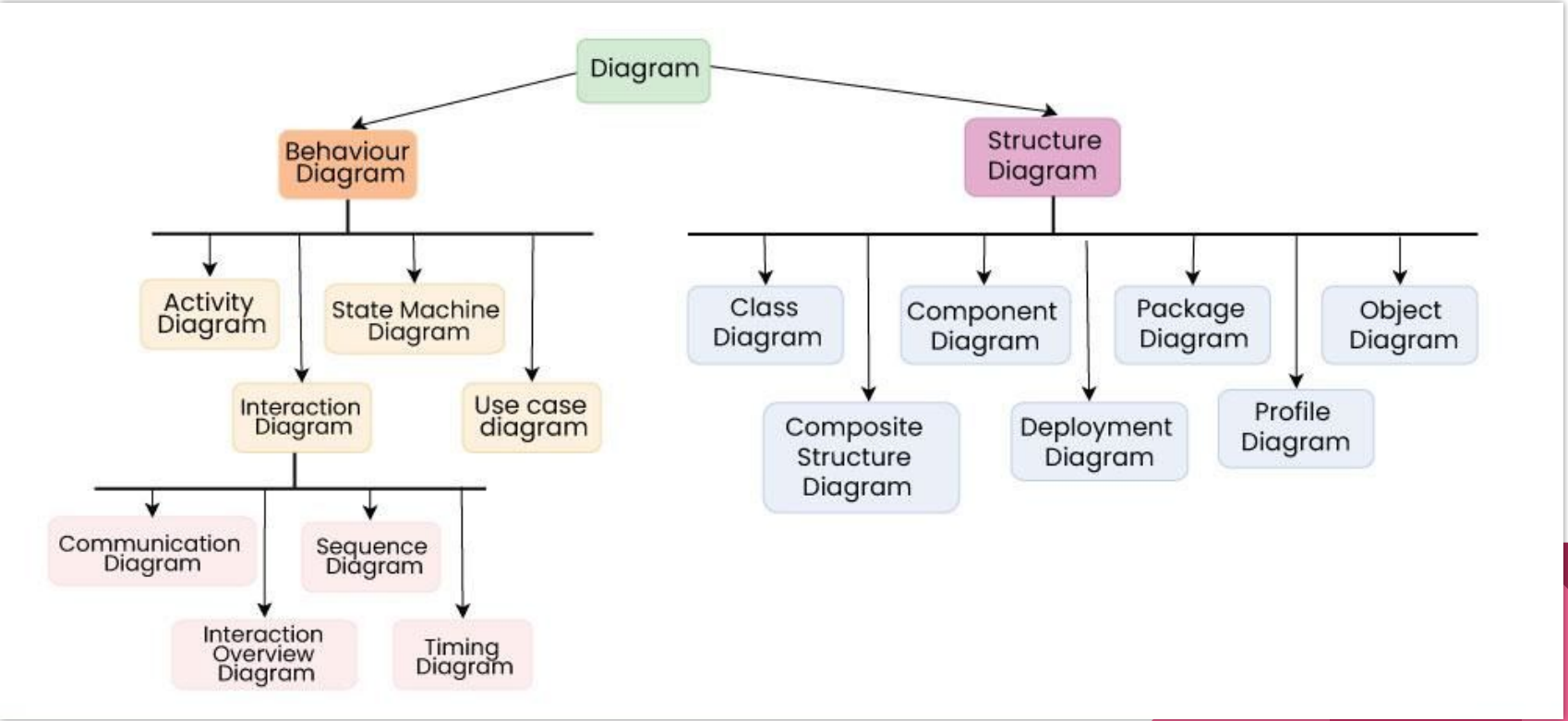
Class Diagrams in UML

SCS 1310 - Object Oriented Programming and Modelling

Isurika Arunodi
isurikaarunodi@gmail.com

Types of UML Diagrams



What is a Class Diagram?

- A class diagram represents the static structure of a system.
- It shows
  - **Classes**
  - **Attributes**
  - **Methods**
  - **Relationships among classes.**
- Fundamental to object-oriented design and software development.

