II.Static modelling

❑ Revision

- **What is a model?**

✓ A simplification with a purpose.

✓ Abstract representation of a real-world.

- **What is modelling?**

✓ Designing a software application before coding

II.Static modelling

❑ What is a static model?

- **Abstract representation of the real world; a simplification with the purpose of addressing only the static structural view of the problem - which does not vary with time.**

Is child of

II.Static modelling

❑ What is a static model?

- **It defines the classes of the system, the attributes of the classes, the relationships between classes and the operations of each class;**

Is child of

II.Static modelling

❑ What is a static modelling?

- A modelling process where the UML class diagram notation is used to depict the static model.
- At the end of static modelling , we have a static model.
- Static model are depicted on a class diagram.

II.Static modelling

❑ Association between classes

What is an association?

- **It defines the relationship between two or more classes.**
- **It denotes a static structural relationship between classes.**

II.Static modelling

❑ Association between classes

Example :

- **Employee works in a department.**

✓ **"Employee" and "Department" are classes.**

✓ **And "works in" is an association.**

NB: The <u>classes are nouns</u> whereas the <u>association is usually a verb or verb phrase.</u>

Is child of

II.Static modelling

❑ Association between classes

A link

- **A connection between instances of the classes (called objects)**
- **it represents an instance of an association between classes.**
- **Exist between two objects if and only if there is a association between their corresponding classes.**

Is child of

II.Static modelling

❑ Association between classes

A link (Example)

- **Jane works in manufacturing.**

✓ **Jane is an instance of Employee**

✓ **Manufacturing is an instance of Department.**

Is child of

II.Static modelling

❑ Association between classes

- Association are bidirectional
- The name of the association is in the forward direction.
- There is also an opposite direction of the association which is often not explicitly stated;

Is child of

II.Static modelling

❑ Association between classes

- **Example**

➢ **Employee Works in Department**

➢ **Department Employs Employee**

Is child of

II.Static modelling

❑ Association between classes

- Association types
- ➢ Associations are most often binary – representing relationship between two classes.
- The can also be unary (self associations)

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams

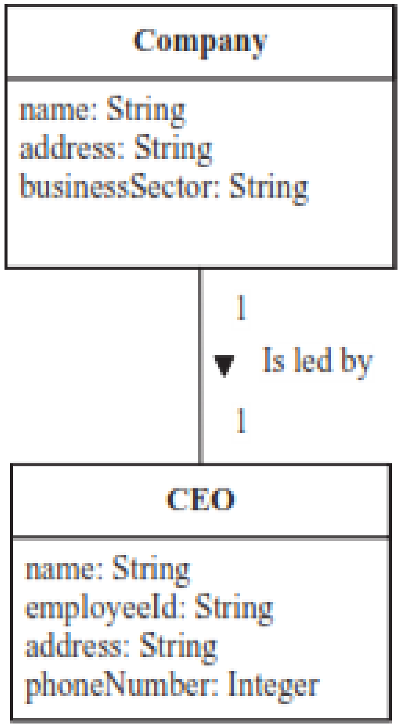
- On class diagram, an association is shown as an arc joining the two classes boxes, with the name of the association next to the arc.

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(1)

▪ **Company Is led by CEO**



Example of one-to-one association

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Multiplicity of Association**

➢ **one-to-one association b/t two classes**

✓ **Its one-to-one in both directions**

✓ **An object of either class has a link to only one object of the other class.**

✓ **Example : Company is led by CEO**

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Multiplicity of Association(1)**

➢ **One-to-many association b/t two classes**

✓ **Its one-to-many association in one direction and one-to-one association between them in the opposite direction.**

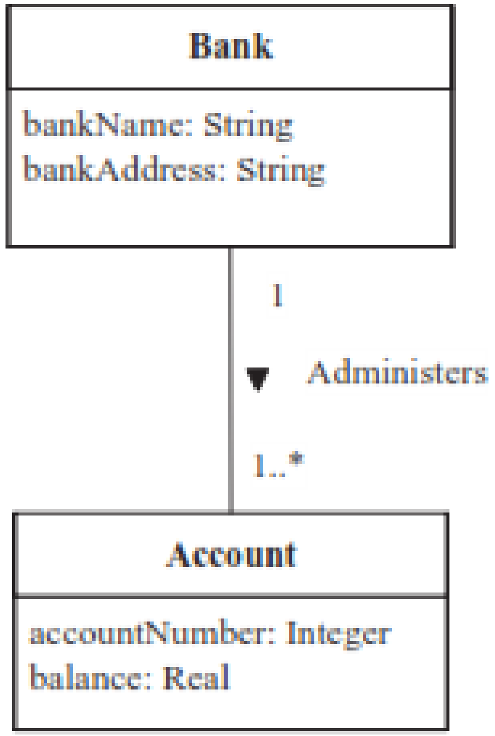
II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Multiplicity of Association**

➢ **One-to-many association**

One-to-many association



Is child of

II.Static modelling

❑ **Depicting Association on class Diagrams(2)**

- **Multiplicity of Association(2)**

➢ **One-to-many association b/t two classes**

✓ **Example : Bank Administers Account; an individual bank administers many accounts, but an individual account is administered by only one bank.**

Is child of

II.Static modelling

- **Depicting Association on class Diagrams(2)**
  - **Multiplicity of Association(2)**
    - **Numerically specified association**
      - ✓ **Association that refers to specific number**
      - ✓ **The opposite direction is one-to-one**

