CSE4006: Software Engineering

Lab 5: Unified Modeling Language (1)

Software Engineering Lab

lab(se);

UML

Unified Modeling Language

- A general-purpose modeling language which is designed to provide standard way to visualize the design of a system.
- Maintained by the OMG (Object Management Group)
  - http://www.omg.org

CRC Cards

Class-resoponsibility-colleboration Cards

- A brainstorming tool used in object-oriented design
- Proposed by Ward Cunningham and Kent Beck.

Class : Student	
Responsibility	**Colleborator**
goes to school does homework skip a class	Bus Computer

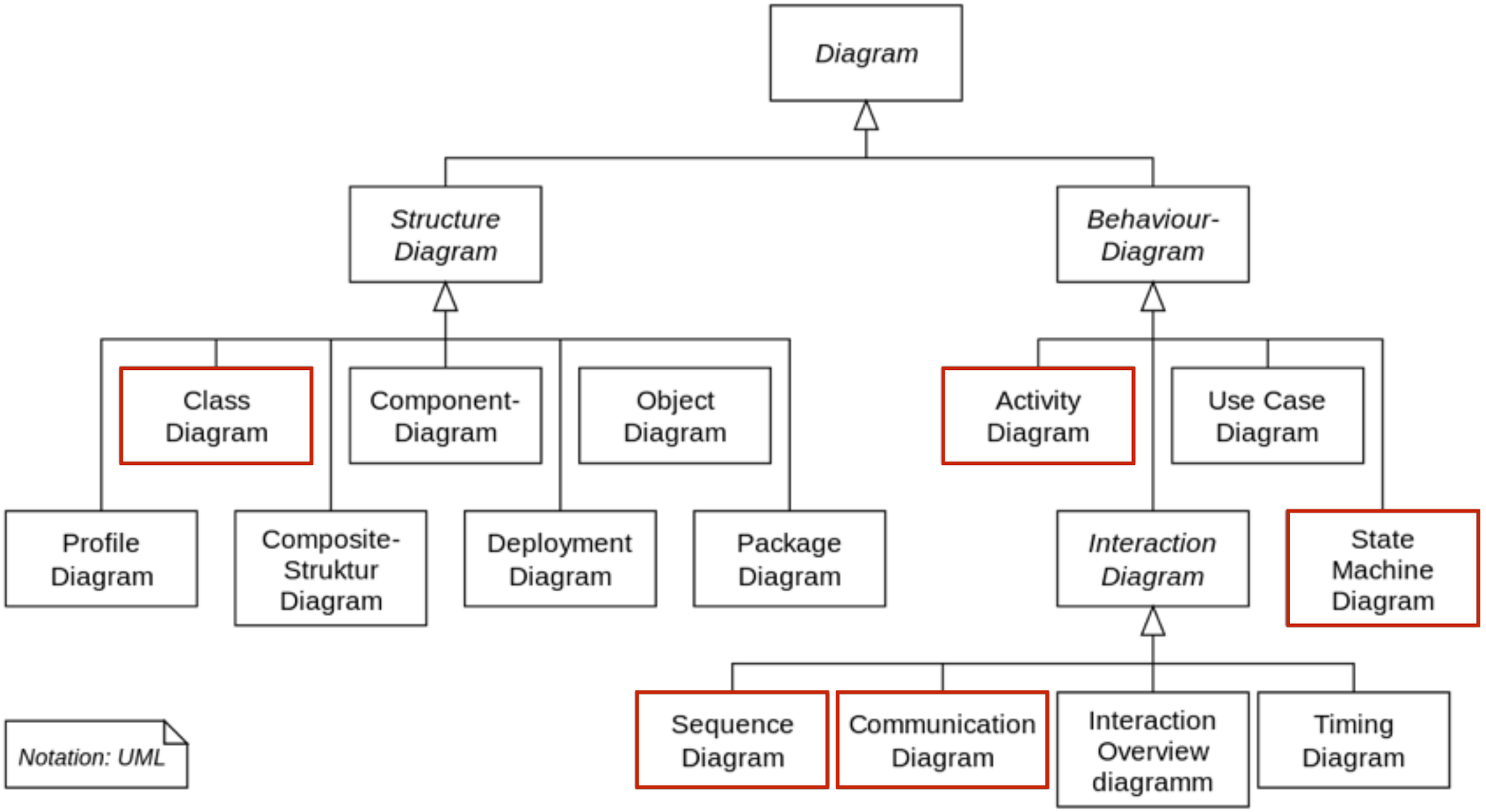
CRC Cards - Example

Class: FloorPlan

Description:

Responsibility:	**Collaborator:**
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

