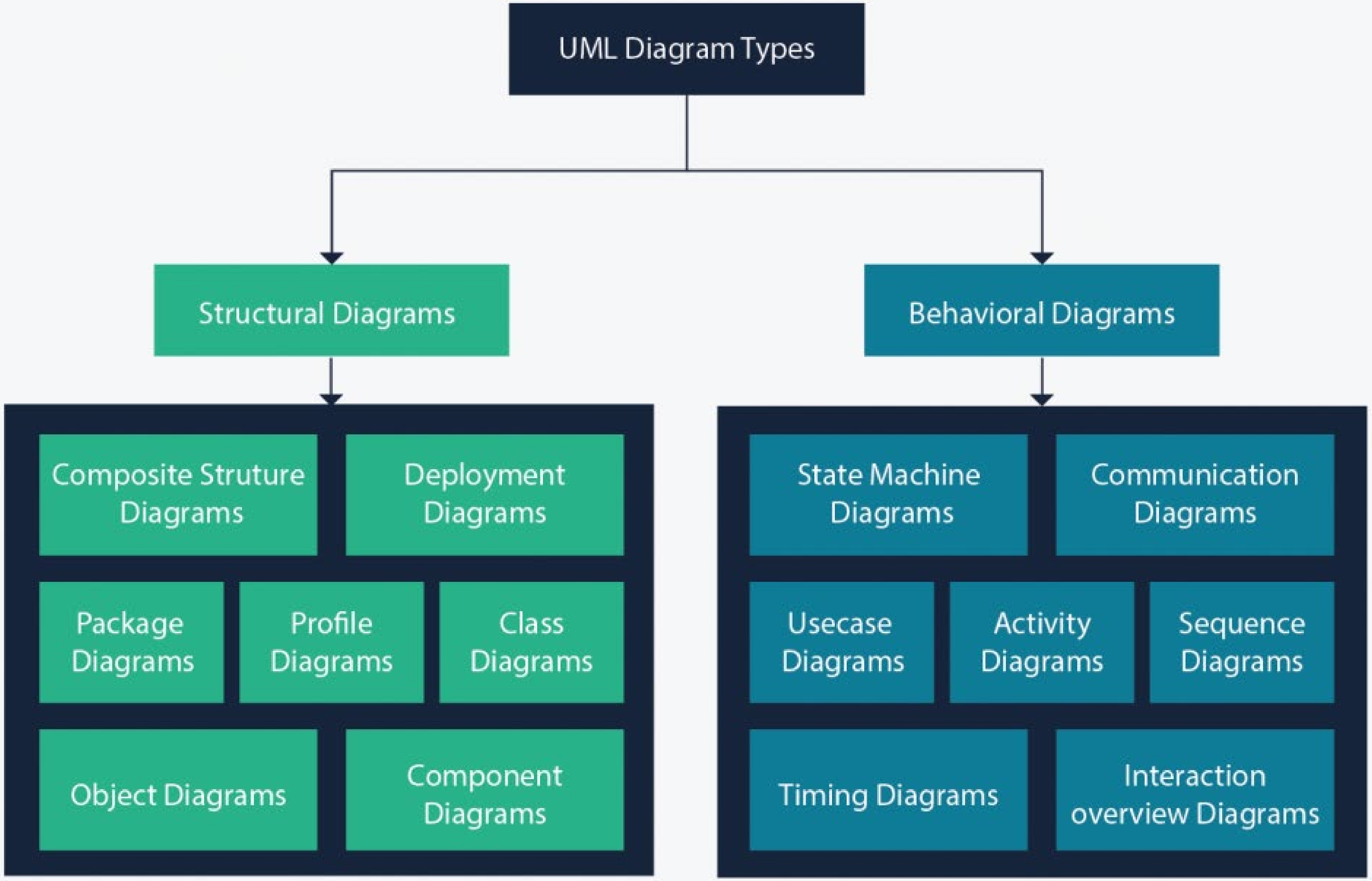
STRUCTURAL MODELING

Majid Askari

SENG 300 Tutorial


UML Diagram Types
Structural Diagrams
Behavioral Diagrams
Composite Struture Diagrams
Deployment Diagrams
Package Diagrams
Profile Diagrams
Class Diagrams
Object Diagrams
Component Diagrams
State Machine Diagrams
Communication Diagrams
Usecase Diagrams
Activity Diagrams
Sequence Diagrams
Timing Diagrams
Interaction overview Diagrams