Example : ***Car Is entered through a Door.***

- ✓ **One car has two doors or four doors but never one, three , or five doors.**

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Multiplicity of Association(3)**

➢ **Optional association**

✓ **There may not always be a link from one object in one class to an object in the other class.**

✓ **It is possible to have zero, one or more association.**

Is child of

II.Static modelling

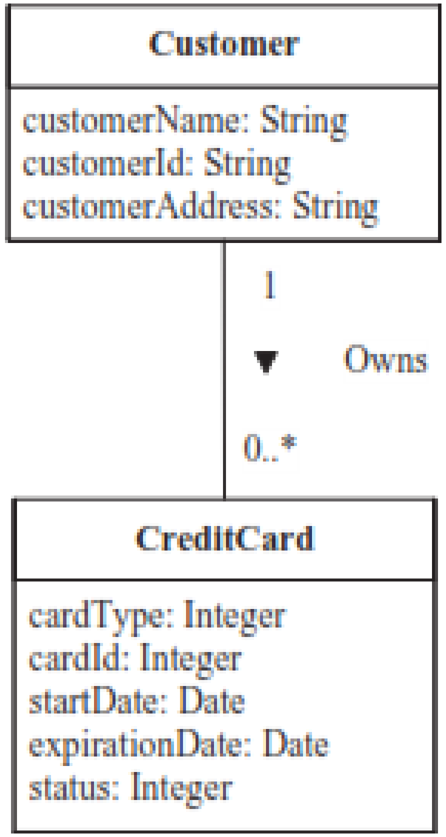
❑ Depicting Association on class Diagrams(2)

- **Multiplicity of Association(4)**

➢ **Optional association**

✓ **Example:**

Optional (zero, one , or many)



Is child of

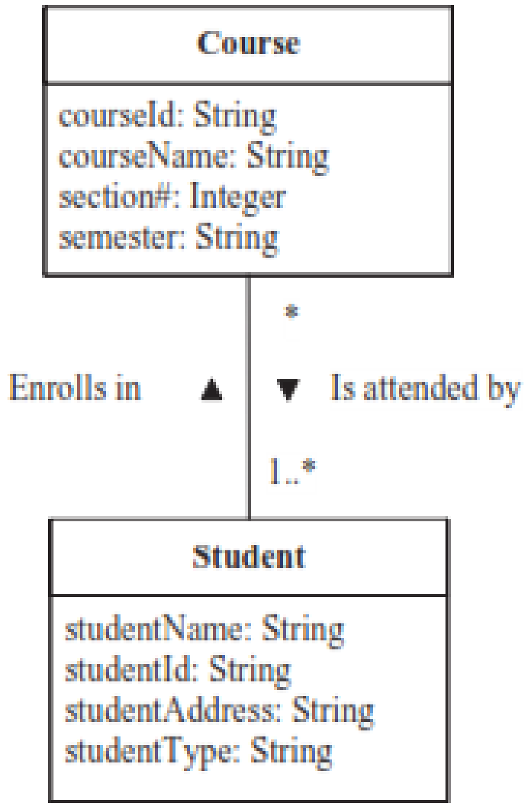
II.Static modelling

❑ Depicting Association on class Diagrams(2)