UML Diagrams

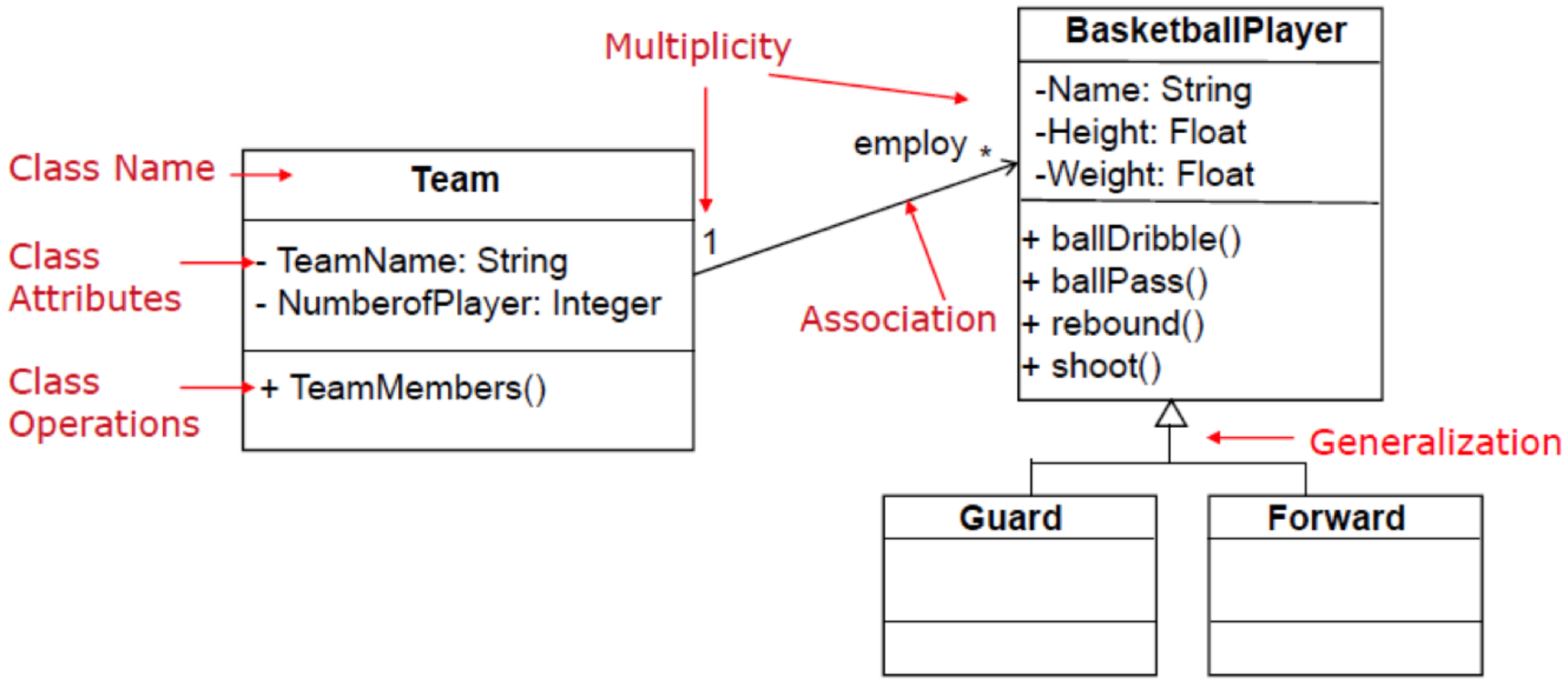


lab(se);

Class Diagrams

Class Diagram

- Description of static structure
- Showing the types of object in system and the relationships between them
- Foundation for the other diagrams

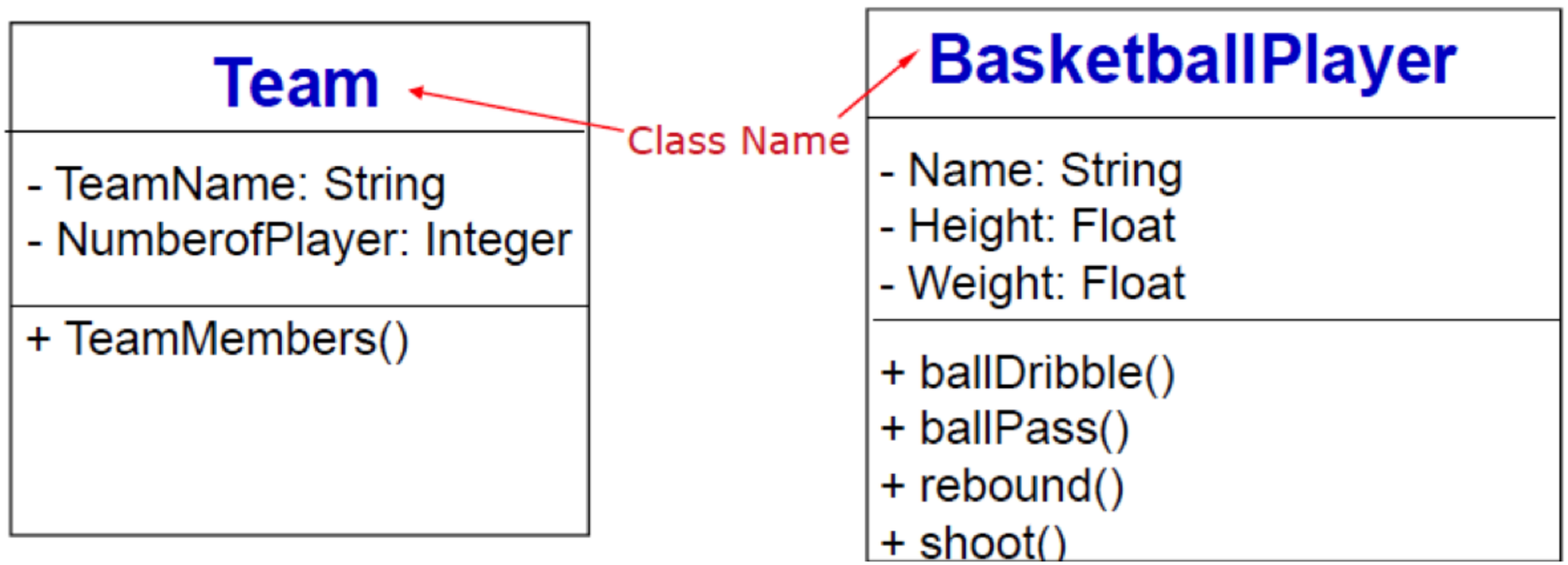


lab(se);

Class Diagrams (cont)

Classes

- Description of a set of objects
- Abstraction of the entities



lab(se);