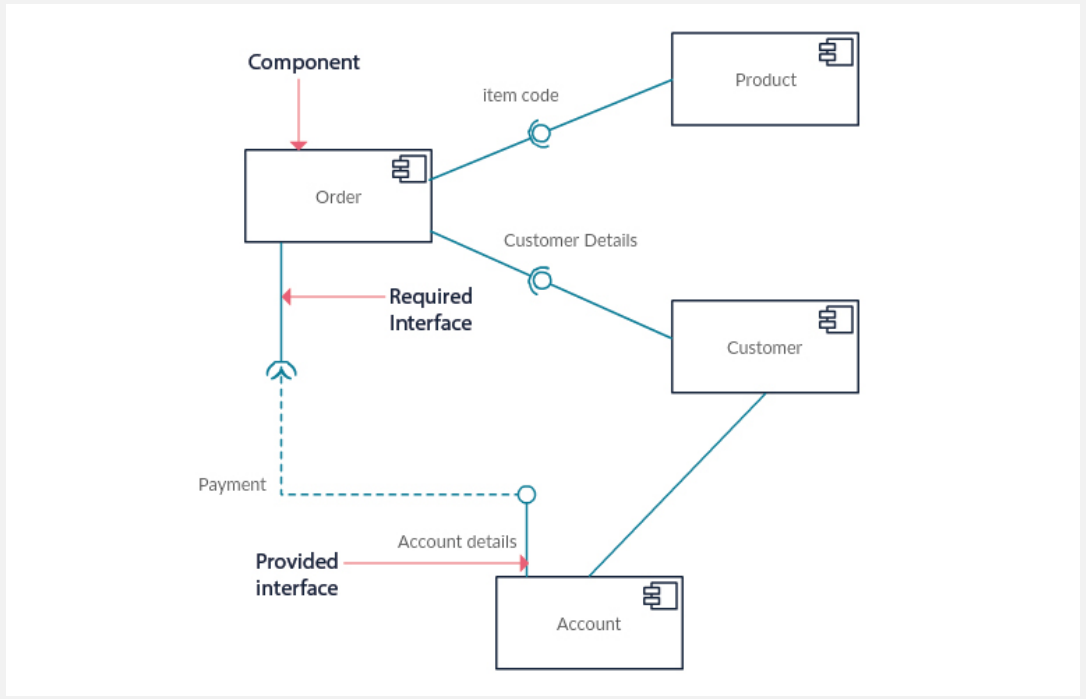
*I

CLASS DIAGRAM

- Mainstay of object-oriented analysis and design
- Defines type of classes and their static relationships including:
    - Association
    - Inheritance
    - Aggregation

COMPONENT DIAGRAM

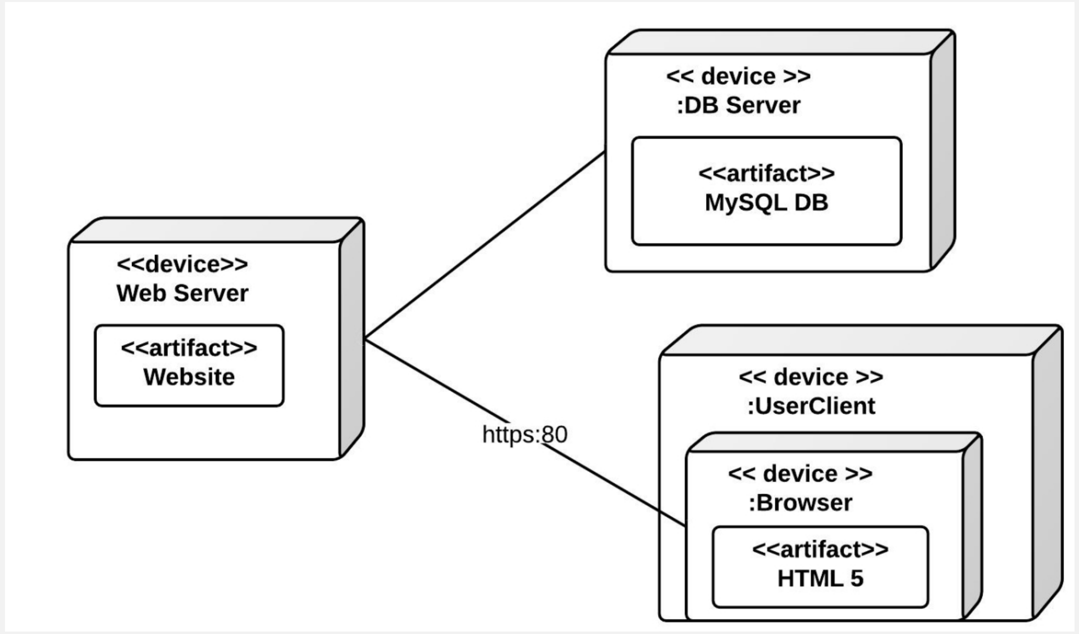
- How components are put together to form larger systems
- It shows architecture of components and dependencies between them
- Includes:
  - Run-time Components
  - Executable Components
  - Source Code Components



*I

DEPLOYMENT DIAGRAM

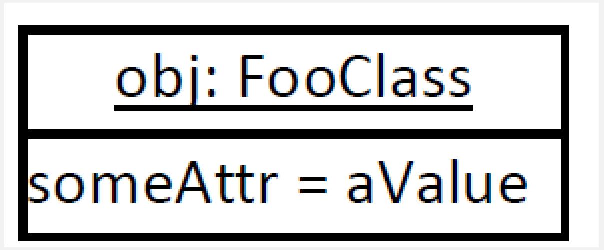
- They show the structure of the run-time system
- Model physical hardware that will be used to implement the system and communication between them

