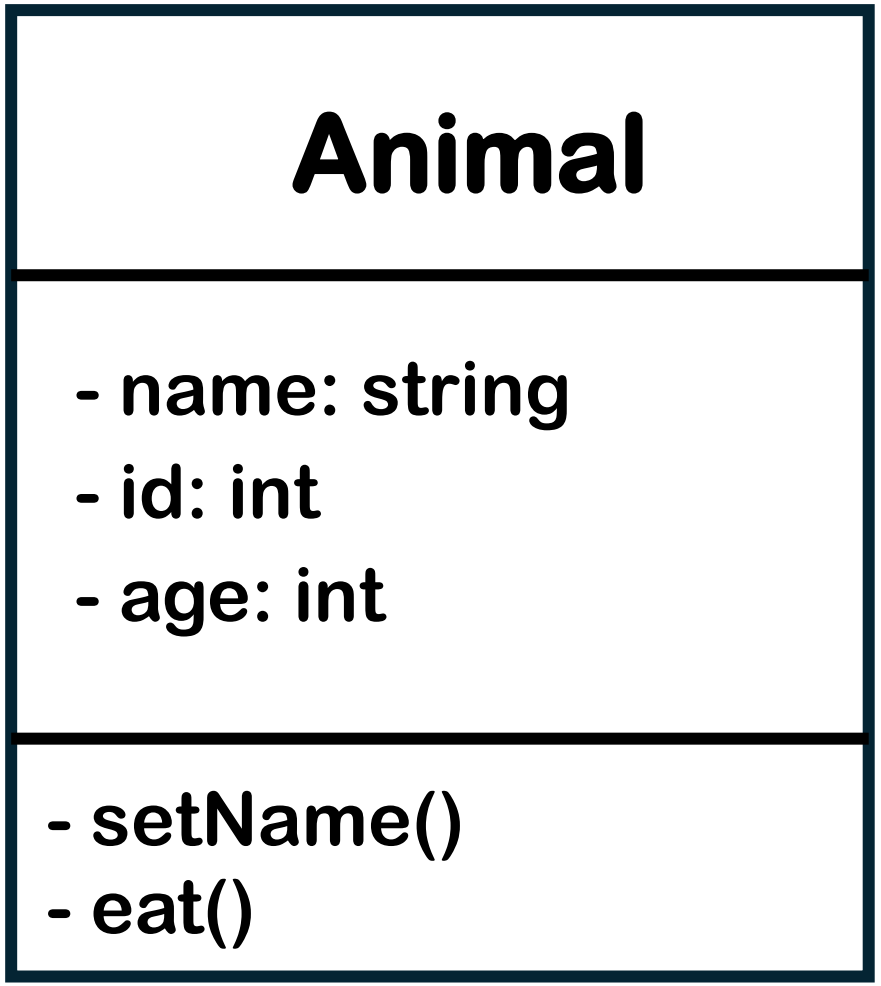
UML – Class Diagram Example



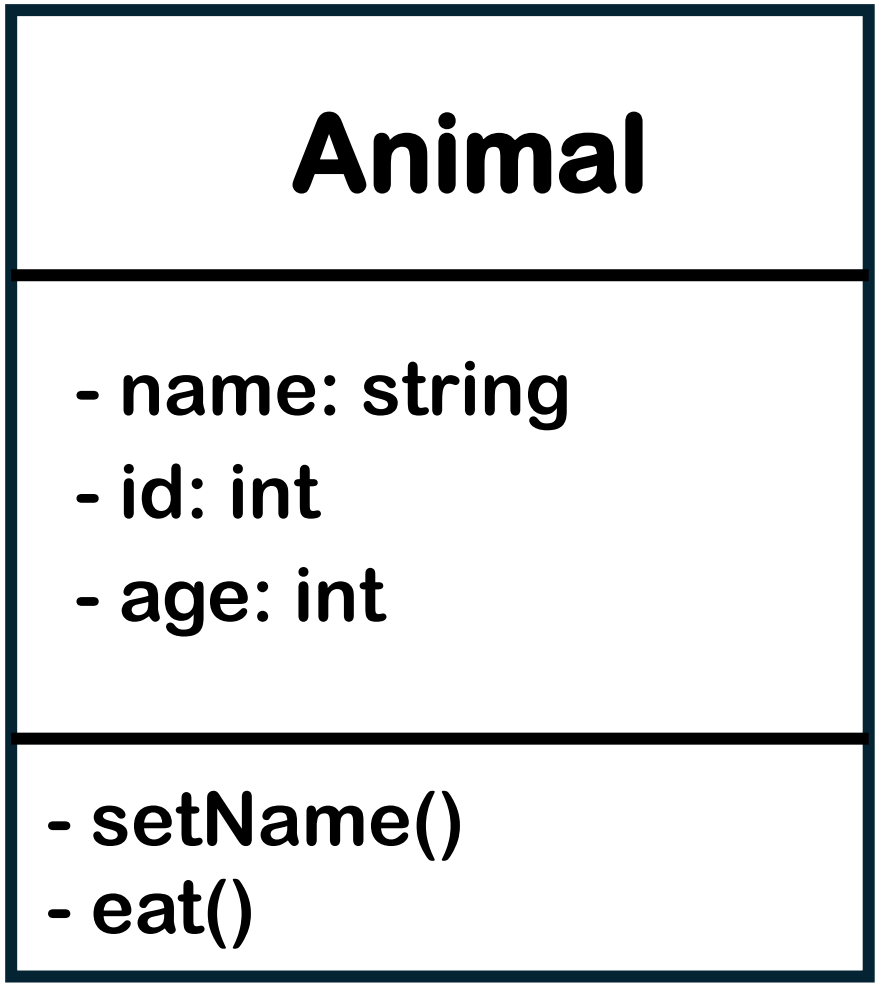
UML – Class Diagram Example



UML – Class Diagram Example



UML – Class Diagram Example

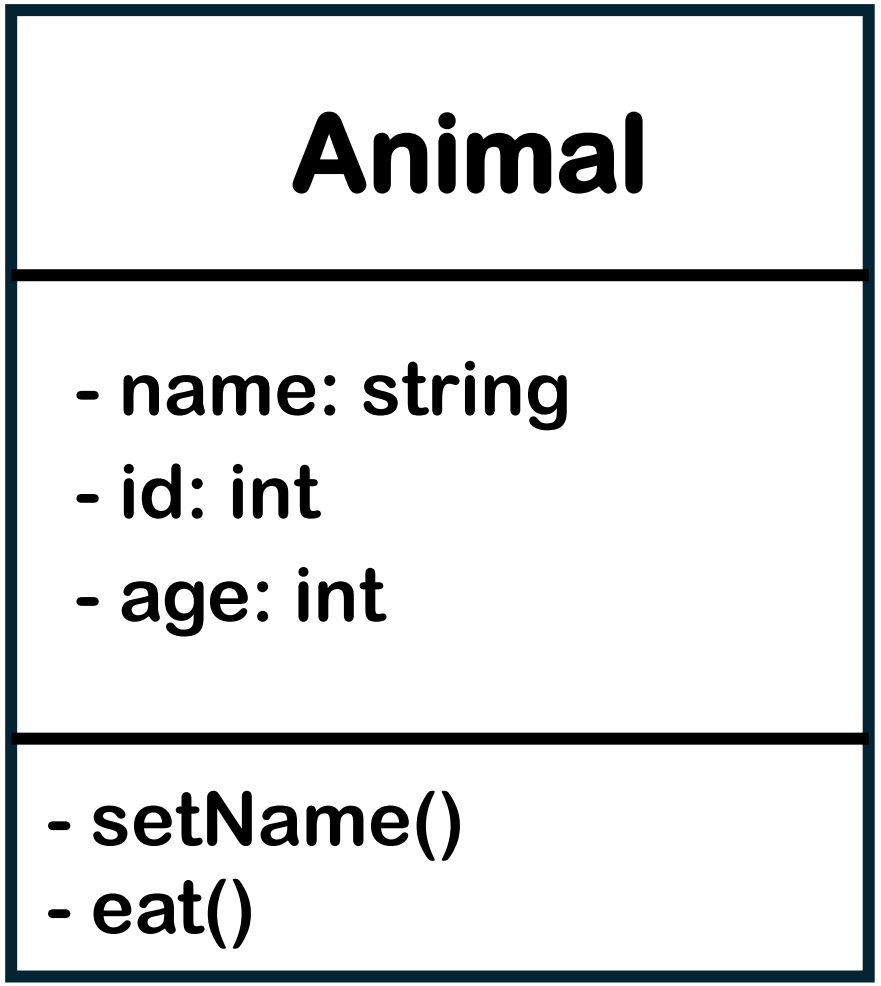


Visibility

Sets the accessibility for that attribute or method

- private
+ public