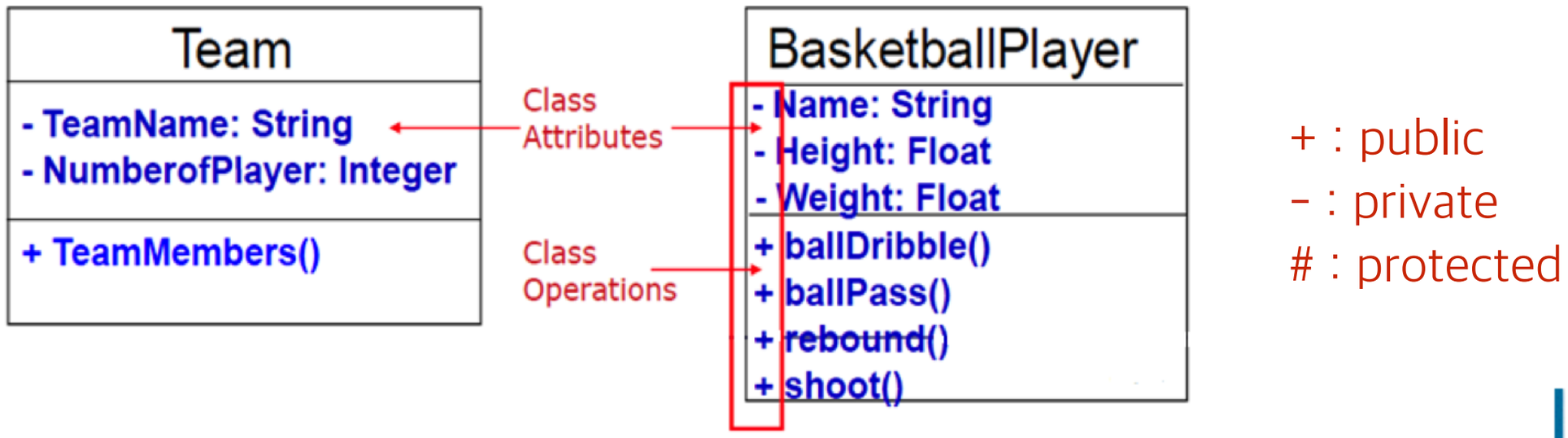
Class Diagrams (cont)

Attributes

- Represent some property of the thing being modeled
- Syntax: attributeName : Type

Operations

- Implement of a service requested from any object of the class
- Syntax: operationName(param1:type, param2:type, ⋯) : Result



lab(se);

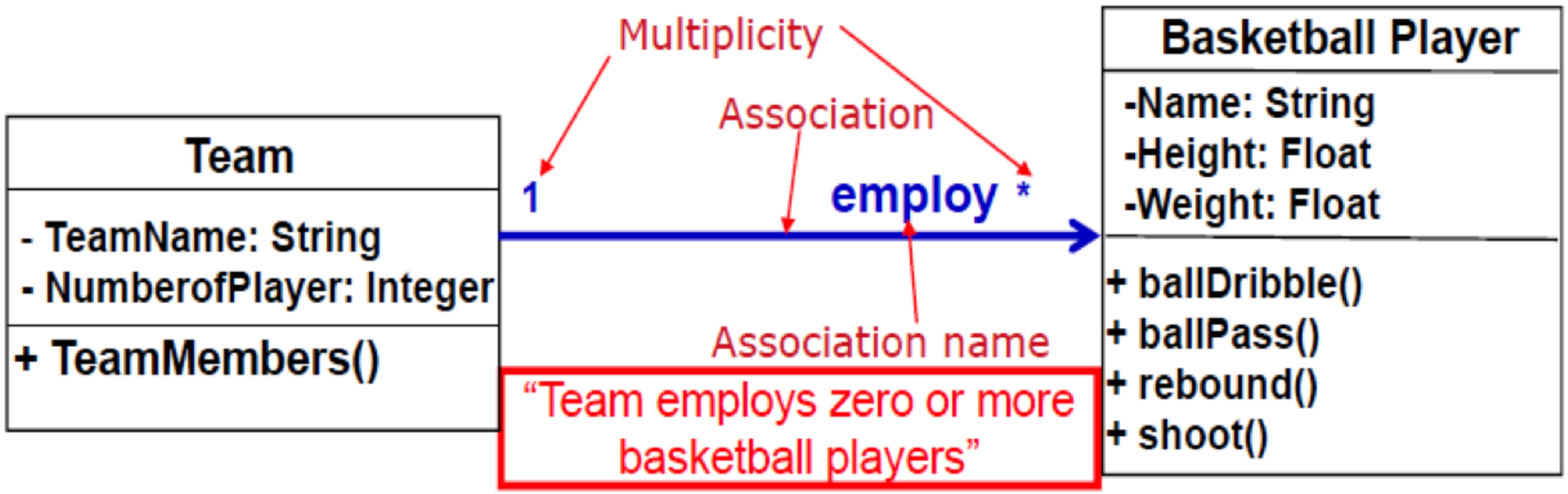
Class Diagrams (cont)

Association

- Relationship between classes that specifies connections among their instances

Multiplicity

- Number of instances of one class related to ONE instance of the other class



lab(se);