▪ **Multiplicity of Association(5)**

➢ **Many-to-Many association**

✓ **Example:**

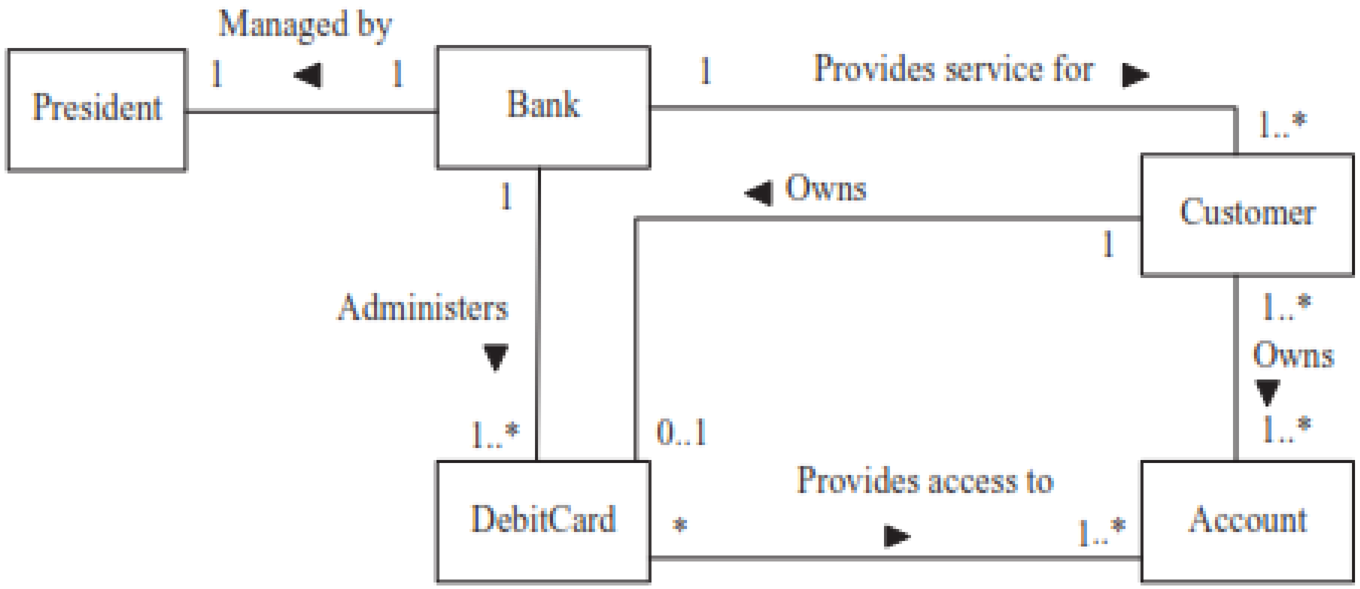


II.Static modelling

❑ Depicting Association on class Diagrams(2)

▪ **Multiplicity of Association(6)**

➢ **Example of associations on a class diagram**



Is child of

II.Static modelling

❑ **Depicting Association on class Diagrams(2)**

- **Ternary Associations**

➢ **Three-way association among classes**

➢ **Shown as a diamond joining the three classes.**

✓ **Eg. Association of the classes Buyer, Seller, Agent.**

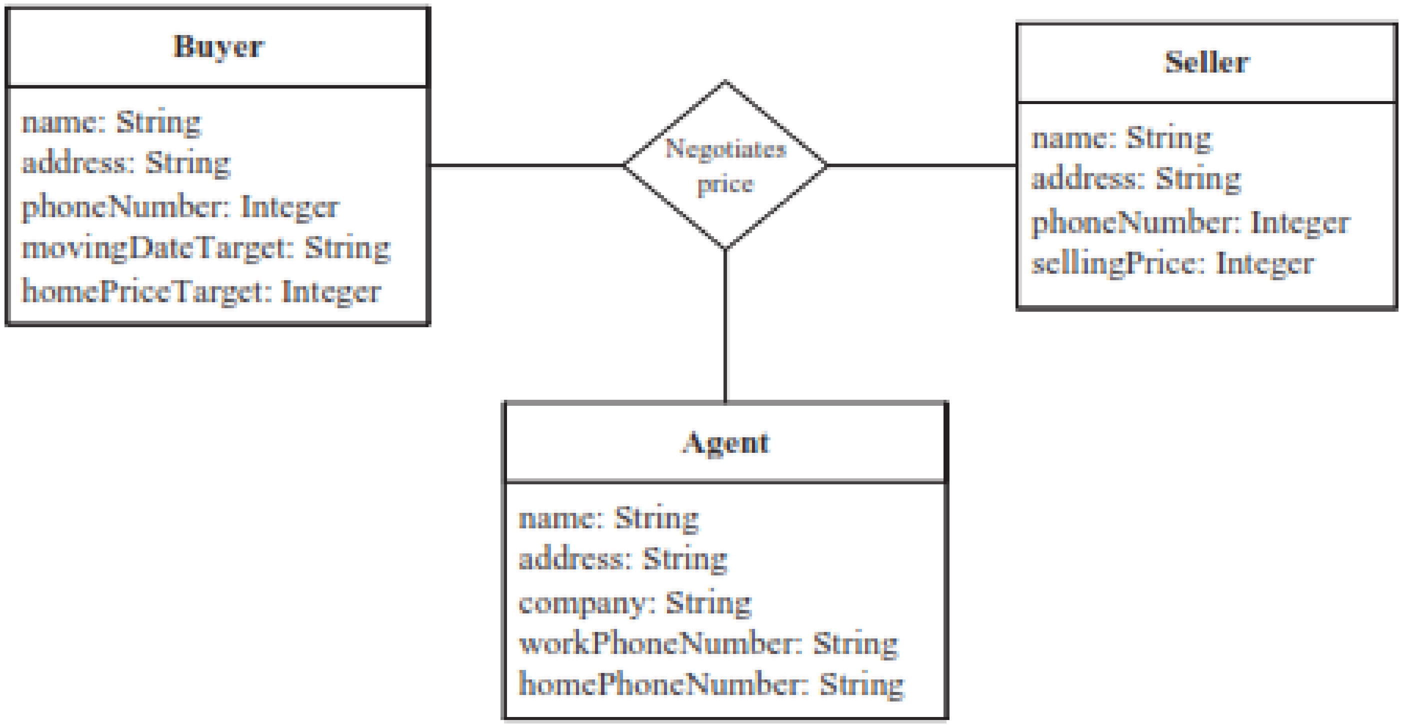
Is child of

II.Static modelling

- Depicting Association on class Diagrams(2)

  - Ternary Associations(1)

➢ **Example:**



Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **unary Associations(2)**

➢ **A self association**

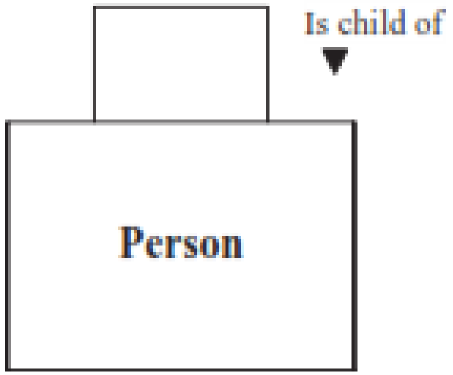
➢ **Association between an object of one class and another object in the same class.**