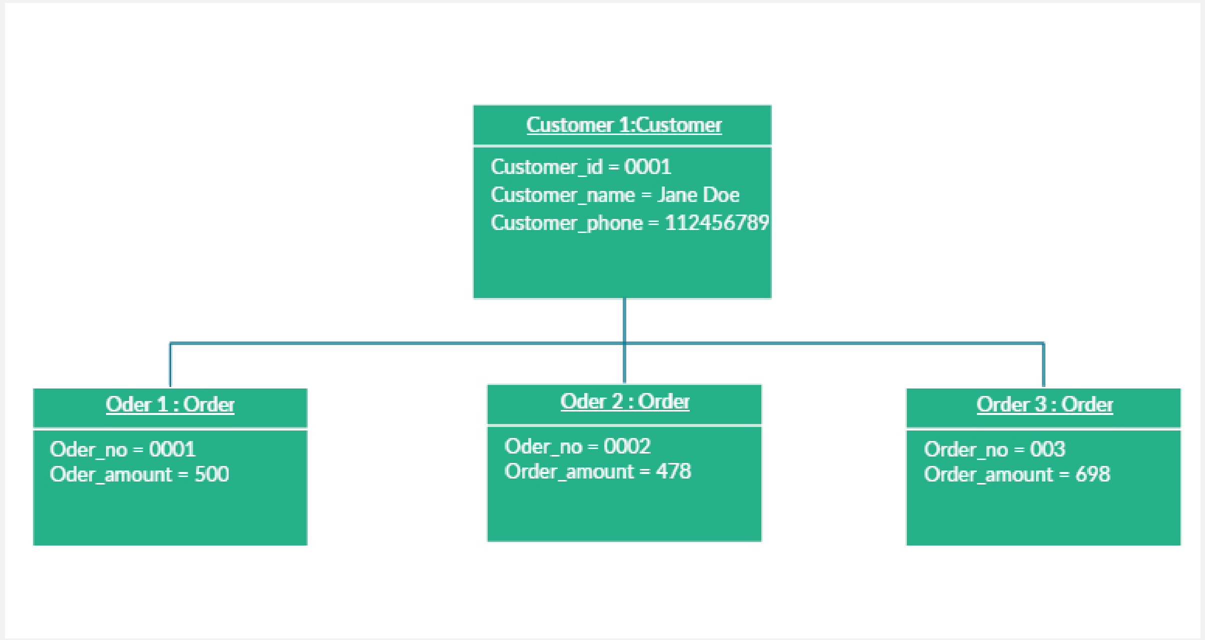


*2

OBJECT DIAGRAM

- Instance of a class diagram
- Shows the objects in the system at a particular time
- Shows the relationship between objects
- Syntax is <name>:<type>
- If there is no underlining: role

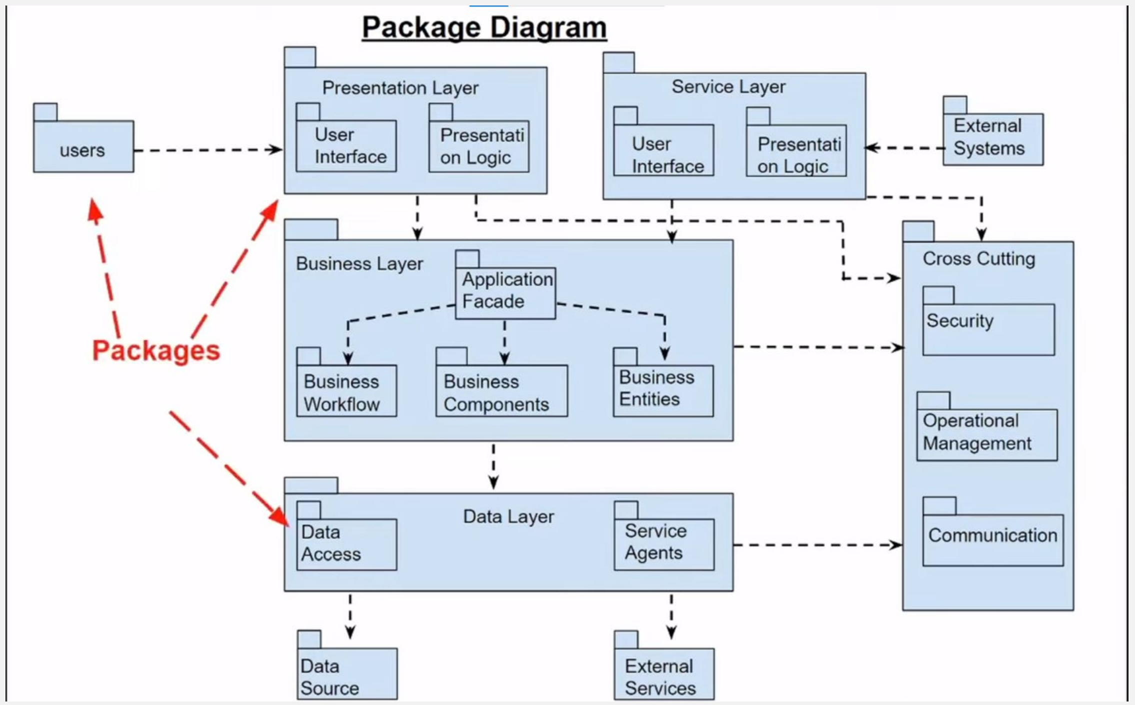




*I

PACKAGE DIAGRAM

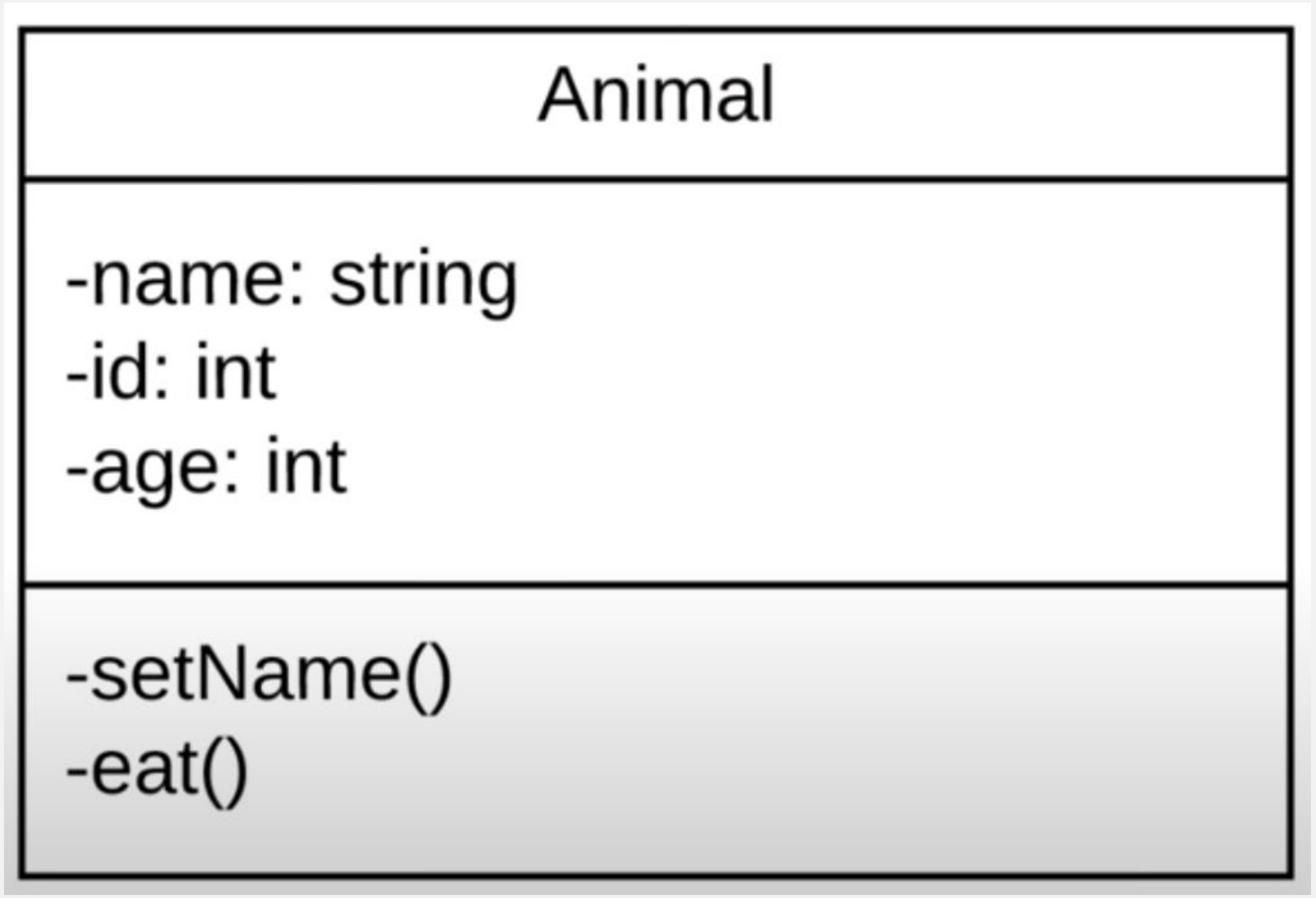
- Shows packages and dependencies between them