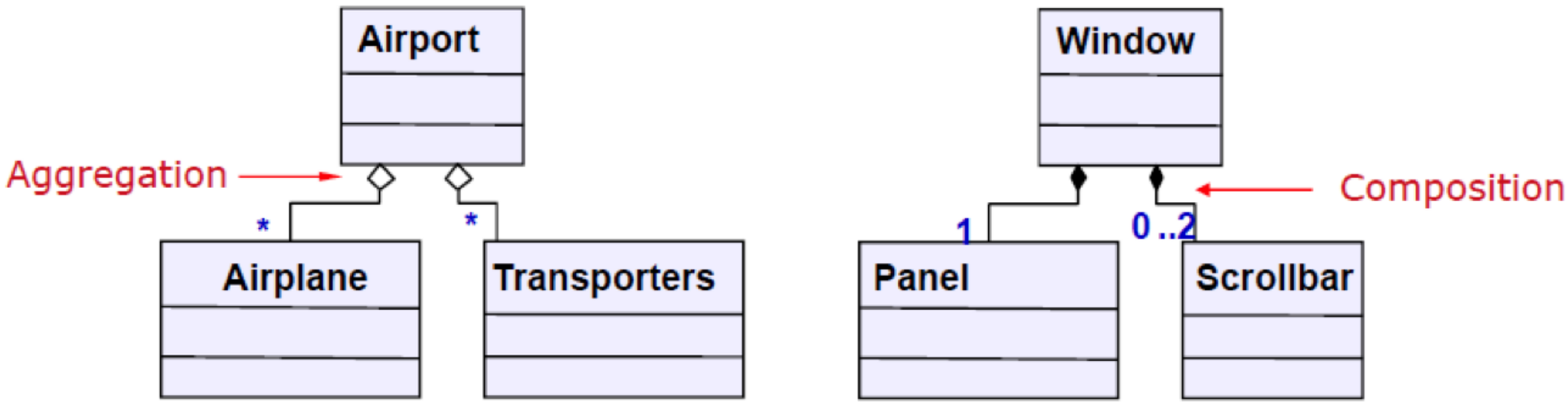
Class Diagrams (cont)

Aggregation

- **Weak** “whole-part” relationship between elements
- e.g., An airport has many airplanes

Composition

- **Strong** “whole-part” relationship between elements
- e.g., Window ‘contains a’ scrollbar

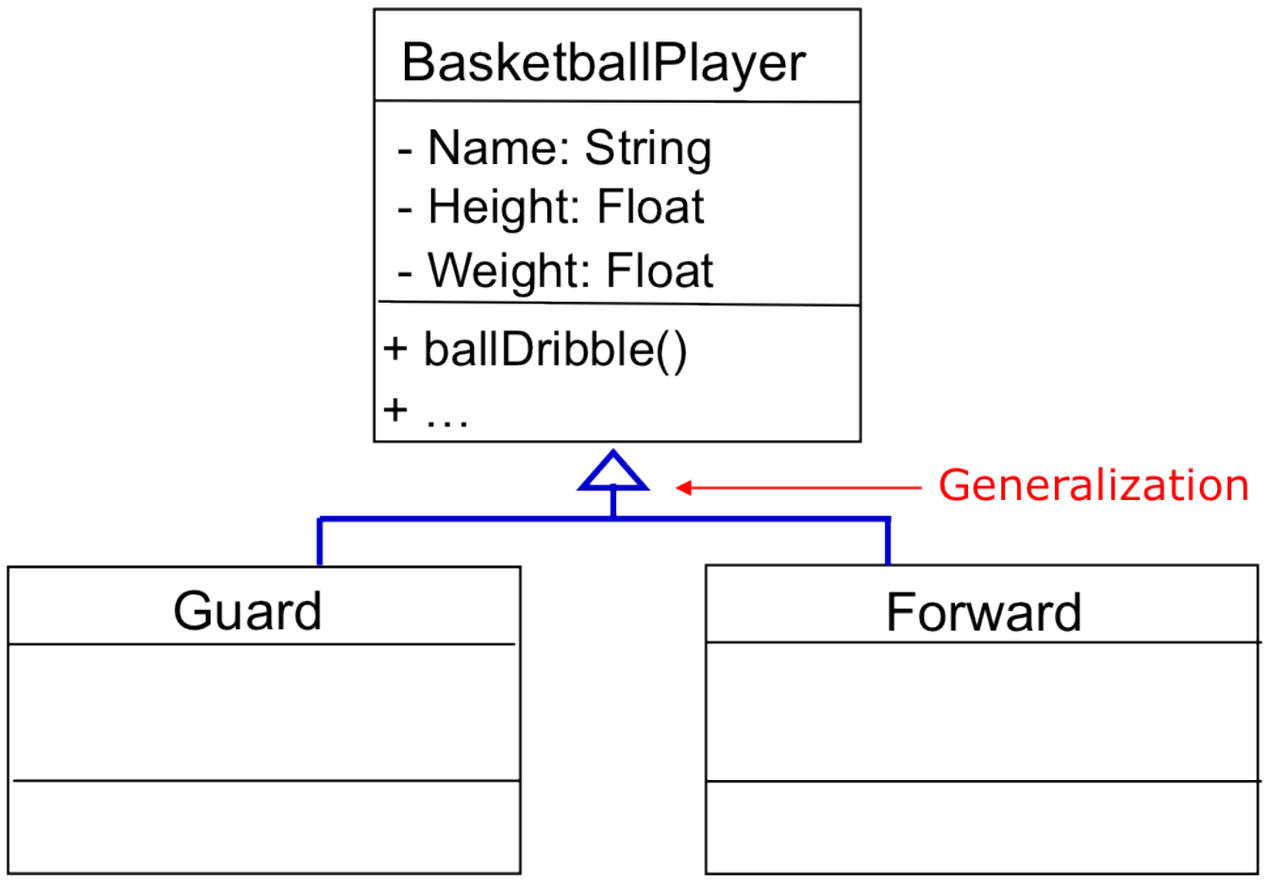


lab(se):

Class Diagrams (cont)

Inheritance

- Relationship between superclass and subclasses
- All attributes and operations of the superclass are part of the subclasses



lab(se);