UML – Class Diagram Example



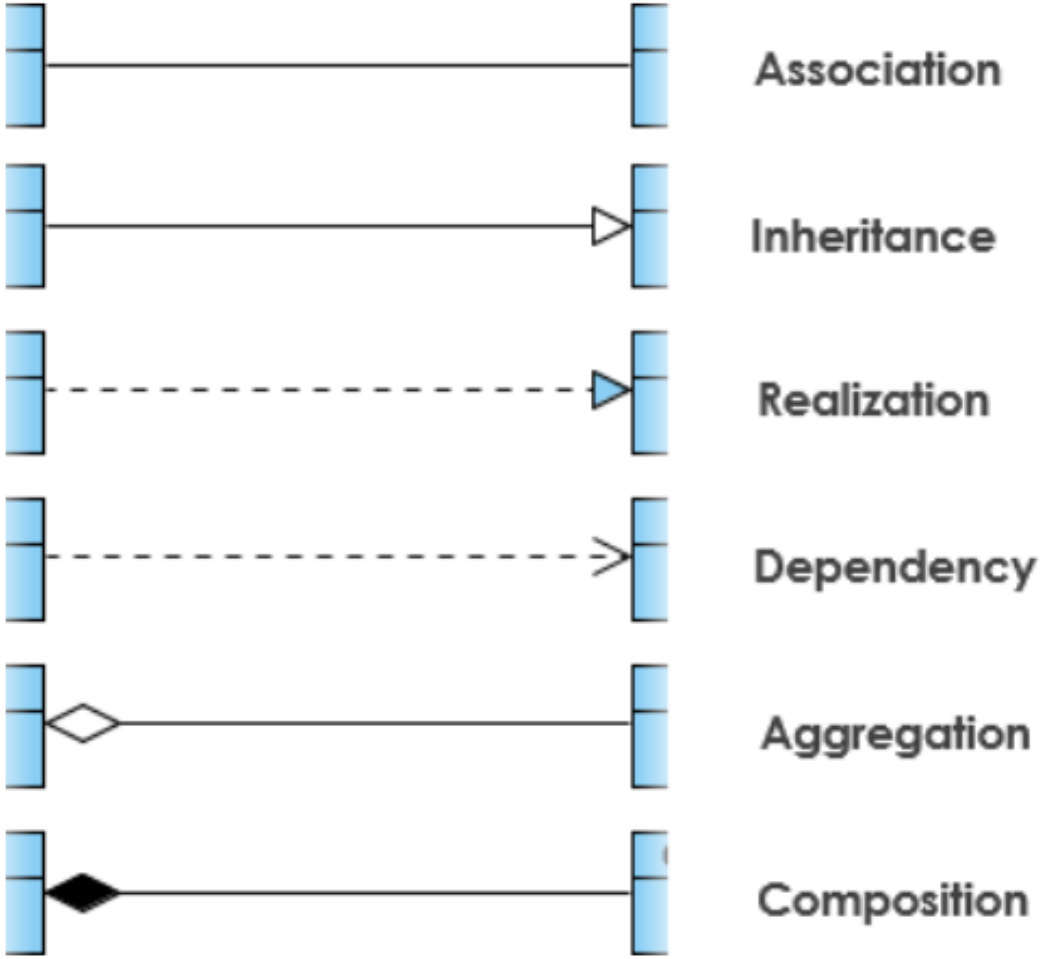
Visibility

Sets the accessibility for that attribute or method

- private
+ public
# protected
~ package/default

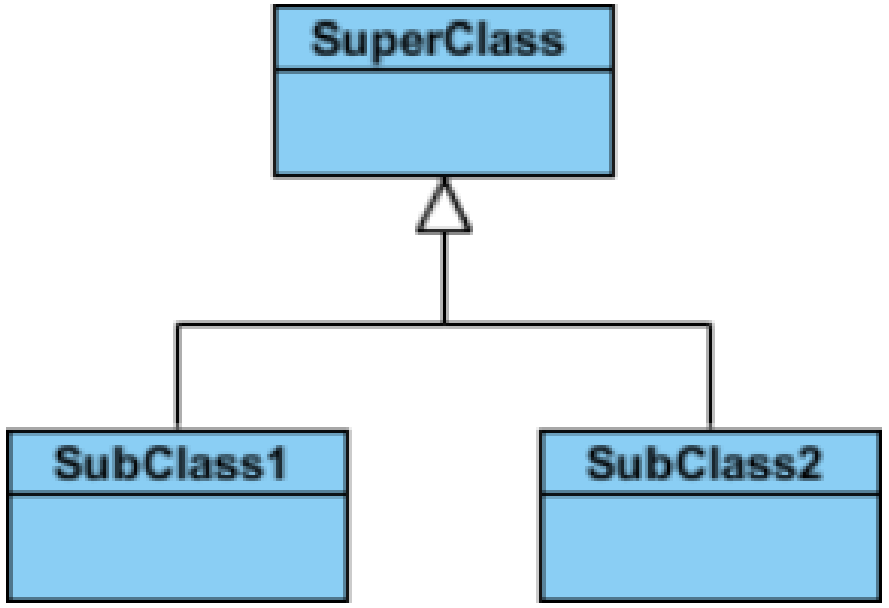
UML – Relationships between classes

A class may be involved in one or more relationships with other classes. A relationship can be one of the following types:



UML –Inheritance Relationship

- The figure to the right shows an example of inheritance hierarchy. SubClass1 and SubClass2 are derived from SuperClass.
- The relationship is displayed as a solid line with a hollow arrowhead that points from the child element to the parent element.



Any Questions?