➢ **Example :**

✓ **Person Is child of Person**

✓ **Person is married to Person**

✓ **Employee is boss of Employee**



Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Association Classes**

➢ **A class that models an association between two ore more classes.**

➢ **The attributes of the association class are the attributes of the association**

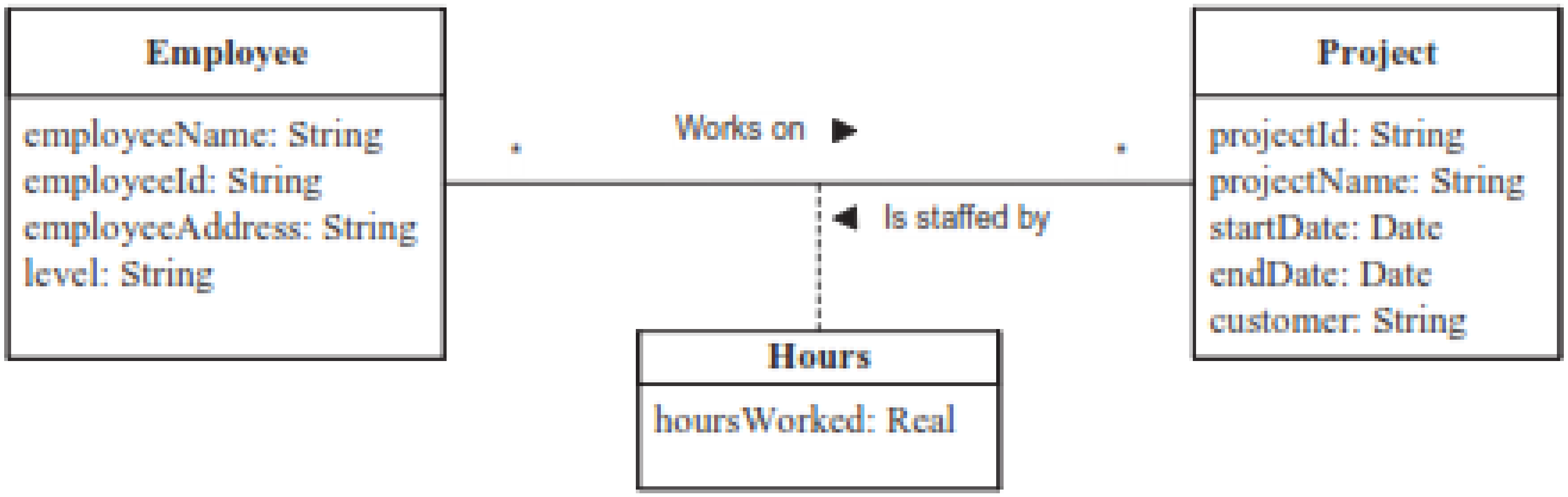
➢ **It happens often in many-to-many association where an attribute does not belong to any of the classes but to the association.**

II.Static modelling

❑ Depicting Association on class Diagrams(2)

▪ **Association Classes**

➢ **Example:**



II.Static modelling

- **Depicting Association on class Diagrams(2)**

- **Composition and aggregation hierarchies**

- **Both addresses a class that is made up of other classes.**

- **Both are special forms of a relationship in which two classes are connected by the whole/part relationship.**

- **The relationship between *parts and the whole* is an *Is-part-of* relationship.**

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Composition and aggregation hierarchies(1)**