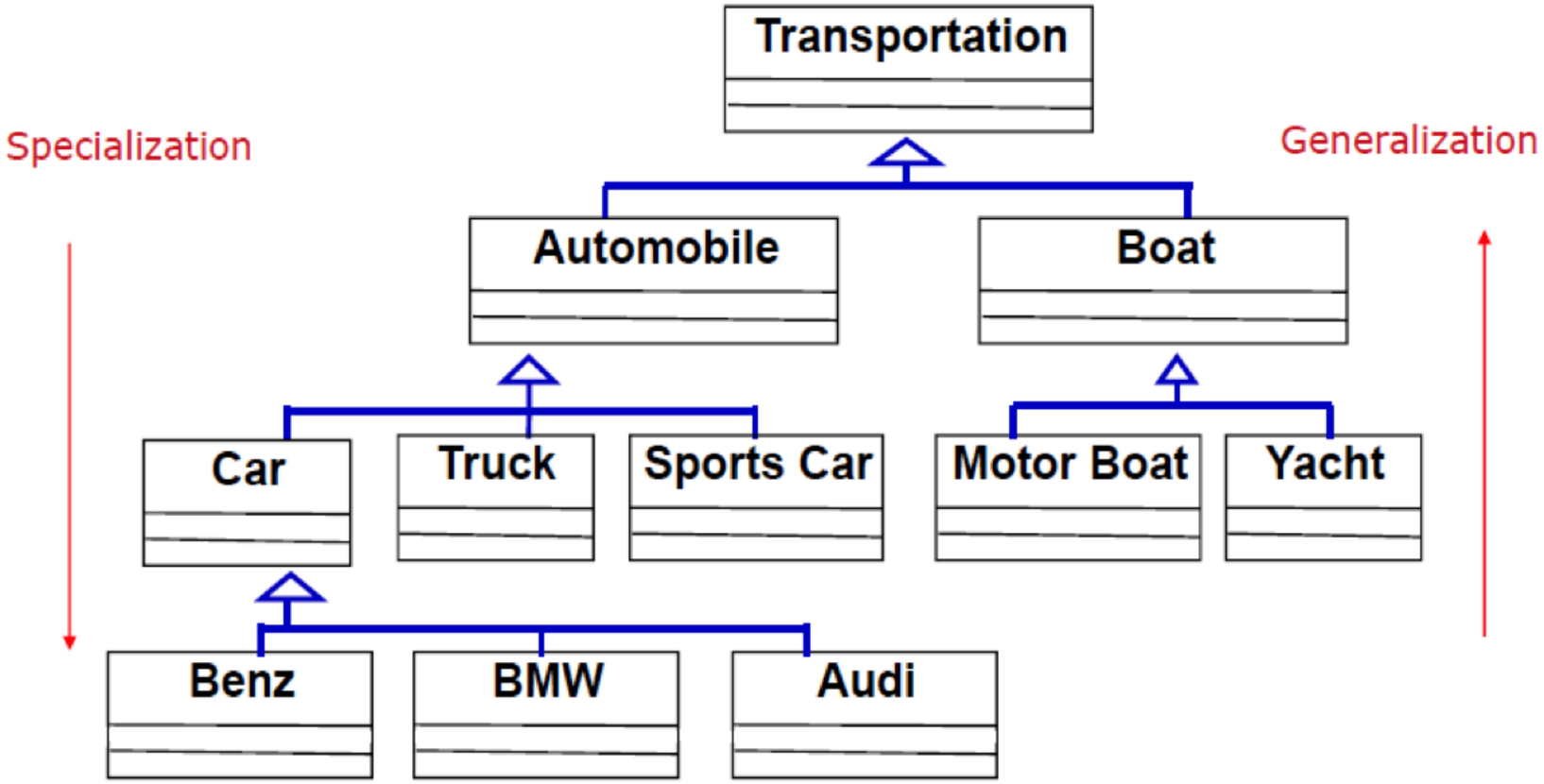
Class Diagrams (cont)

Generalization

- Building a more general class from a set of specific classes

Specialization

- Creating specialized classes base on a more general class



lab(se);

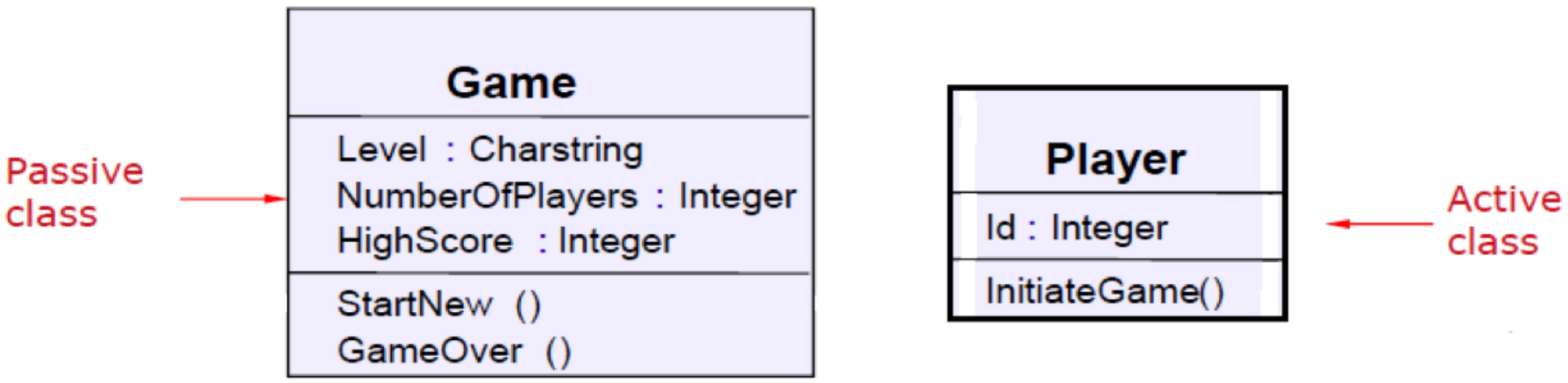
Class Diagrams (cont)