*3

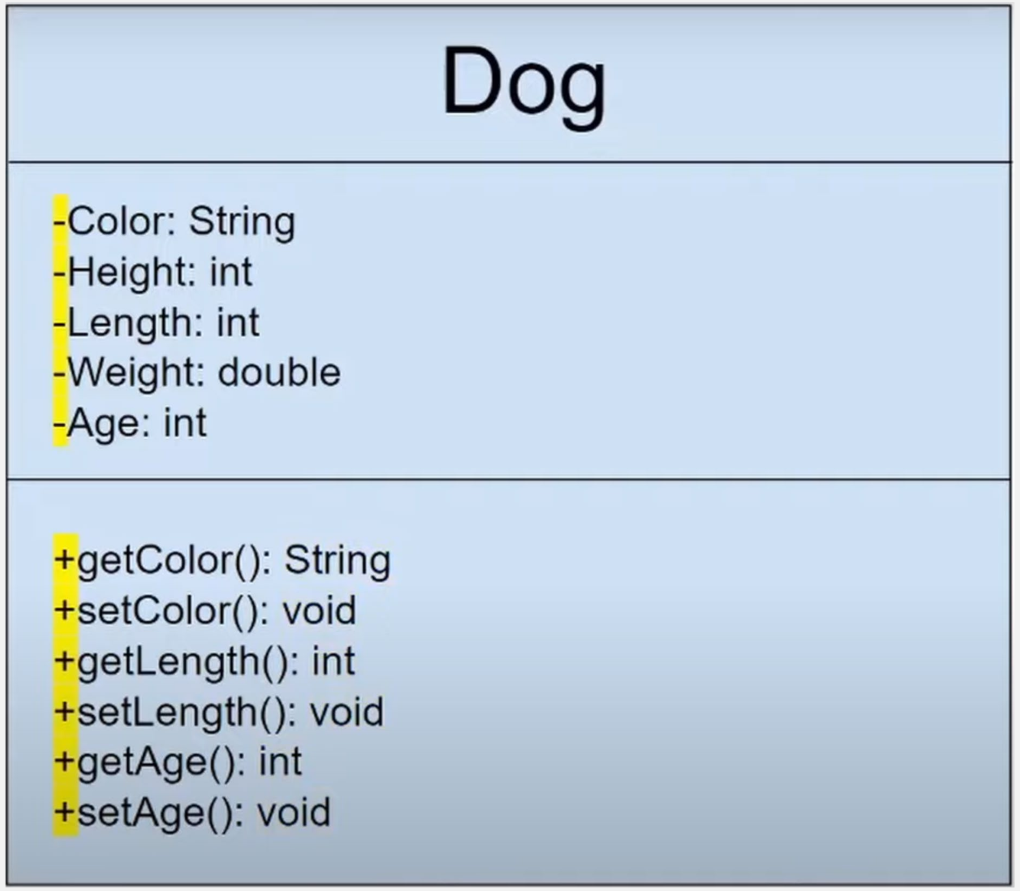
ESSENTIAL ELEMENTS OF A CLASS DIAGRAM

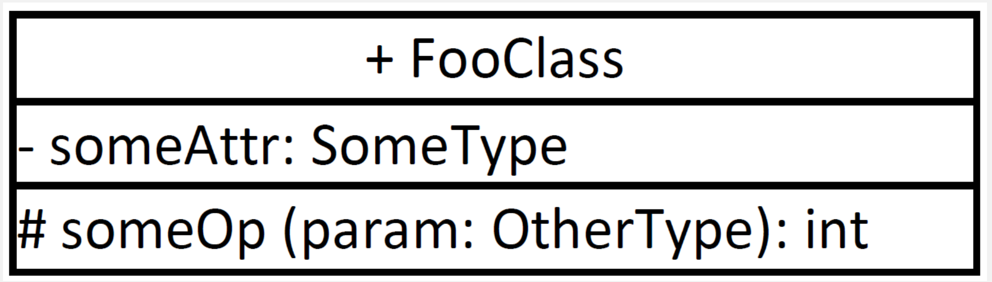
- Class Name
- Attributes
- Operations



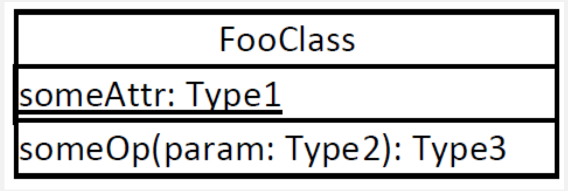
DETAILS OF CLASS IN CLASS DIAGRAM

- Name of attributes, methods and their parameters
- Type of attributes, params and return type
- Access modifiers