➢ **Composition**

✓ **Stronger relationship than an aggregation.**

✓ **Aggregation is a stronger relationship than an association.**

✓ **Relationship among instances**

➢ **The parts object are created , live , and die together with the whole.**

Is child of

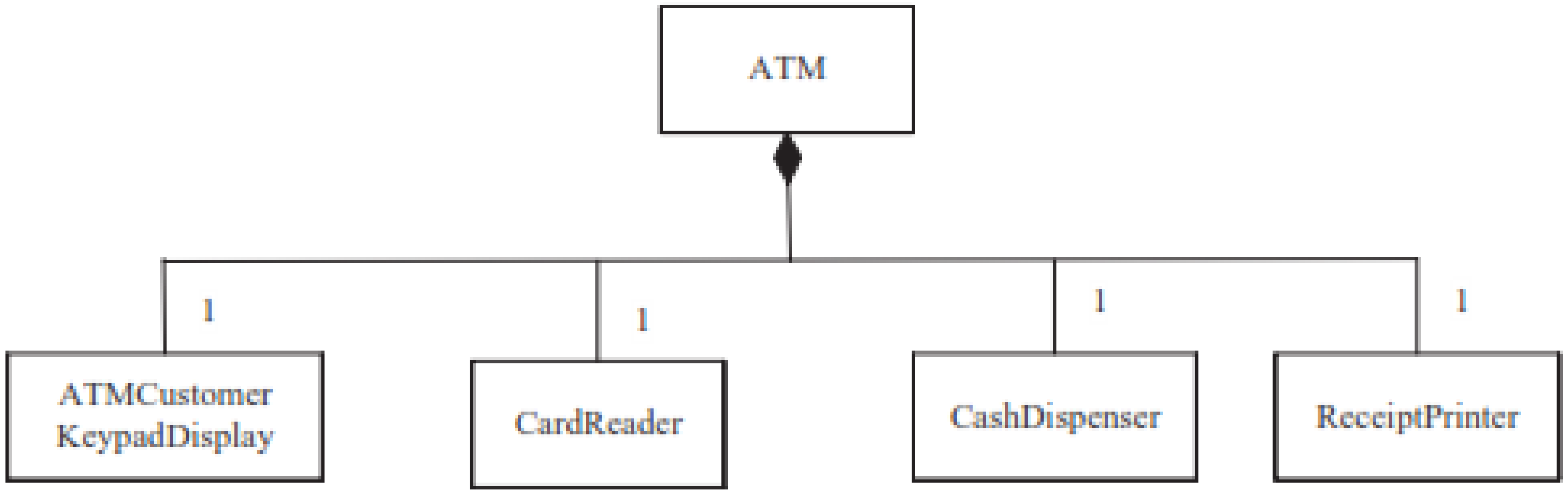
II.Static modelling

- **Depicting Association on class Diagrams(2)**
- **Composition and aggregation hierarchies(2)**
- **Composition**
  - ✓ **The part object can belong to only one whole**
  - ✓ **A composite class often involves a physical relationship between the whole and the parts.**
  - ✓ **ATM is machine having four parts: Card Reader, a cash Reader, a Receipt Printer, and ATM customer keypad display classes.**

Is child of

II.Static modelling

- ❑ **Depicting Association on class Diagrams(2)**
- ▪ **Composition and aggregation hierarchies(3)**
- ➢ **Example: Composition hierarchy**



Is child of

II.Static modelling

- ❑ **Depicting Association on class Diagrams(2)**
  - ▪ **Composition and aggregation hierarchies(3)**
    - ➢ **Aggregation**
      - ✓ **A weaker form of whole/part relationship**
      - ✓ **Part instances can be added to and remove from the aggregate whole.**
      - ✓ **Aggregation are likely to be use to model conceptual classes than physical classes.**
      - ✓ **A part could belong to more than one aggregation.**

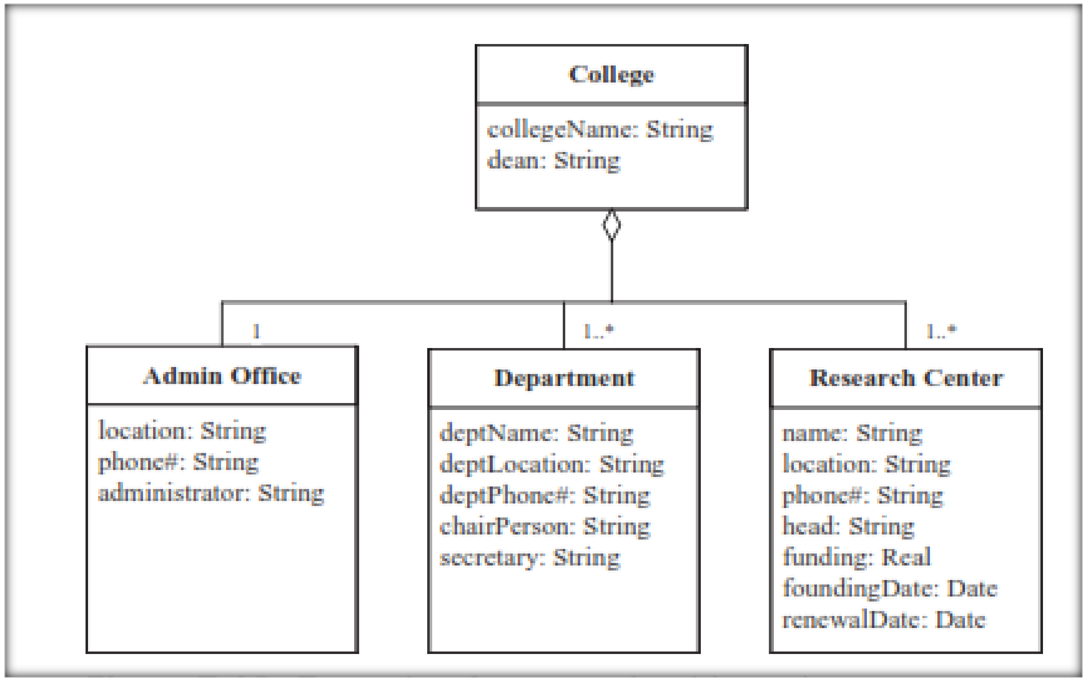
Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

▪ **Composition and aggregation hierarchies(4)**

➢ **Aggregation**



Is child of

II.Static modelling

❑ **Depicting Association on class Diagrams(2)**

▪ **Generalisation/specialisation hierarchy**

➢ **Some classes are similar but not identical**

➢ **Some classes have some attributes in common and other that are different**

➢ **Common attributes are abstracted into a generalized class – superclass**

➢ **The different attributes are properties of the specialized referred to as a subclass.**

Is child of

II.Static modelling

- **Depicting Association on class Diagrams(2)**

- **Generalisation/specialisation hierarchy(1)**

- **There is Is-a relationship between the subclass and the superclass.**
- **The superclass is also referred to as a parent class or ancestor class.**
- **The subclass is also referred to as a child class or descendent class.**

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Generalisation/specialisation hierarchy(2)**

➢ **Each subclass inherits the properties of the superclass but then extends these properties in different ways.**