Active class

- Own a **thread control** and can initiate control activity
  - used when asynchronous communication is necessary
  - typically modeled with a state machine of its behavior
  - encapsulated with **ports** and **interfaces**

Passive class

- created as part of an action by another object
  - own address space, but not thread of control
  - executed under a control thread anchored in an active object



.ab(se);

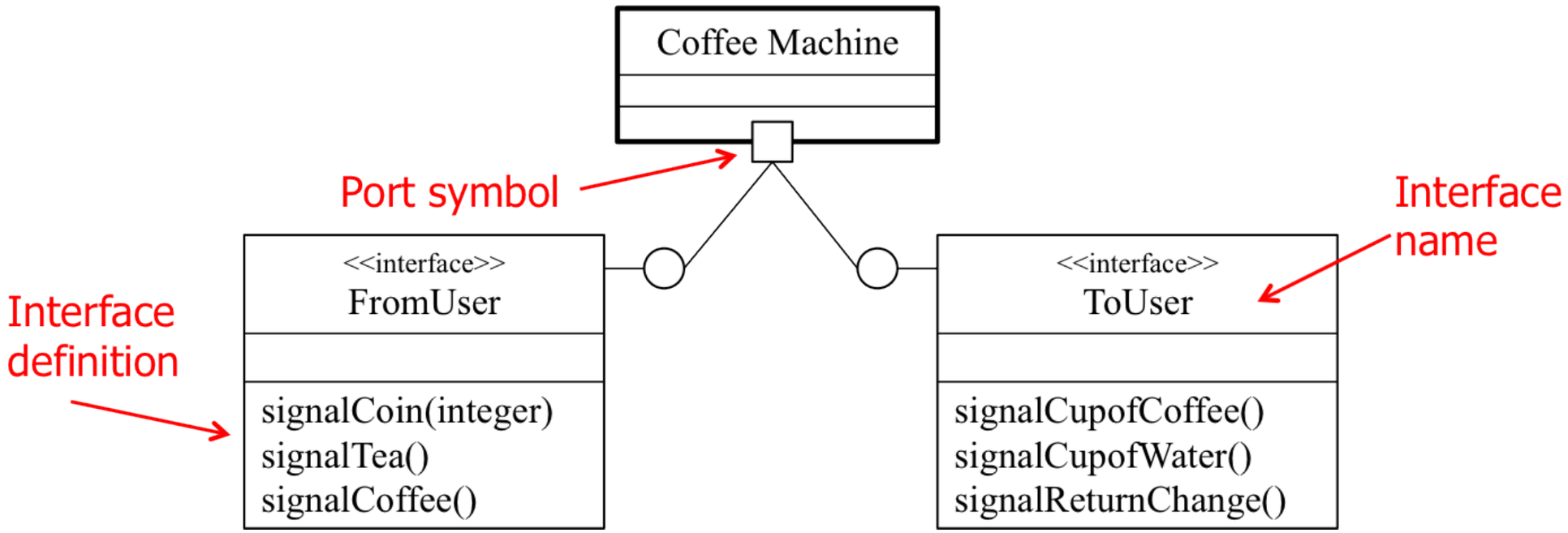
Class Diagrams (cont)

Ports

- Define an interaction point on a classifier with external environment

Interface

- Describe behavior of objects without giving their implementation
  - each class implements the operations found in the interface