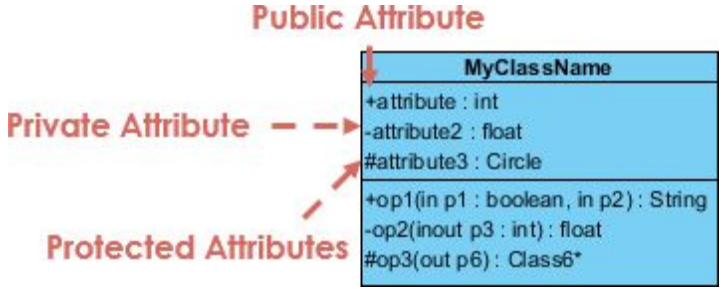
UML Class notation

- **Class name**
  - Write <<interface>> on top of interfaces' names
- **Attributes** (optional)
  - Includes fields of the class
- **Methods** (optional)
  - Includes behaviors of the class

UML Class notation

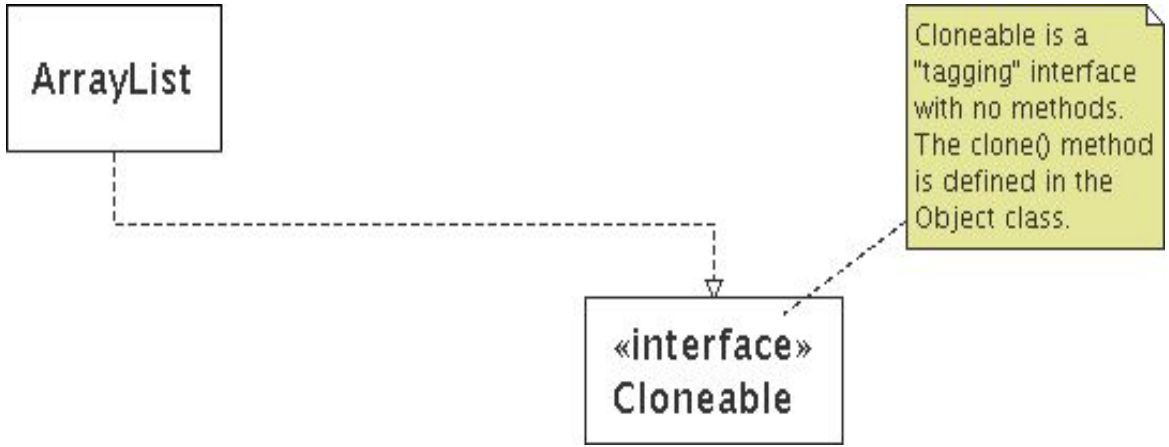
- **Visibility** (fields, instance variables)
  - Symbols before an attribute and operation name in a class.
  - \+ public
    \# protected
    \- private

- **Parameter Directionality**
  - Each parameter in an operation (method) may be denoted as in, out or inout.
  - Specifies its direction with respect to the caller.
  - Directionality is shown before the parameter name.



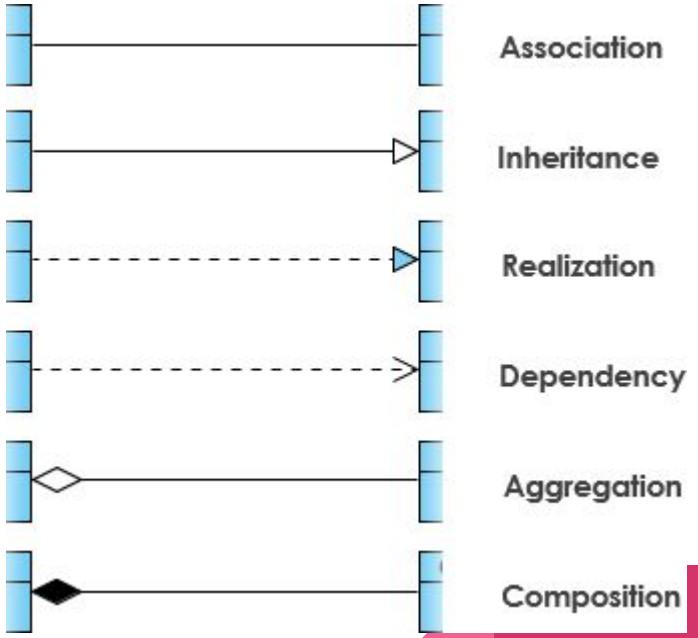
Notes

- Represented as a folded note, attached to the appropriate class.