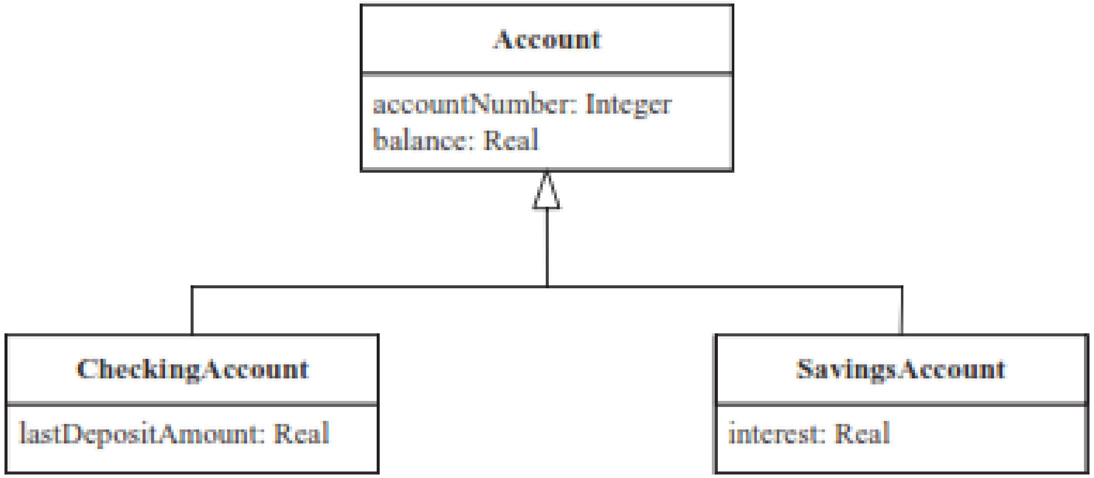
➢ **The properties of a class are its attributes or operations**

➢ **Inheritance allows the adaptation of the superclass to form the superclass.**

Is child of

II.Static modelling

- **Depicting Association on class Diagrams(2)**
- **Generalisation/specialisation hierarchy(3)**
- **Saving Account is a Account**
- **Checking Account is a Account**



Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Generalisation/specialisation hierarchy(4)**

➢ **A discriminator is an attribute that indicates which property of the object is being abstracted by the generalization relationship**

➢ **Example : The discriminator in the Account generalization just given, account Type, discriminates between checking Account and Saving Account.**

Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

▪ **Generalisation/specialisation hierarchy(5)**

➢ **The discriminator does not need to be an attribute of the generalized or specialised classes.**

➢ **It not an attribute of the Account superclass or the two subclasses**

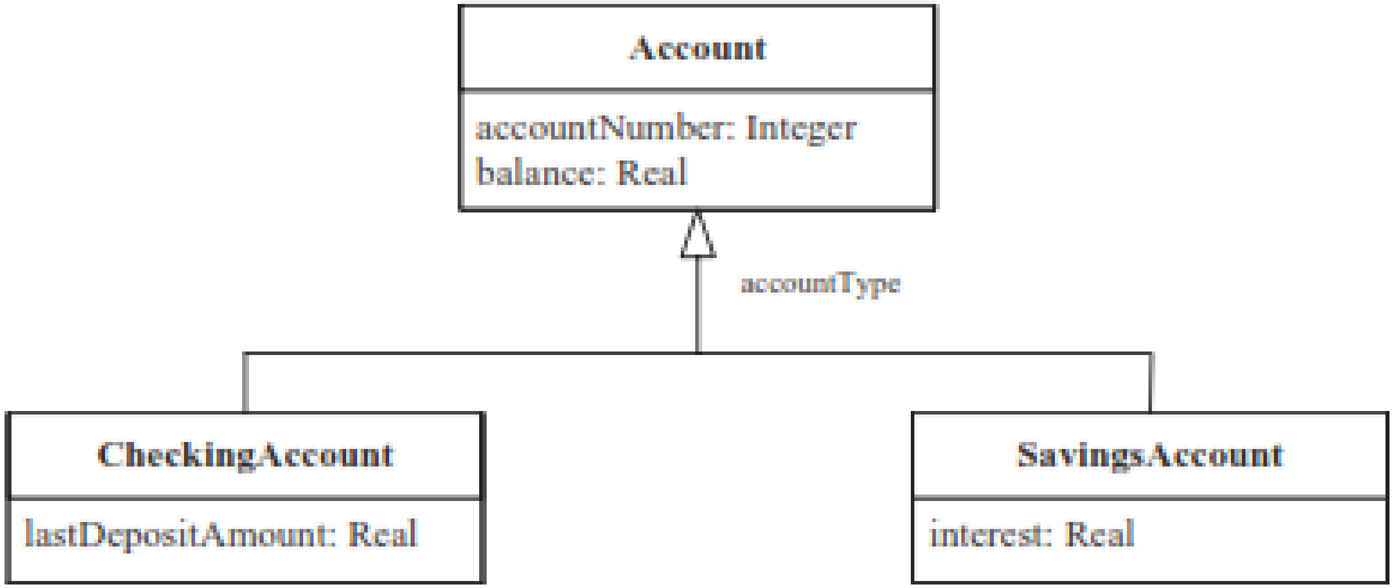
Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Generalisation/specialisation hierarchy(6)**