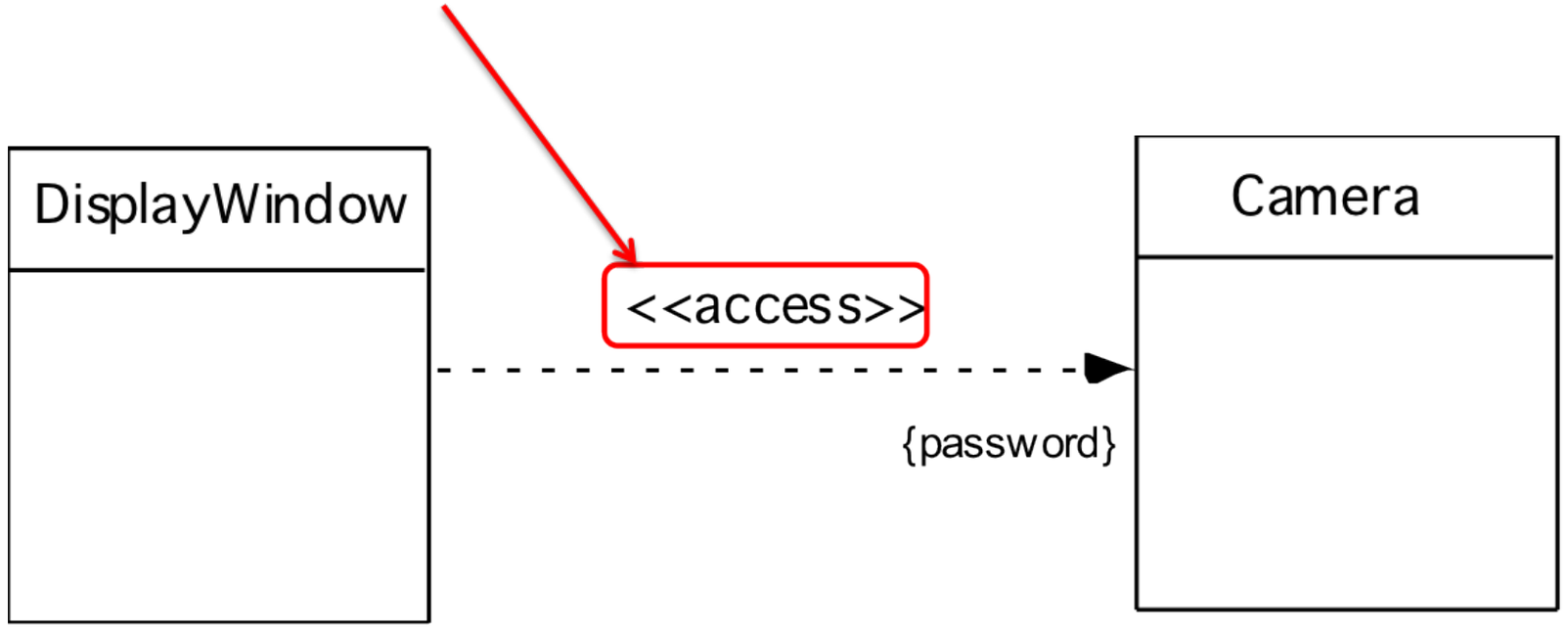


lab(se);

Class Diagrams (cont)

Stereotype in UML

- UML extension mechanism
- enables users to define the meaning of special modeling element
- represented with double angle brackets



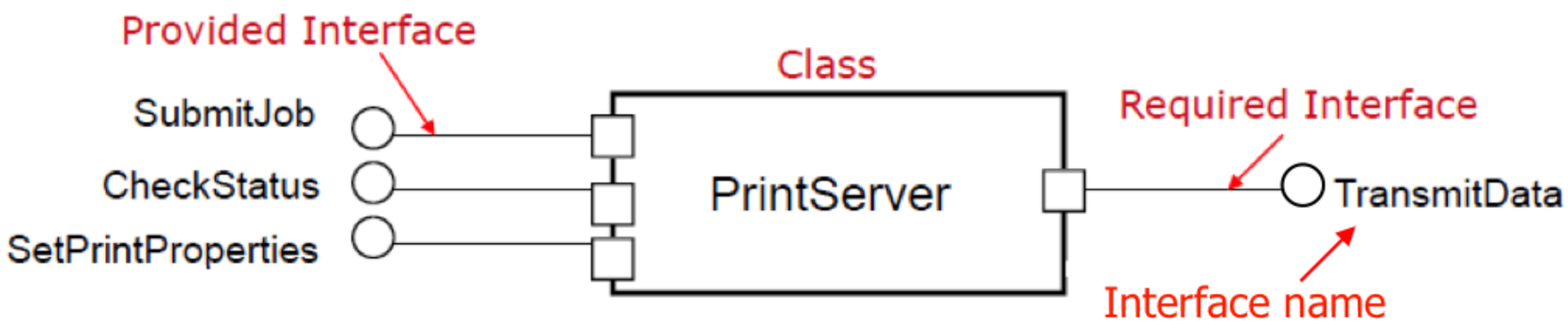
Class Diagrams (cont)

Provided interface

- Class provides the services of the interface to outside callers
- what the object can do
- Provided interface accept incoming signal from outside callers