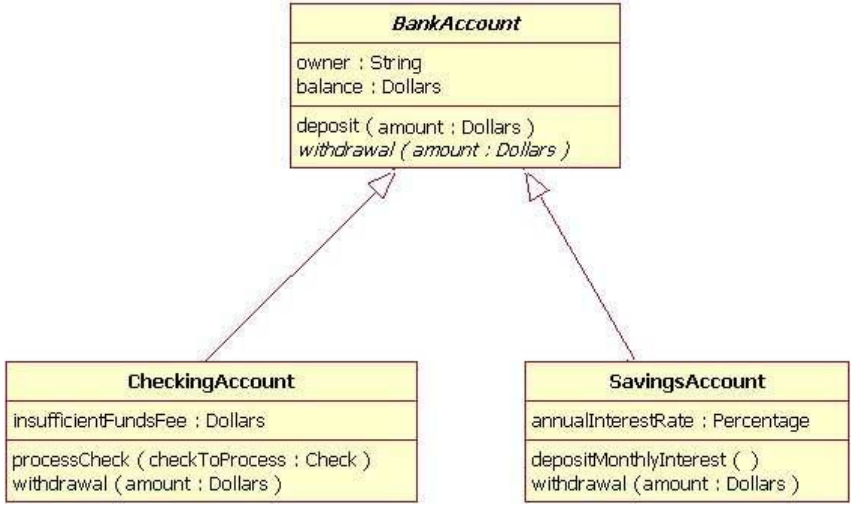
Relationships between classes

- **Generalization**: an inheritance relationship
  - Inheritance between classes
  - Interface implementation
- **Association**: a usage relationship
  - Dependency
  - Aggregation
  - Composition
  - Realization



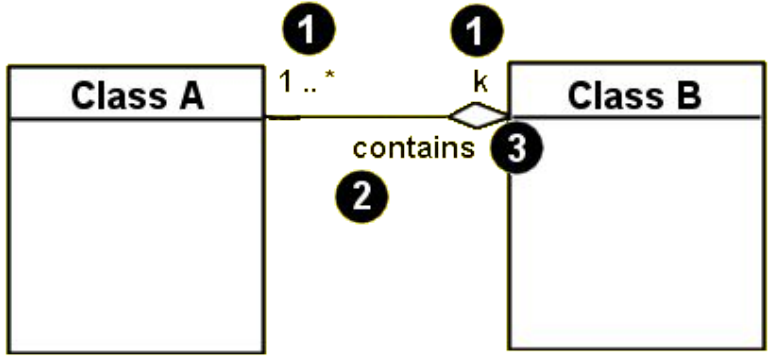
Generalization (Inheritance)

- Arrows point upward to parent
- Represents an "is-a" relationship.
- An abstract class name is shown in italics.
- Sub classes are specializations of SuperClass.



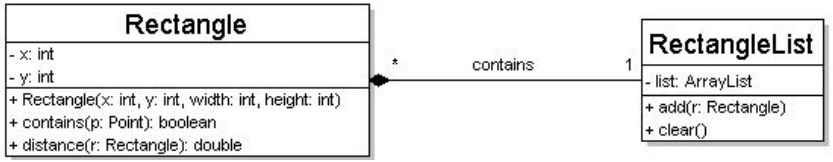
Association

- Associational relationship contains
  - Multiplicity (how many are used)
    - * ⇒ 0, or more
    - 1 ⇒ 1 exactly
    - 2..4 ⇒ between 2 and 4, inclusive
    - 3..* ⇒ 3 or more
  - Name (what relationship the objects have)



Cardinality of association

- **One-to-one**
  - Ex:Each student must carry exactly one ID card
- **One-to-many**
  - Ex:one rectangle list can contain many rectangles
- **Many-to-many**
  - Ex:A student can enroll in multiple courses.A course can have multiple students enrolled



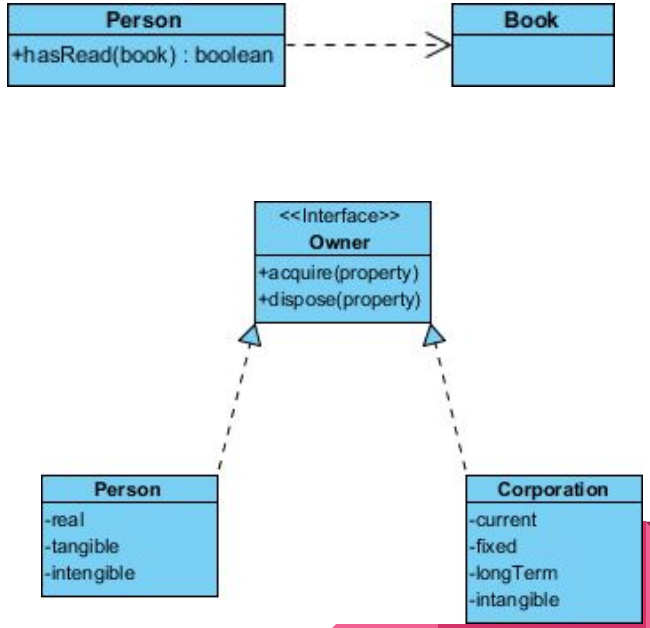
Association types

- **Aggregation**:“is part of”
  - symbolized by a clear white diamond

- **Composition**: “is entirely made of”
  - stronger version of aggregation
  - The parts live and die with the whole
  - symbolized by a black diamond

Association types

- **Dependency**: “uses temporarily”
  - Symbolized by dotted line
  - Exists between two classes if changes to the definition of one may cause changes to the other.
- **Realization**
  - Relationship between the blueprint class and the object containing its respective implementation level details



UML class diagram : Order System