➢ **Example of Discriminator - accountType**



Is child of

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Constraints**

➢ **It specifies a condition or restriction that must be true**

➢ **It is express in a textual language**

➢ **One kind of constraint is a restriction on the possible values of an attribute**

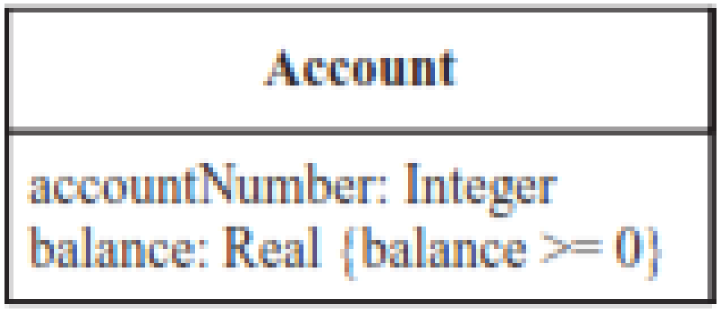
➢ **Example : Accounts are not allowed to have a negative balance.**

II.Static modelling

❑ Depicting Association on class Diagrams(2)

- **Constraints(1)**

➢ **Accounts in banking example can not have a negative balance and it could be express as a constraint on the attribute balance of the Account class to state that the balance is not allowed to be negative. { balance>=0}**



Is child of

II.Static modelling

❑Static Modeling of the Problem Domain.

II.Static modelling of the Problem domain

- **What is a problem domain?**

- **All information that defines the problem and constraints the solution.**
- **It includes the goals that the problem owner wishes to achieve, the context within which the problem exist, and all rules that define essential functions or other aspects of any solution product.**

Is child of

II.Static modelling of the Problem domain

❑ **What is a solution domain?**

- **It defines the environment where the solution will come to work.**
- **The process by which the solution is arrived at.**
- **The design, construction , testing, operation and functions of the solution product itself.**

II.Static modelling of the Problem domain(1)

- In static modeling of a problem domain, initial emphasis are :-

✓ Modeling Physical classes and Entity classes.

➢ **Physical classes** are classes that can be seen and touched.

➢ **Entity Classes** are data-intensive classes that are often persistent – long living.

Is child of

II.Static modelling of the Problem domain(2)

- **Modeling Physical classes**
- ✓ **Classes that have physical characteristics.**

Example :

- **Physical devices often found in embedded application, users, external systems, and timers**

Is child of

II.Static modelling of the Problem domain(3)

- **Modeling Entity classes : mostly long living classes found mostly in information system ;**

- **Example**

In a banking application – having account and transaction.

Is child of

II.Static modelling of the Problem domain(4)

- **Assignment :**

1. Give 10 examples of entity classes
2. Give 8 examples of physical classes

Is child of

II.Static modelling of the Problem domain(4)

- In embedded systems with several physical devices such as sensors and actuators, class diagram can help in modeling the real-world devices.

II.Static modelling of the Problem domain(5)

- Example(1) of a static model of the problem domain for the banking application where **a bank provides a service for several ATMs with each ATM being a composite class consisting of a Card Reader, a cash Dispenser, a Receipt printer and an ATM customer keypad display.**

Is child of

II.Static modelling of the Problem domain(5)

- Example(2) in the Banking system , modeling the ATM system where their associations and multiplicity can be modeled.
- Use composite classes to show how a real-world class is composed of other classes.

II.Static modelling of the Problem domain(6)

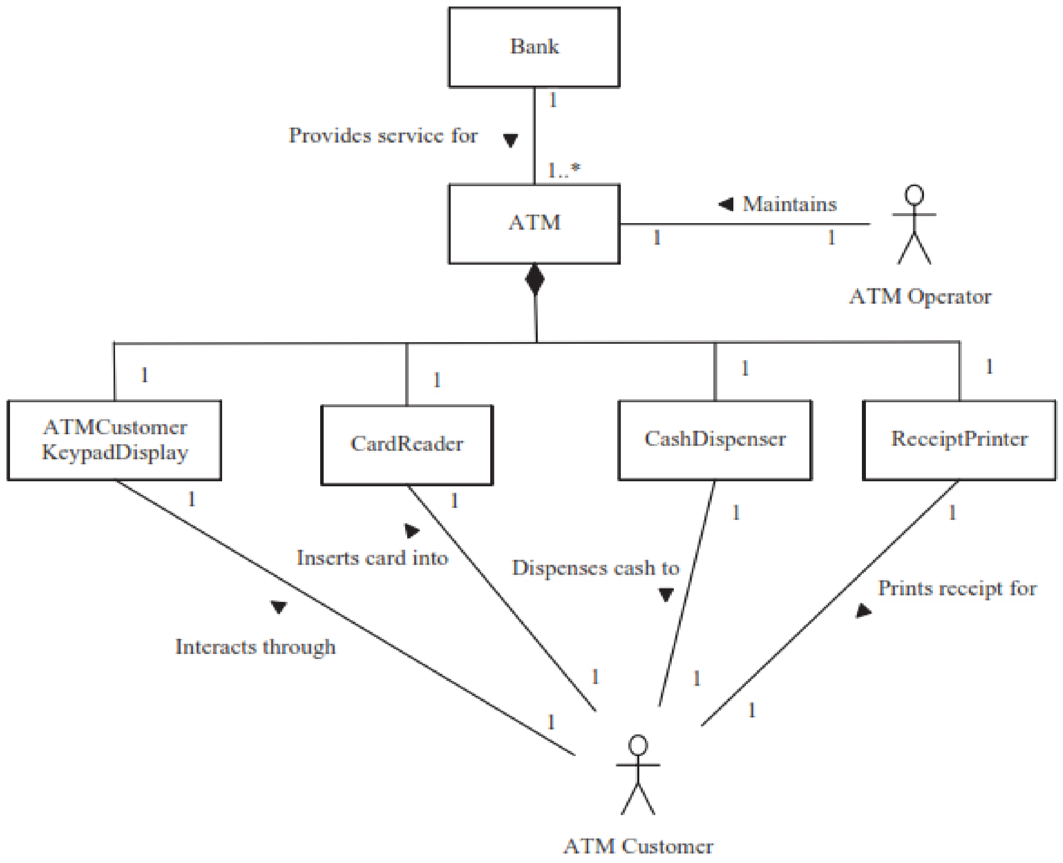
How the system works?