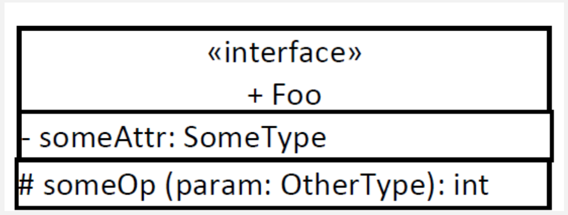




DETAILS OF CLASS IN CLASS DIAGRAM

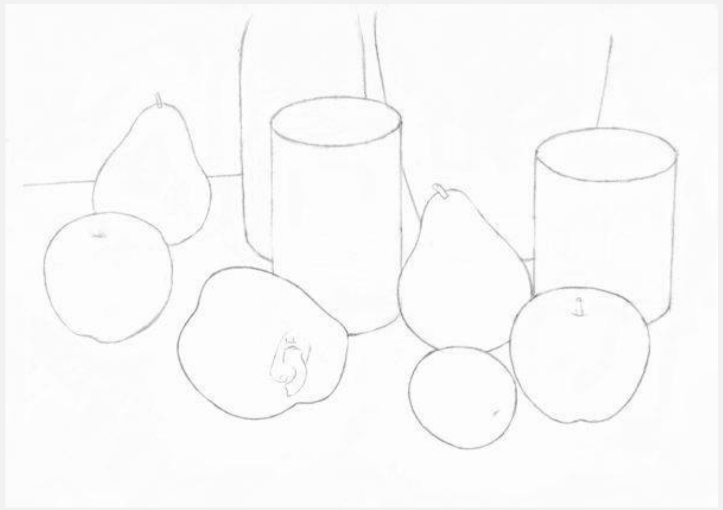
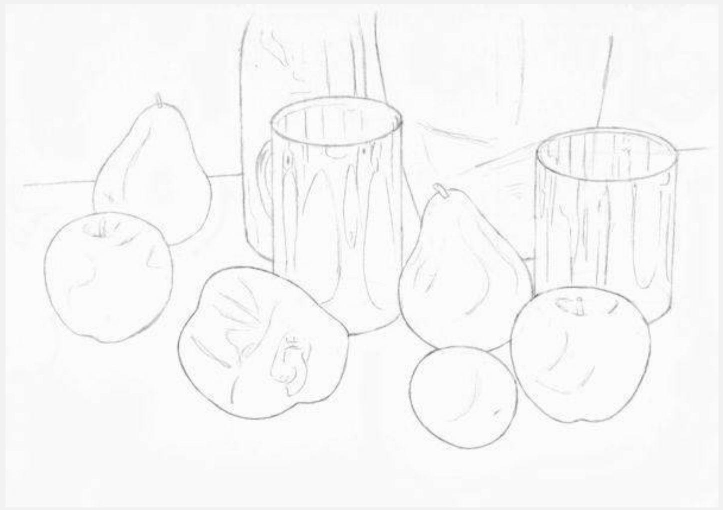
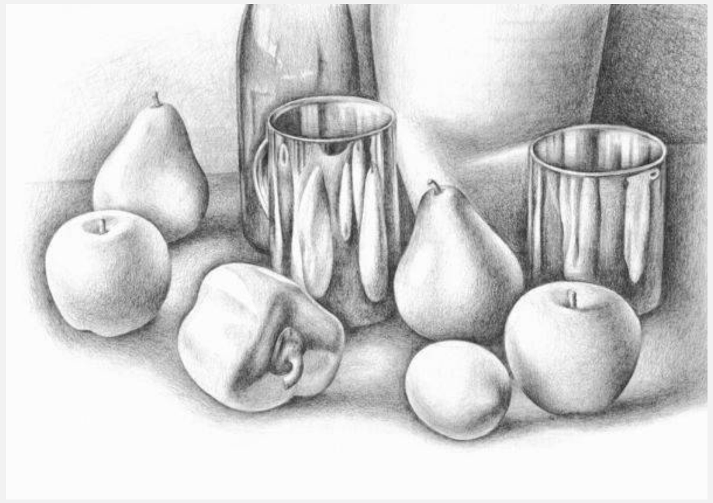


- Static methods and attributes can be shown by underlining
- Interfaces have the same syntax of a class but with <<interface>> keyword
- Abstract class names are written in *Italic*
- Access modifiers



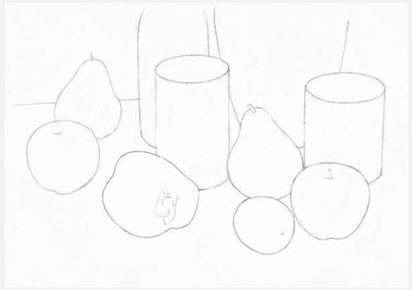
LEVELS OF SPECIFICATION

- 1. Conceptual Perspective
- 2. Specification Perspective
- 3. Implementation Perspective