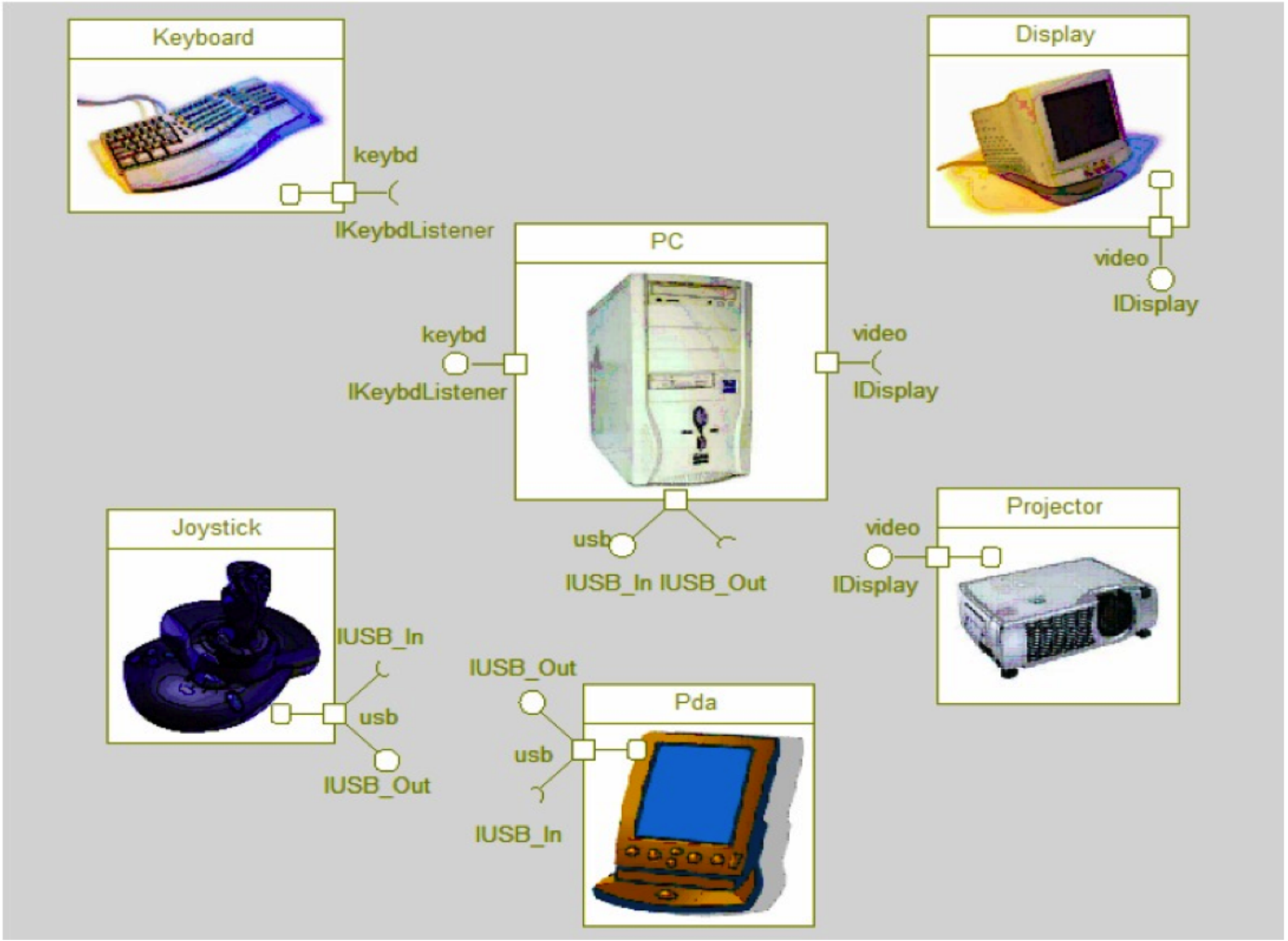
Required interface

- Class uses to implement its internal behavior
- What the object needs to do
- Outgoing signal are sent via required interface



lab(se);

Computer Device Example



Tips for Class Modeling

Finding Classes

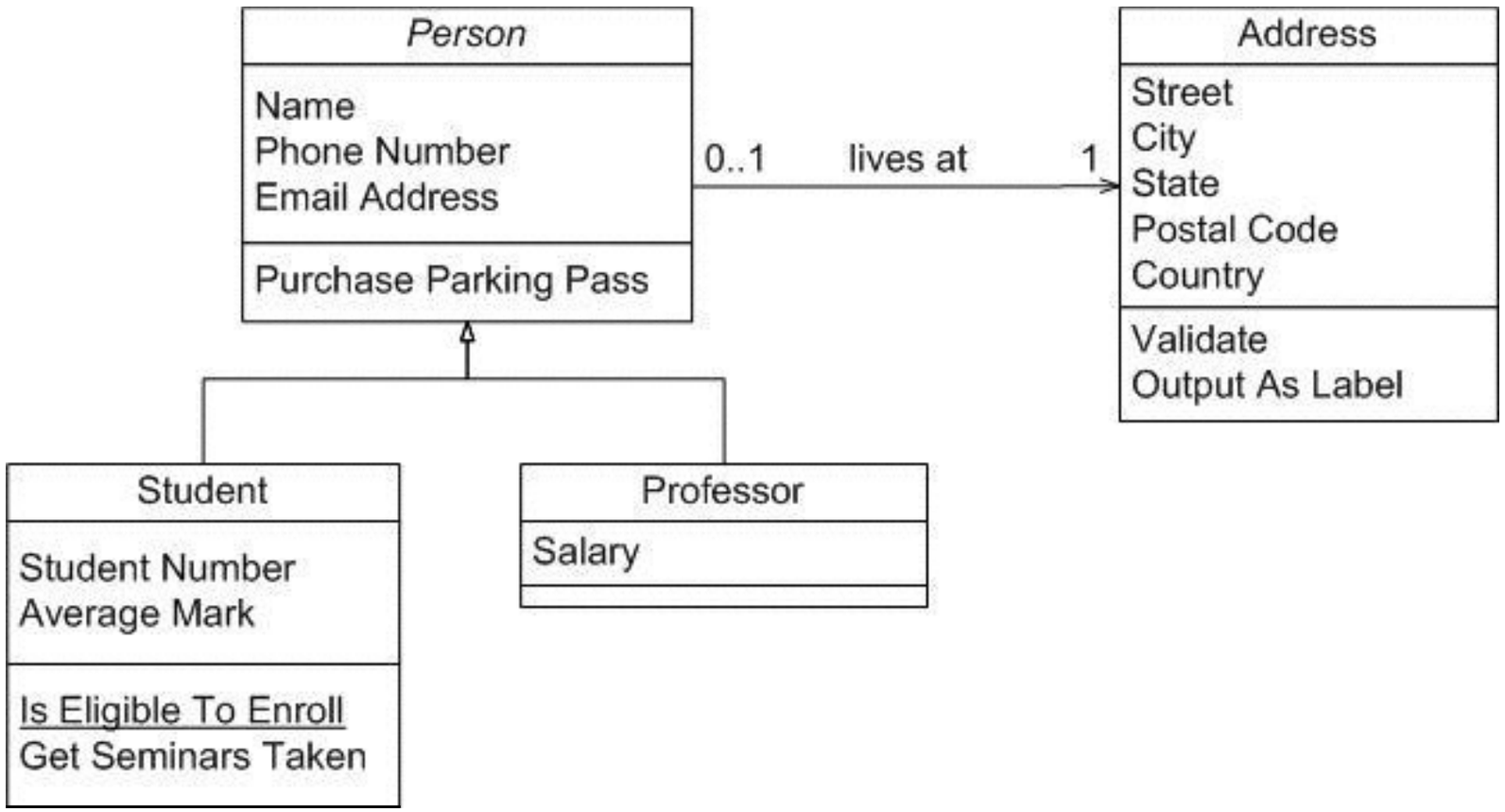
- Do we have things that should be stored or analyzed?
- Do we have external system?
  - external system is modeled as class
- Do we have any patterns, class libraries, components, etc?
- Are there devices that the system must handle?

Make explicit traceability whenever possible

- Try to capture class/attributes from nouns of use-cases and operations from verb of use-cases
- Always draw class diagram in conjunction with some form of behavioral diagrams

lab(se);

Class Diagrams – Example



Exercise

Write CRC cards and class diagram for POS system of book store

- Stock management, Payment management
- At least 4 classes