- The **<u>ATM Customer actor</u>** inserts the card into the Card Reader and interacts through the ATM Customer Keypad Display.
- The Cash Dispenser dispenses cash to the ATM Customer actor. The Receipt Printer prints a receipt for the ATM Customer actor.
- The physical entities represent classes in the problem domain for which there will need to be a conceptual representation in the software system.

Is child of

II.Static modelling of the Problem domain(7)

- **Static Model of a bank system with ATM**



II. Static modeling of the system context(1)

- Its very important to understand the scope of a computer system

✓ What is to be include inside the system

✓ And what is to be left outside the system

Is child of

II. Static modeling of the system context(1)

- Context modeling explicitly identifies what is inside the system and what is outside.
- Context modeling can be done at the

✓ **Total system (hardware and software) level**

✓ **Or at software system (software only) level**

NB: Context modeling at Total system or software system level

Is child of

II.Static modelling of the system context(2)

What is a system context diagram?

- **A diagram that explicitly shows the border between the system (hardware and software), which is treated as a black box, and the external environment .**
- ***System(black box) and external environment.***

Is child of

II.Static modelling of the system context(3)

Context diagram vs Use case diagram

- These views of the border around the system are more detailed than those usually provided by a use case diagram.

II.Static modelling of the system context(3)

Designing the context diagram

- In developing the system context diagram, it is helpful to consider

✓ The context of the total hardware/software system before considering the context of the software system.

Is child of

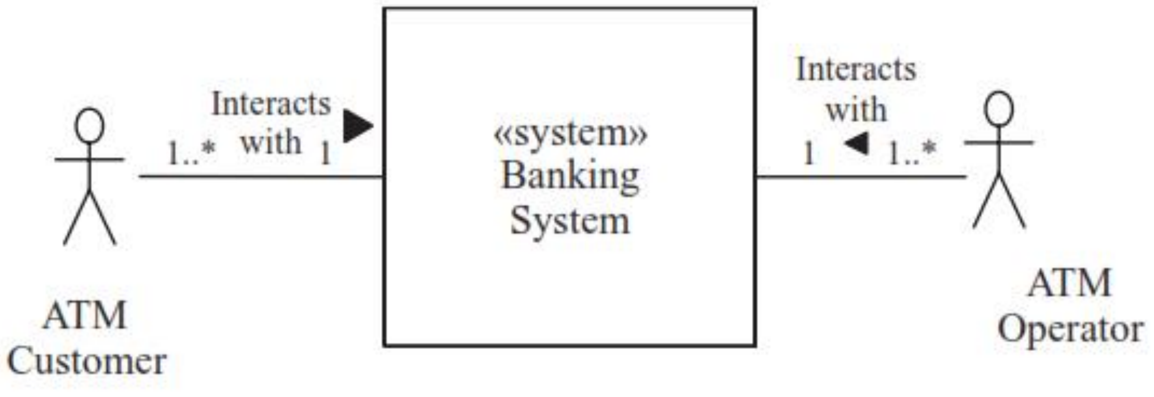
II.Static modelling of the system context(4)

- This is particularly useful in situations in which hardware/software tradeoffs need to be made
- In considering the total hardware/software system only users (i.e., human actors) and external systems are outside the system. I/O devices are part of the hardware of the system and therefore appear inside the total system.

Is child of

II.Static modelling of the system context(7)

As an example, consider the total hardware/software system for the Banking System. From a total hardware/software system perspective, the ATM Customer and ATM Operator actors.



Is child of

II.Static modelling

- ❑ **CATEGORIZATION OF CLASSES USING UML STEREOTYPES**

II.Static modelling

- CATEGORIZATION OF CLASSES USING UML STEREOTYPES

- **A category is a division in a system of classification**
- **According to COMET analysis method, we categorize classes in order to group together classes with similar characteristics whereas classification based on inheritance is an objective of object-oriented modeling.**

II.Static modelling

❑ CATEGORIZATION OF CLASSES USING UML STEREOTYPES(1)

- **Categorization is a decision to organize classes into certain groups because**

✓ **Most software systems have these kinds of classes**

✓ **And because categorizing classes in this way helps to better understand the system being developed.**

Is child of

II.Static modelling

❑ CATEGORIZATION OF CLASSES USING UML STEREOTYPES(2)

- **Stereotypes are used to distinguish among the various kinds of classes.**