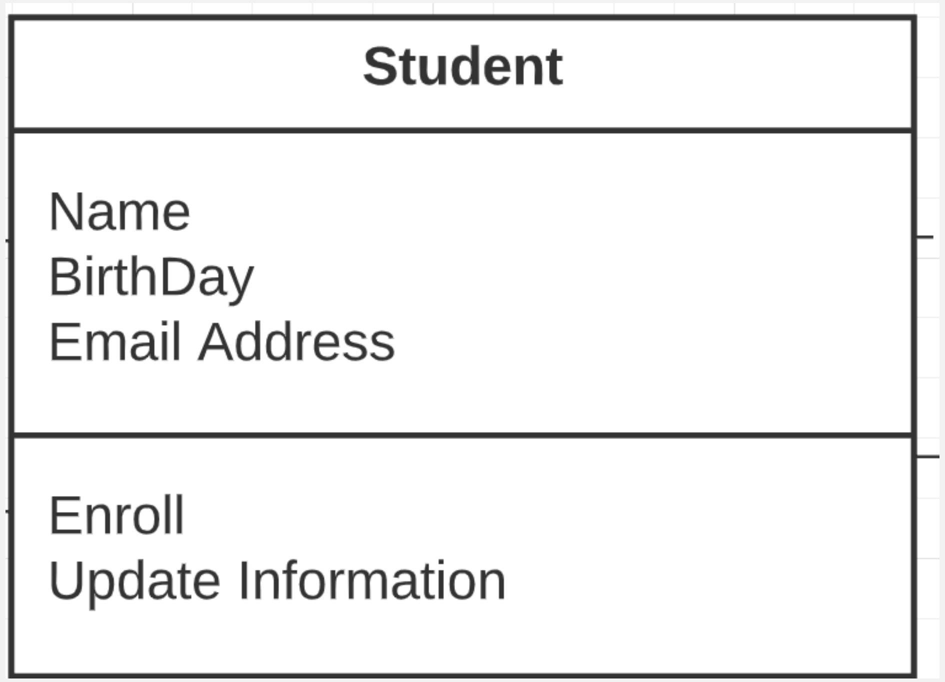




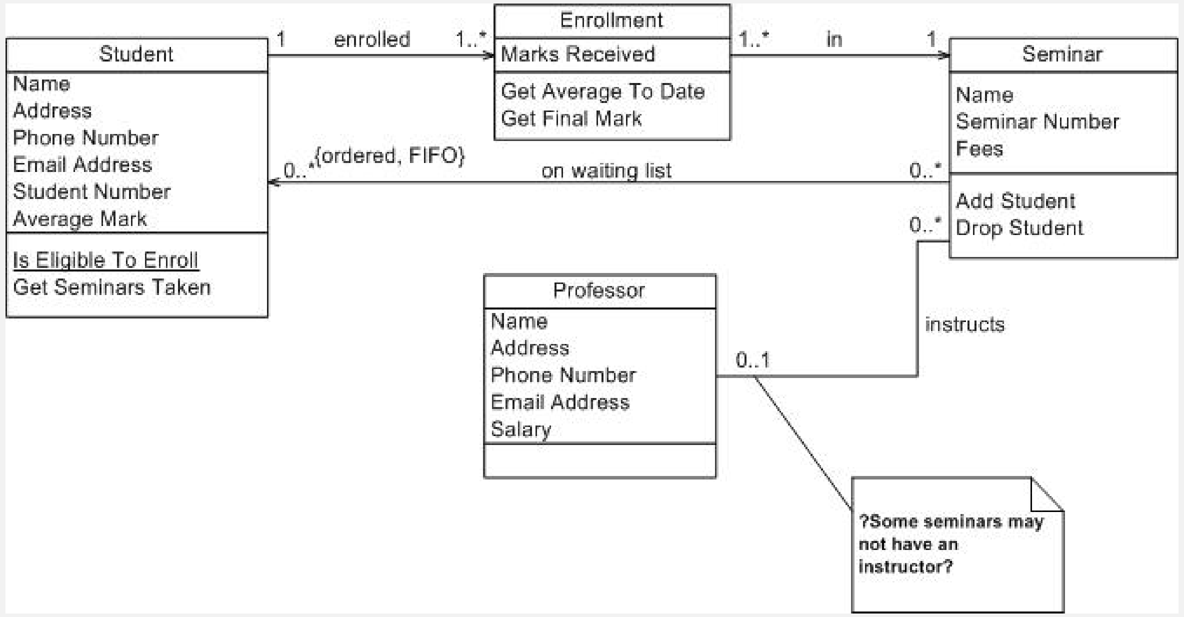
*5

CONCEPTUAL PERSPECTIVE



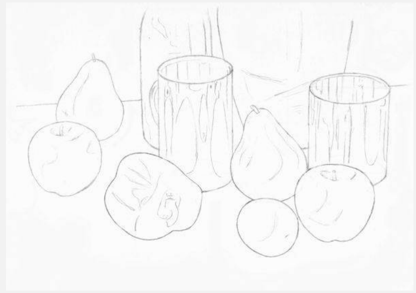
- Concepts of domain
- Language Independent



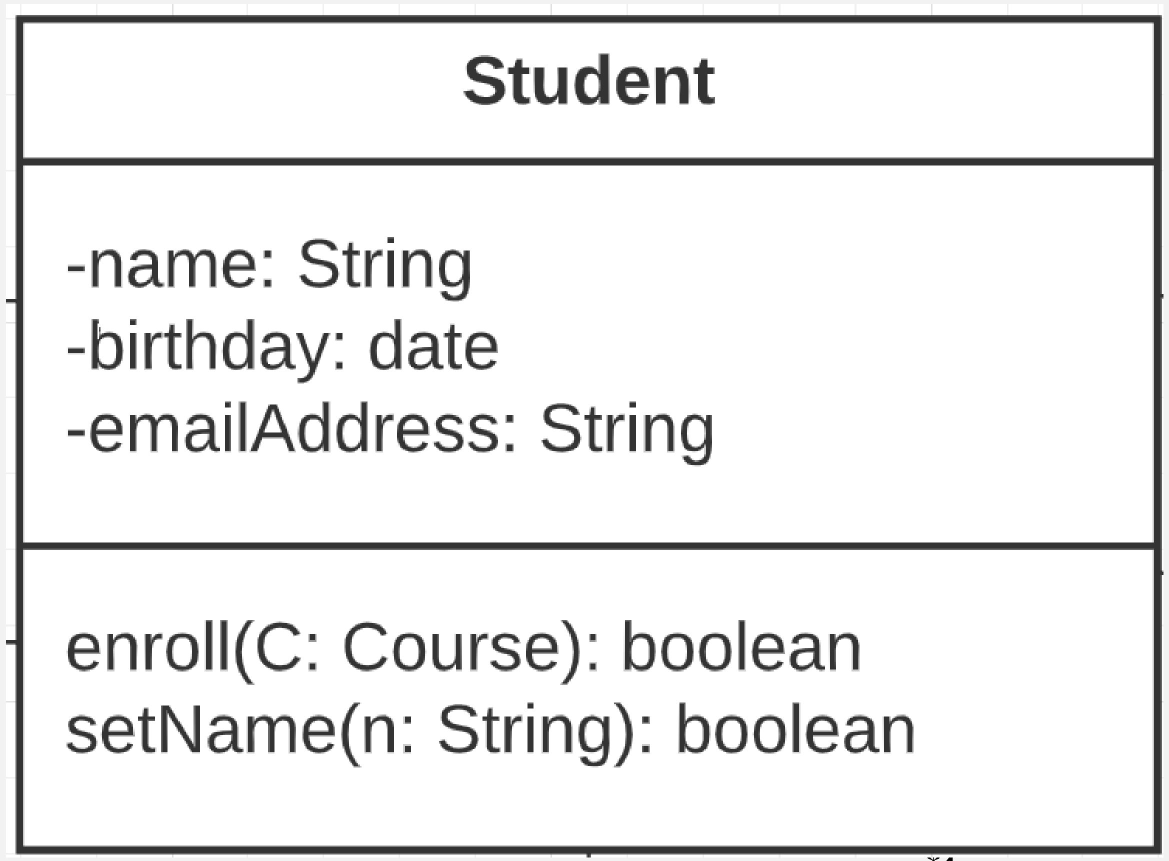


*4

SPECIFICATION PERSPECTIVE

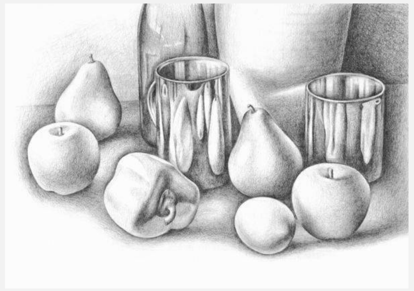


- More Detail
- Describing abstraction of software
- No reference to implementation

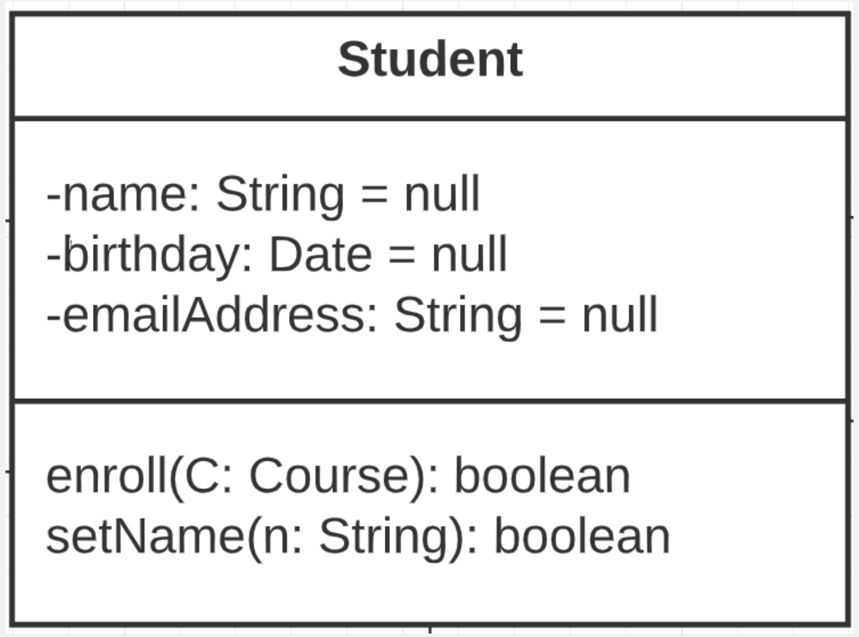


*4

IMPLEMENTATION PERSPECTIVE



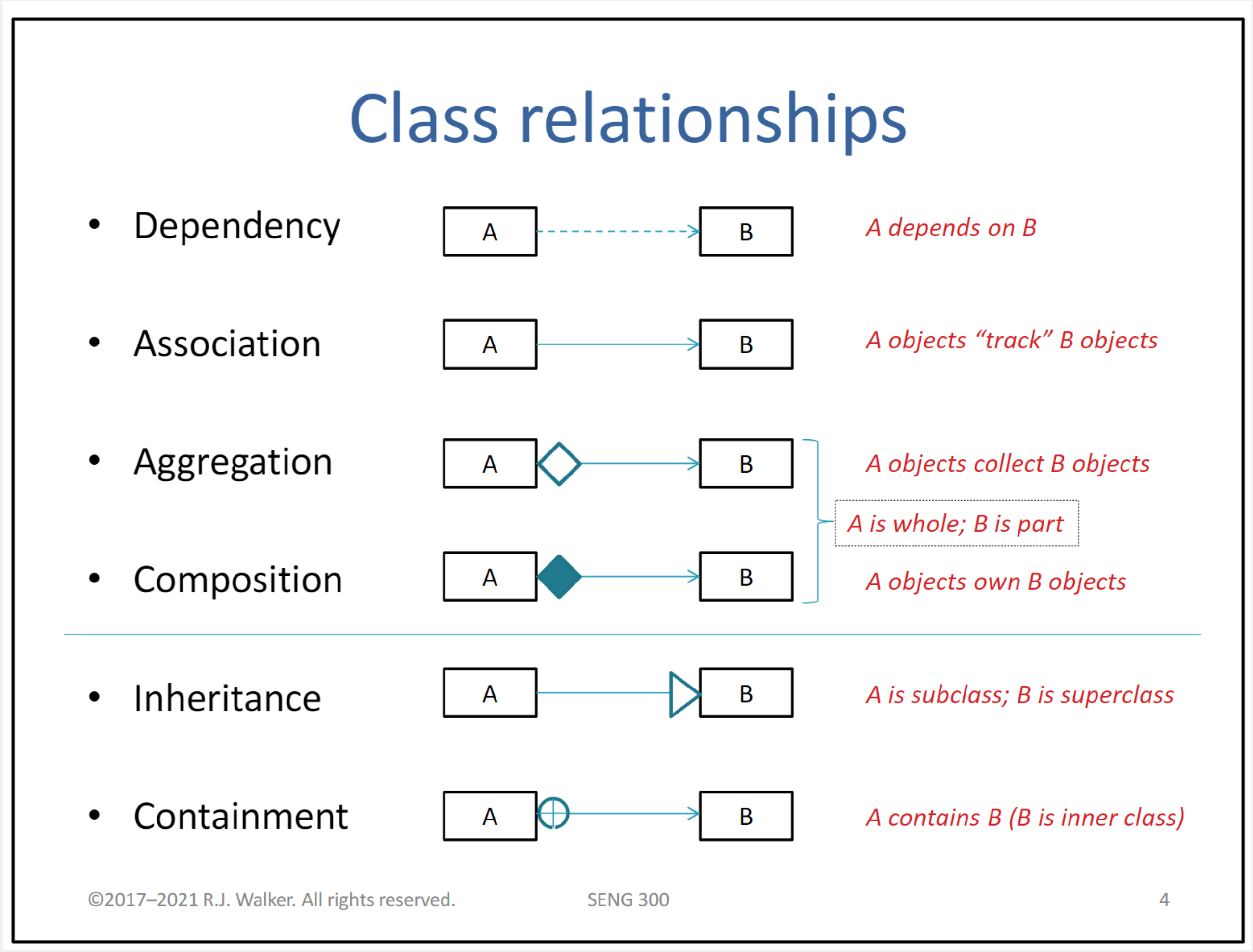
- Description of software implementation in a particular technology (e.g Java or C++)



```
Student
-name: String = null
-birthday: Date = null
-emailAddress: String = null
enroll(C: Course): boolean
setName(n: String): boolean
```

*4

RELATIONS IN CLASS DIAGRAM



Class relationships

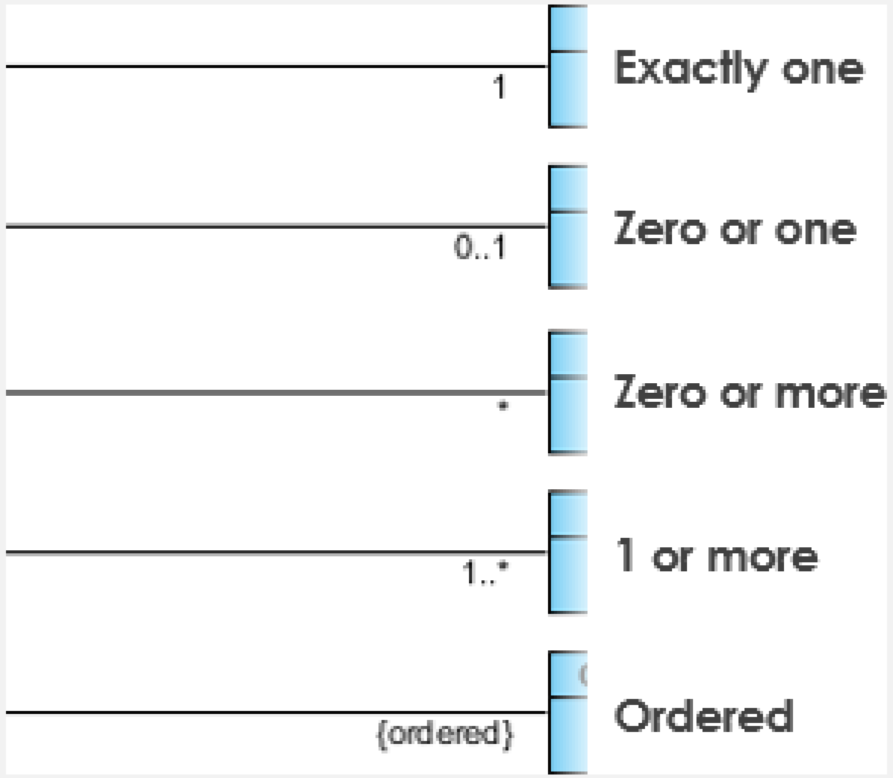
- Dependency
- Association
- Aggregation
- Composition
- Inheritance
- Containment

A
B
A depends on B
A objects "track" B objects
A objects collect B objects
A is whole; B is part
A objects own B objects
A is subclass; B is superclass
A contains B (B is inner class)

CARDINALITY - MULTIPLICITY

- Number of objects that can take part in the relationship

E.g a class can have multiple students



REFERENCES

*1. https://creately.com/blog/diagrams/uml-diagram-types-examples

*2. https://www.lucidchart.com/pages/uml-deployment-diagram

*3. AVE Coders Course on UML

*4. http://www.agilemodeling.com/artifacts/classDiagram.htm

*5. https://www.artyfactory.com/still-life/still_life_pencil.html

*6. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial