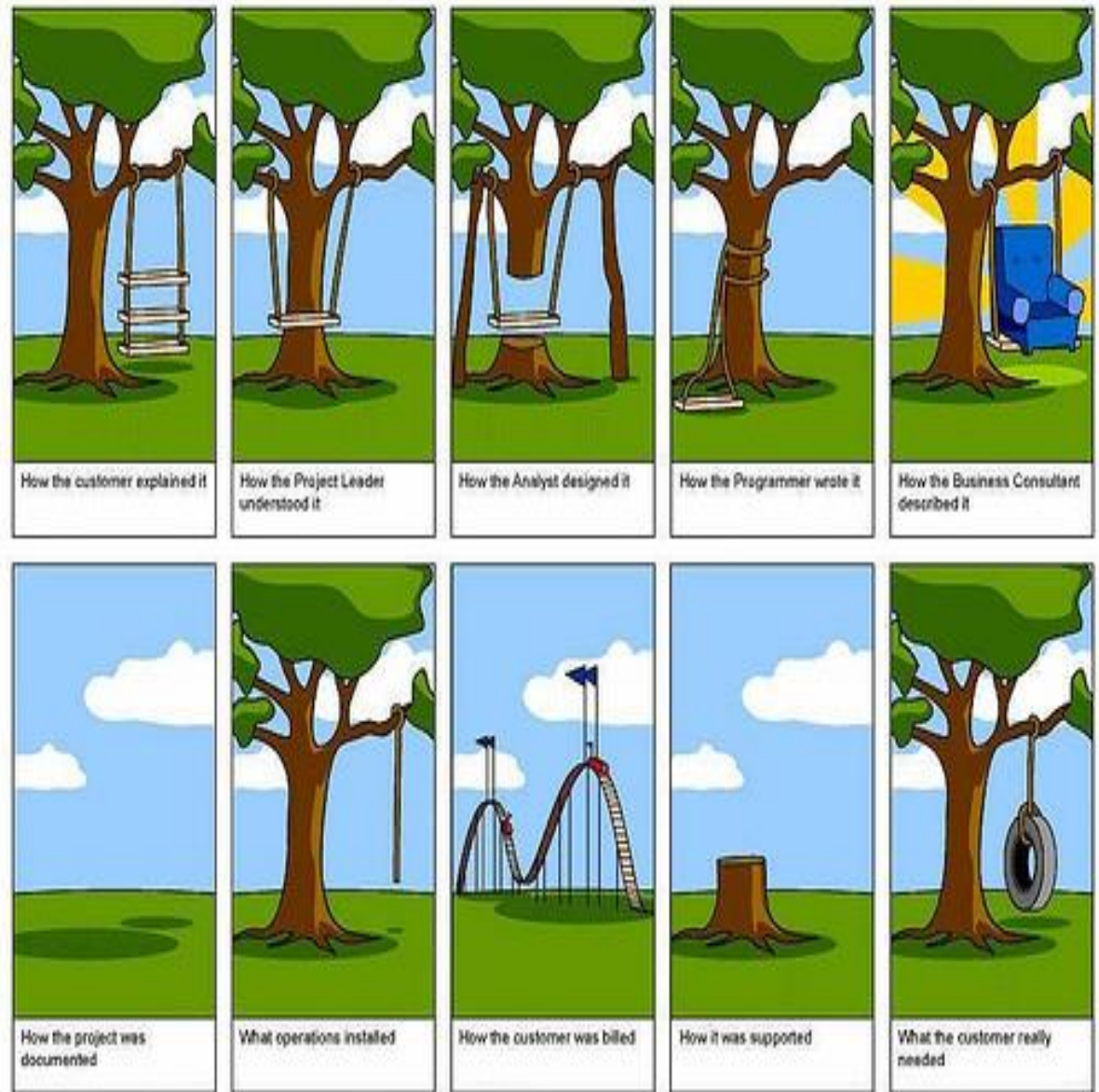


# **Class Diagram (OOAD)**

Nasrul, S.Pd.I, S.Kom, M.Kom  
nasrul@nurulfikri.ac.id

# MILESTONE

1. Use Case Model
  1. Use Case
  2. SSD – System Sequence Diagram
2. Domain Model
3. Design Model
  1. Sequence Diagram
  2. Collaboration Diagram
  - 3. Class Diagram**
  4. Deployment Diagram



# Class Diagram

---

- Diagram statis yang digunakan untuk memodelkan ***static view*** dari sebuah system
- Diagram yang menampilkan struktur desain sistem pada level class dan interfaces
- Diagram menggambarkan atribut dan operasi yang terjadi pada class dan interfaces
- Diagram menggambarkan relasi yang terjadi antara object
- Diagram menggambarkan asosiasi/keterhubungan antar class ( struktur organisasi class)

# Class Diagram

---

- Diagram konstruksi dari sistem aplikasi yang dikembangkan dengan pendekatan Object Oriented Programming
- Komponen Class Diagram :
  - Struktur / Property
    - Ciri pembeda antar object berupa property
  - Behavior
    - Perilaku atau kegiatan yang mampu dilakukan object

# Contoh : RekeningBank

---

- Struktur
  1. No Rekening
  2. Saldo
  3. Pemilik Rekening
- Behaviour
  1. Cek saldo
  2. Tarik ( withdraw)
  3. Setor ( deposit )
  4. Transfer

# Contoh : Pintu

---

- Struktur
  1. Lebar
  2. Tinggi
  3. Warna
- Behaviour
  1. Buka
  2. Tutup

# Class

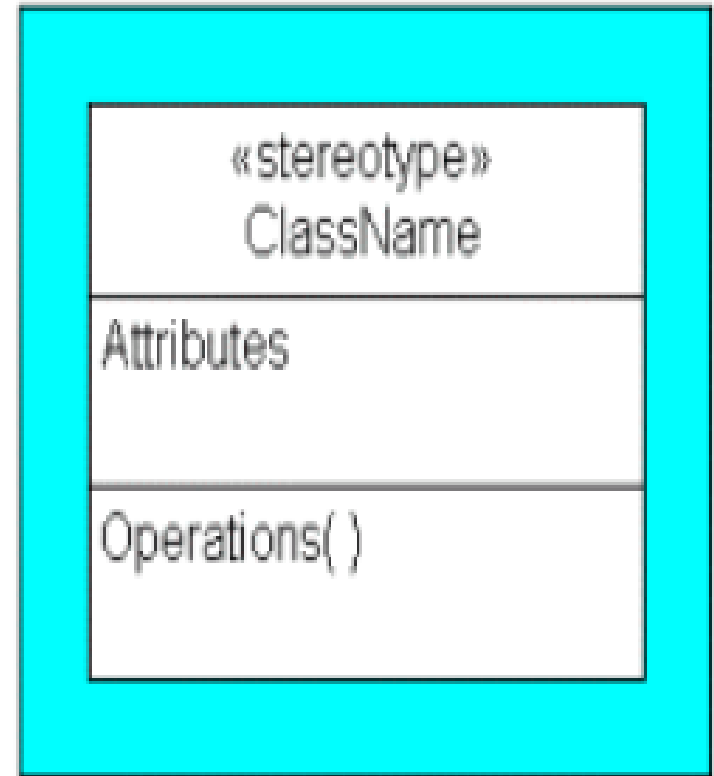
---

- ❑ Class : **Blue Print** dari Object
  - Gambar/Desain Rumah, Resep Masakan
- ❑ Member Class :
  - ❑ Property : norek, saldo, nasabah
  - ❑ Method : cetak, deposit, withdraw, transfer

# UML : Notasi Class

---

- ❑ Nama Class
- ❑ Attributes / property
- ❑ Operations / Method





## Contoh : Class Diagram

AccountBank
-nomor -saldo -nasabah
+cetak() +deposit() +withdraw() +transfer()

Flight
flightNumber : Integer departureTime : Date flightDuration : Minutes
delayFlight ( numberOfMinutes : int ) : Date getArrivalTime ( ) : Date

# Visibility

---

Visibilitas akses member class : attribute dan operation

1. Public : dapat diakses dari dalam class dan luar class
2. Private : hanya bisa diakses dari dalam class
3. Protected : hanya bisa diakses dari dalam class dan class turunannya

<i>+ public</i>
<i>- private</i>
<i># protected</i>

Class Name
- attribute
- attribute
+ operation + operation + operation

# Attribut

---

- Notasi dari atribut
  - visibility name: type multiplicity = default {property-string}
- Contoh
  - - name: String [1] = "Untitled" {readOnly}
  - + berarti public, - berarti private, # berarti protected
  - "Untitled" adalah nilai yang diberikan secara default jika tidak ditentukan saat objek dibuat
  - {readOnly} adalah properti tambahan dari atribut, dimana disini berarti tidak bisa dimodifikasi

# Operations

---

- Notasi dari operations
  - visibility name (parameter-list) : return-type {property-string}
- Parameter pada parameter-list dinotasikan seperti pada atribut
  - direction name: type = default value
  - Direction bisa berupa: in, out, atau inout
- Contoh
  - + balanceOn (date: Date) : Money

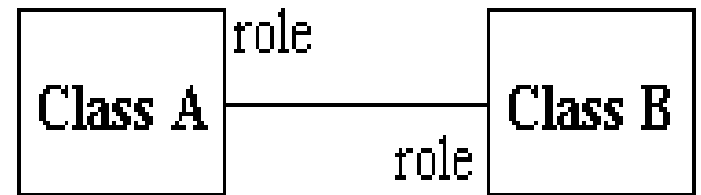
## Contoh : AccountBank

---

AccountBank
-nomor : string -saldo : double +nasabah : object
+cetak() +deposit(in uang : double) +withdraw(in uang : double) +transfer(in account_tujuan : object, in uang : double) #updateSaldo(in uang : double)

# Assosiasi : Keterhubungan

- Menggambarkan relasi statis yang terjadi antara class
- Ditandai dengan garis lurus di atasnya nama relasi
- Diawal dan akhir garis berisi peran/role dari class

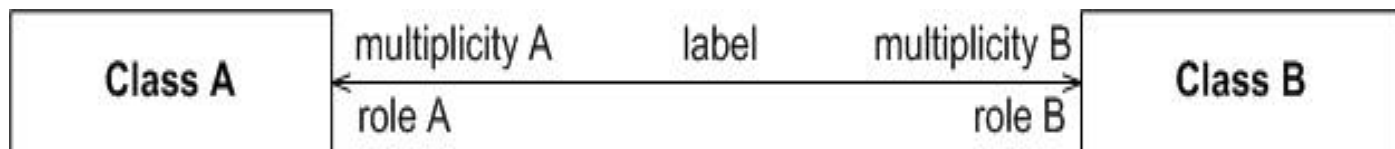


# Assosiasi : Multiplicity

- Relasi antar class dapat digambarkan dengan notasi multi

Table 1. Multiplicity Indicators.

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One or more
n	Only $n$ (where $n > 1$ )
0..n	Zero to $n$ (where $n > 1$ )
1..n	One to $n$ (where $n > 1$ )



# Multiplicity

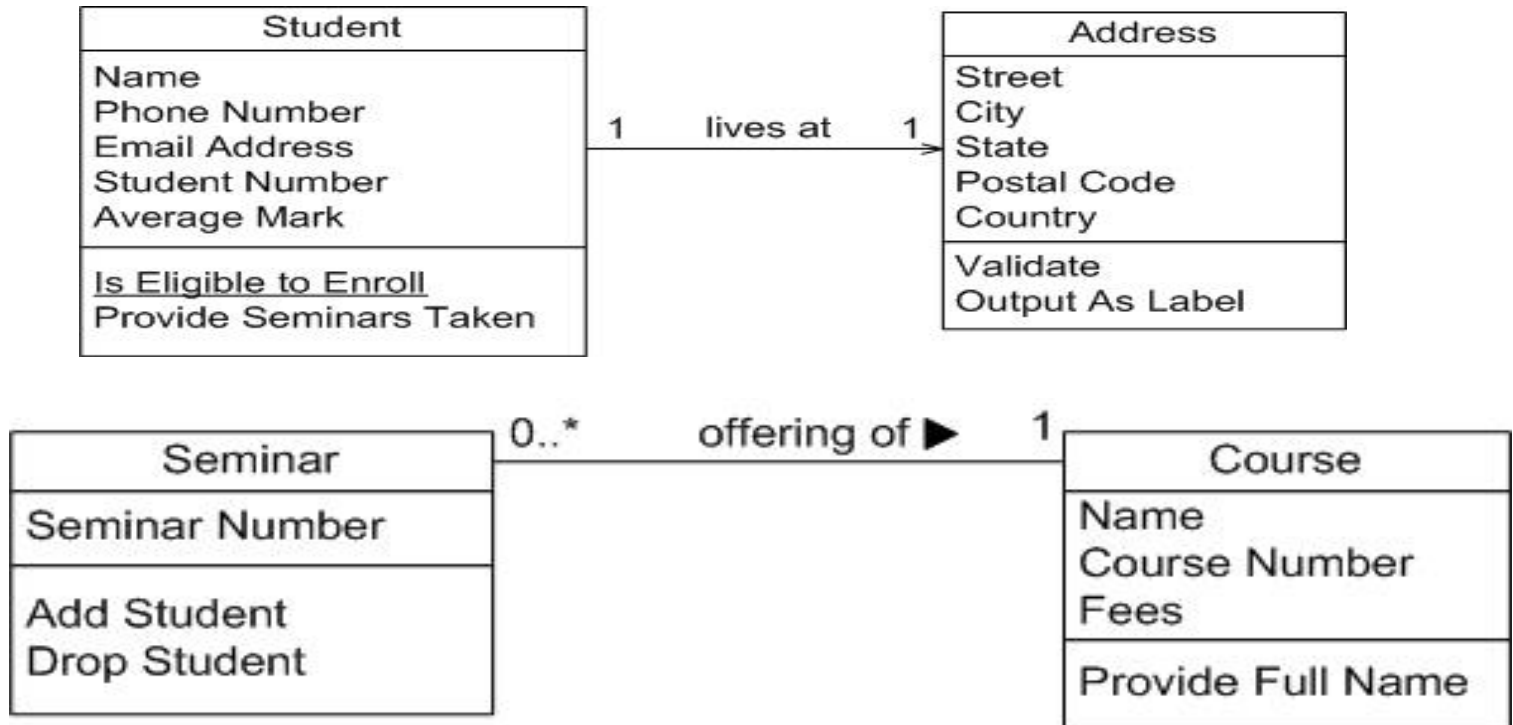
---

- Indikasi berapa banyak objek yang bisa mengisi properti
  - 1 (pasti 1)
  - 0..1 (0 atau 1)
  - \* (Tidak ada batasan, bisa 0, 1, ..., n)
- Biasanya didefinisikan batas bawah dan atas, kecuali untuk yang pasti bernilai 1
- Mirip dengan konsep one-to-one dan one-to-many pada relational database



# Assosiasi : Multiplicity

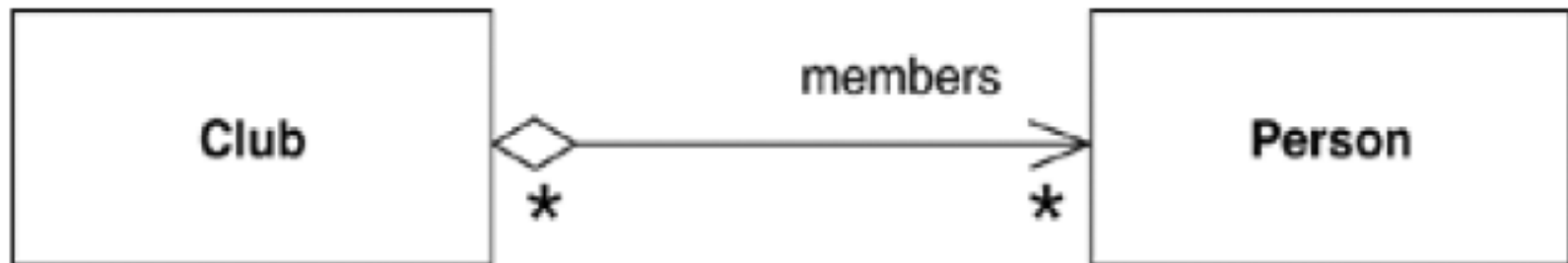
- Relasi antar class dapat digambarkan dengan notasi multiplicity



# Aggregation

---

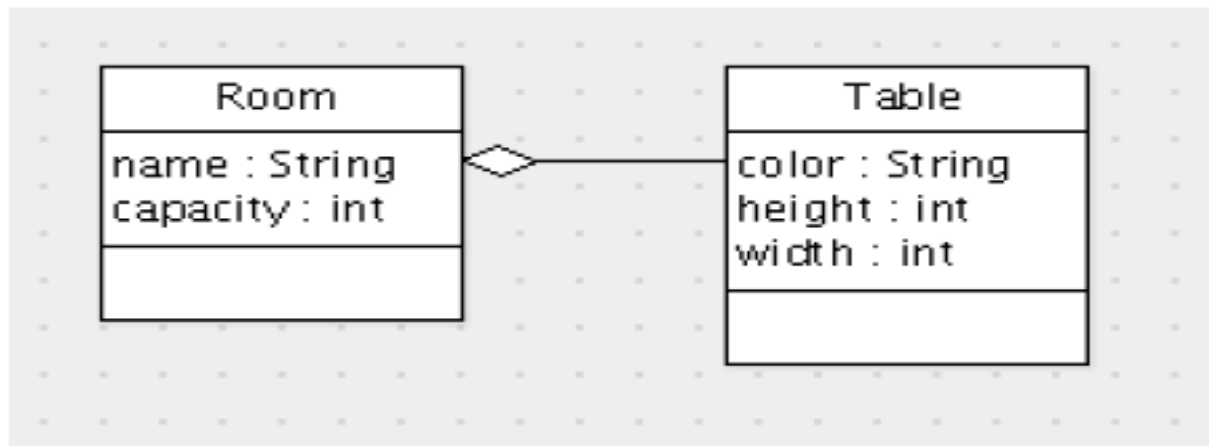
- 'has a' relationship
  - Klub memiliki banyak anggota
  - Orang bisa memiliki makna tersendiri tanpa kehadiran sebuah klub
- Dinotasikan dengan diamond “kosong”
- Jika dipisah, tidak merubah makna



# Contoh : Aggregation

---

- Sebuah ruangan memiliki meja dan kursi
- Tanpa kehadiran ruang, meja dan kursi bisa tetap ada



## Contoh : Aggregation Associaton

---

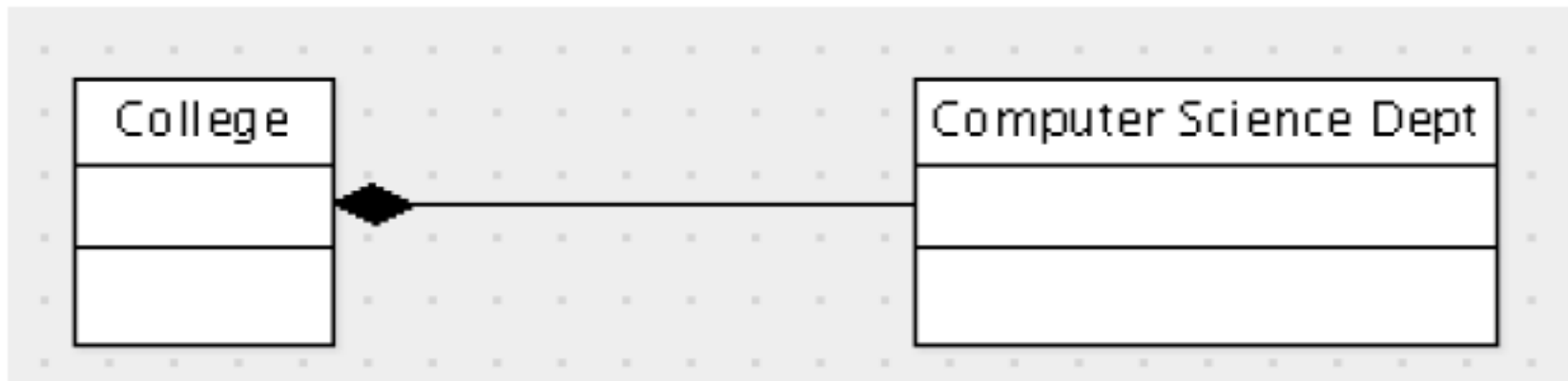
- Sebuah mobil memiliki roda empat



# Composition

---

- 'has a' or 'contains a' relationship (whole-part)
  - Kampus memiliki fakultas CS atau kampus terdiri dari fakultas CS (salah satunya)
  - Tanpa ada kampus, maka tidak ada fakultas CS



## Contoh: Aggregation Composition

---

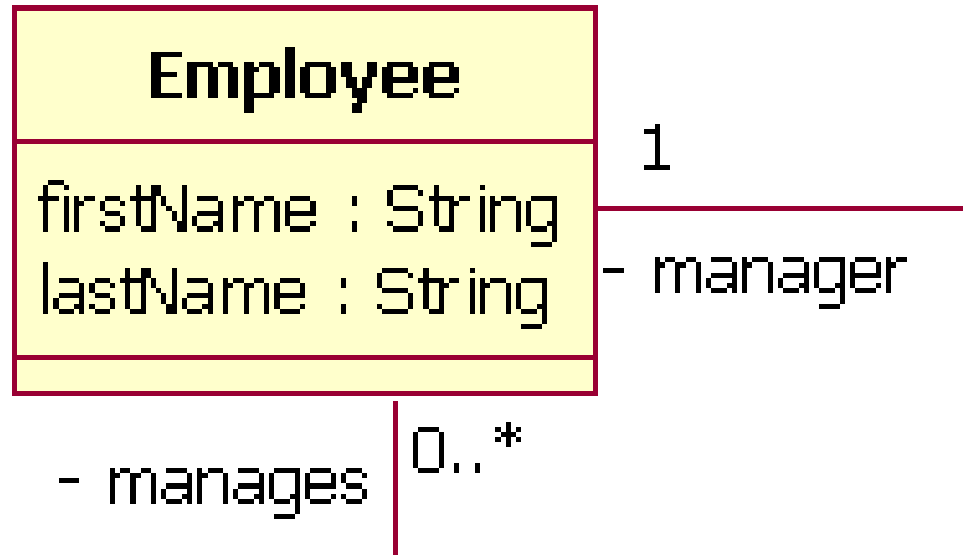
- Dalam sebuah perusahaan minimal ada satu atau lebih departemen



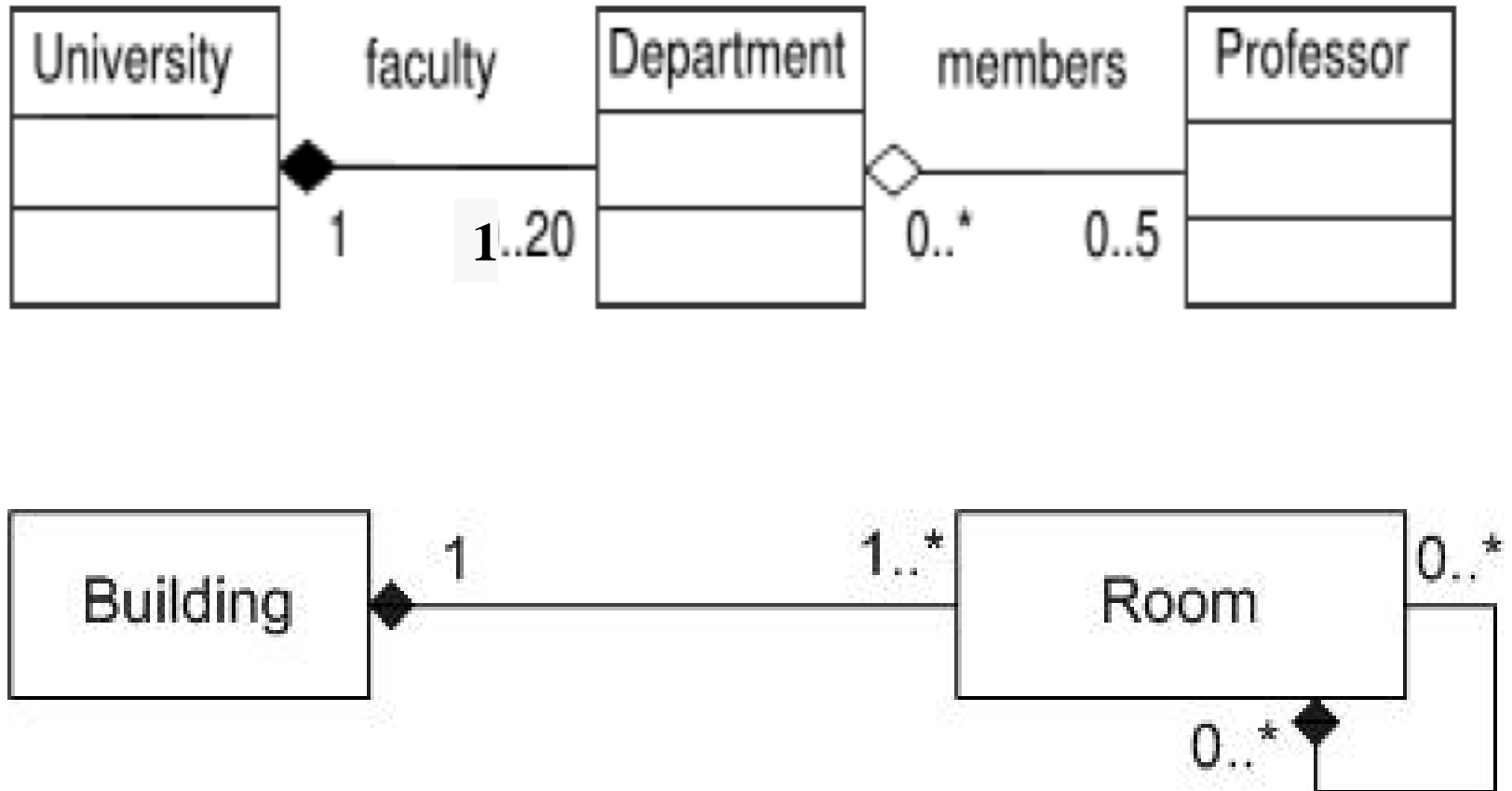
# Contoh: Reflexion Associaton

---

- Relasi antara pegawai dengan manager

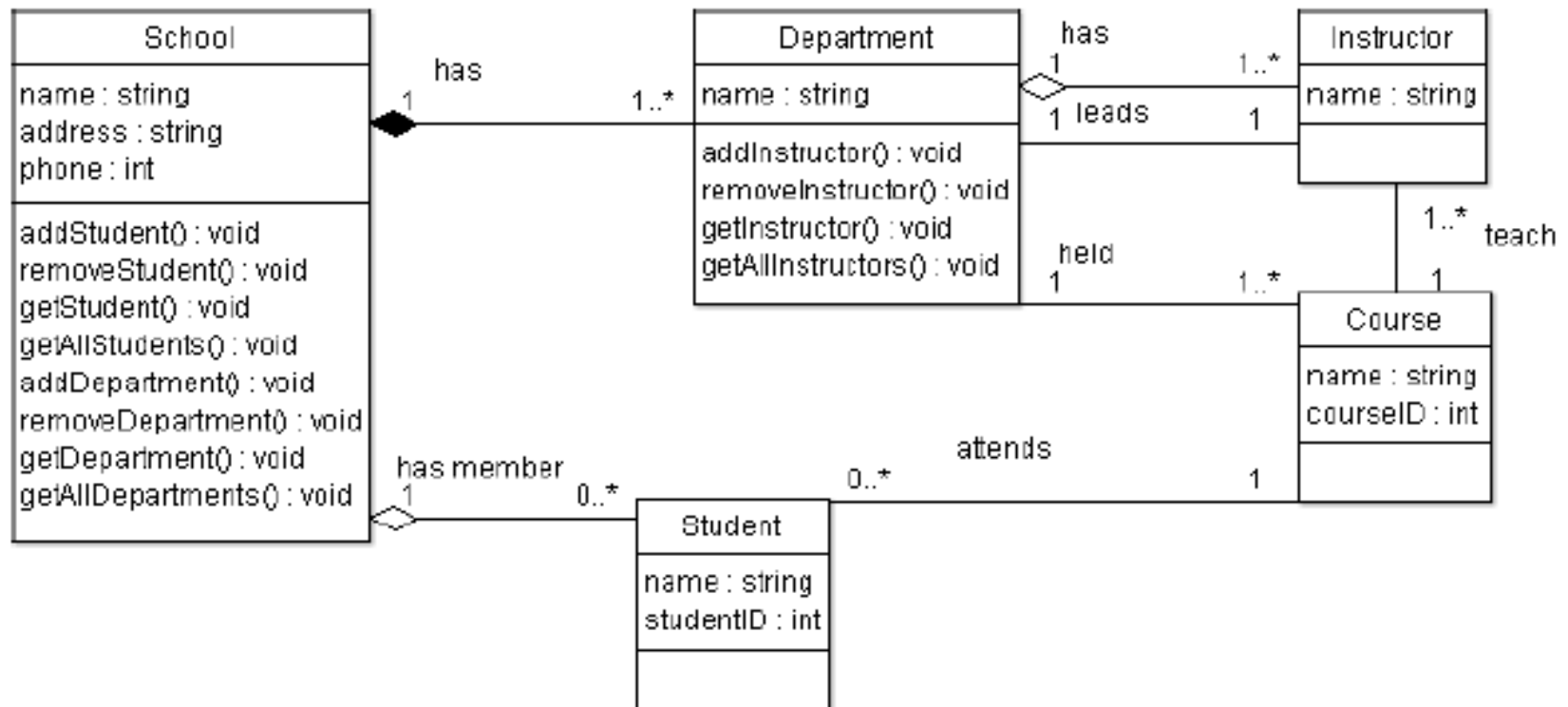


## Contoh : Composition

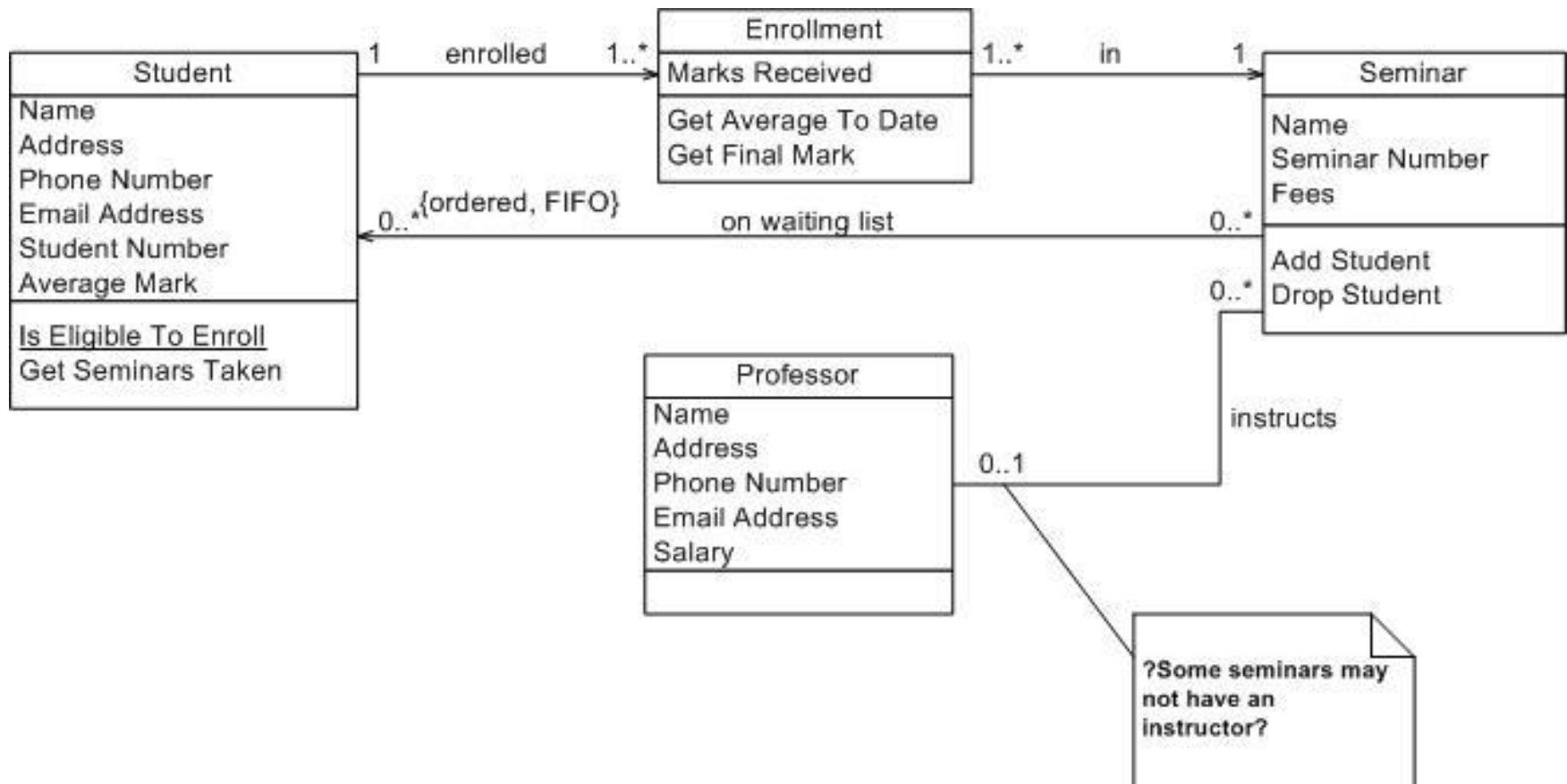




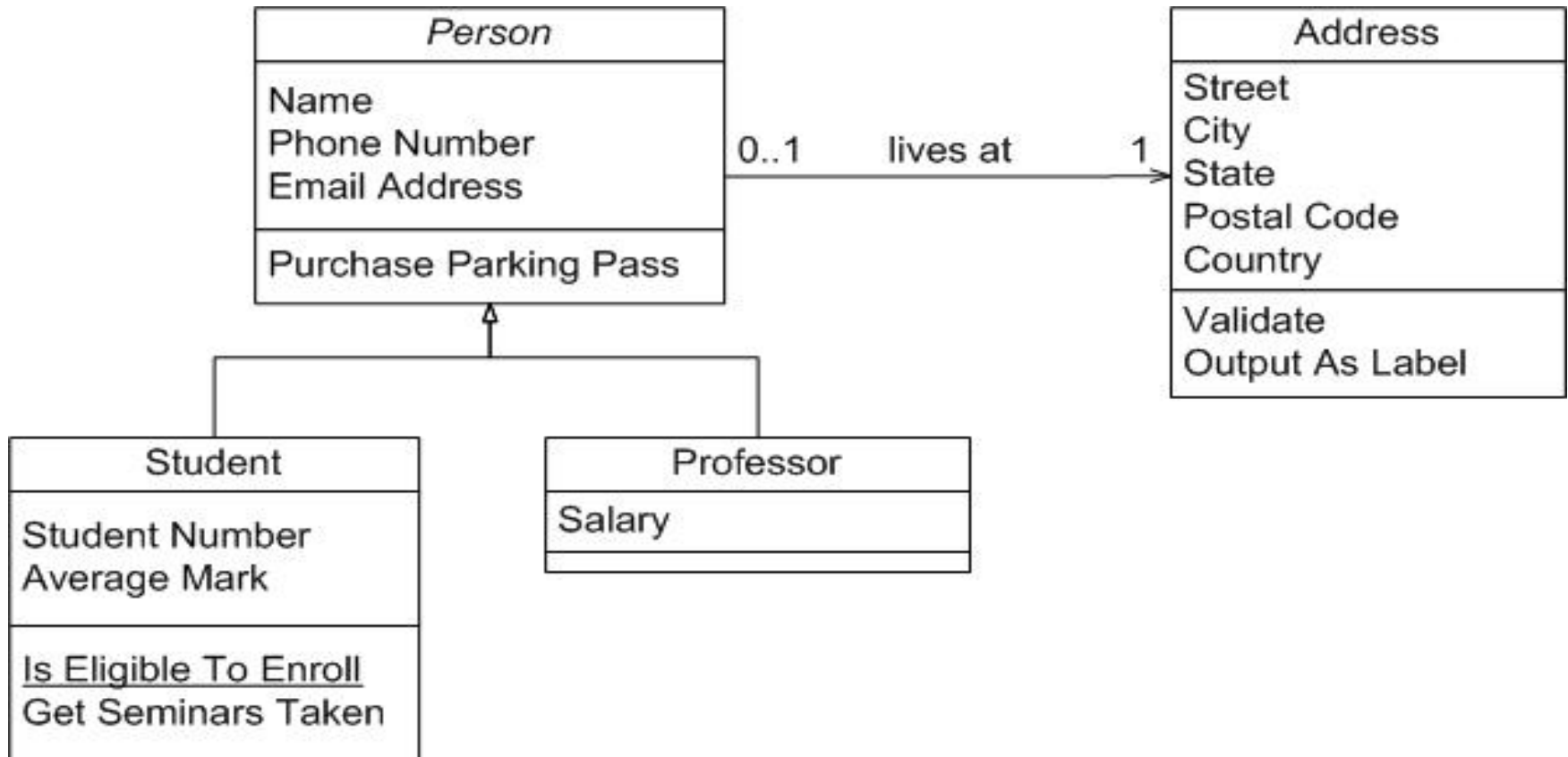
# Contoh : Multiplicity



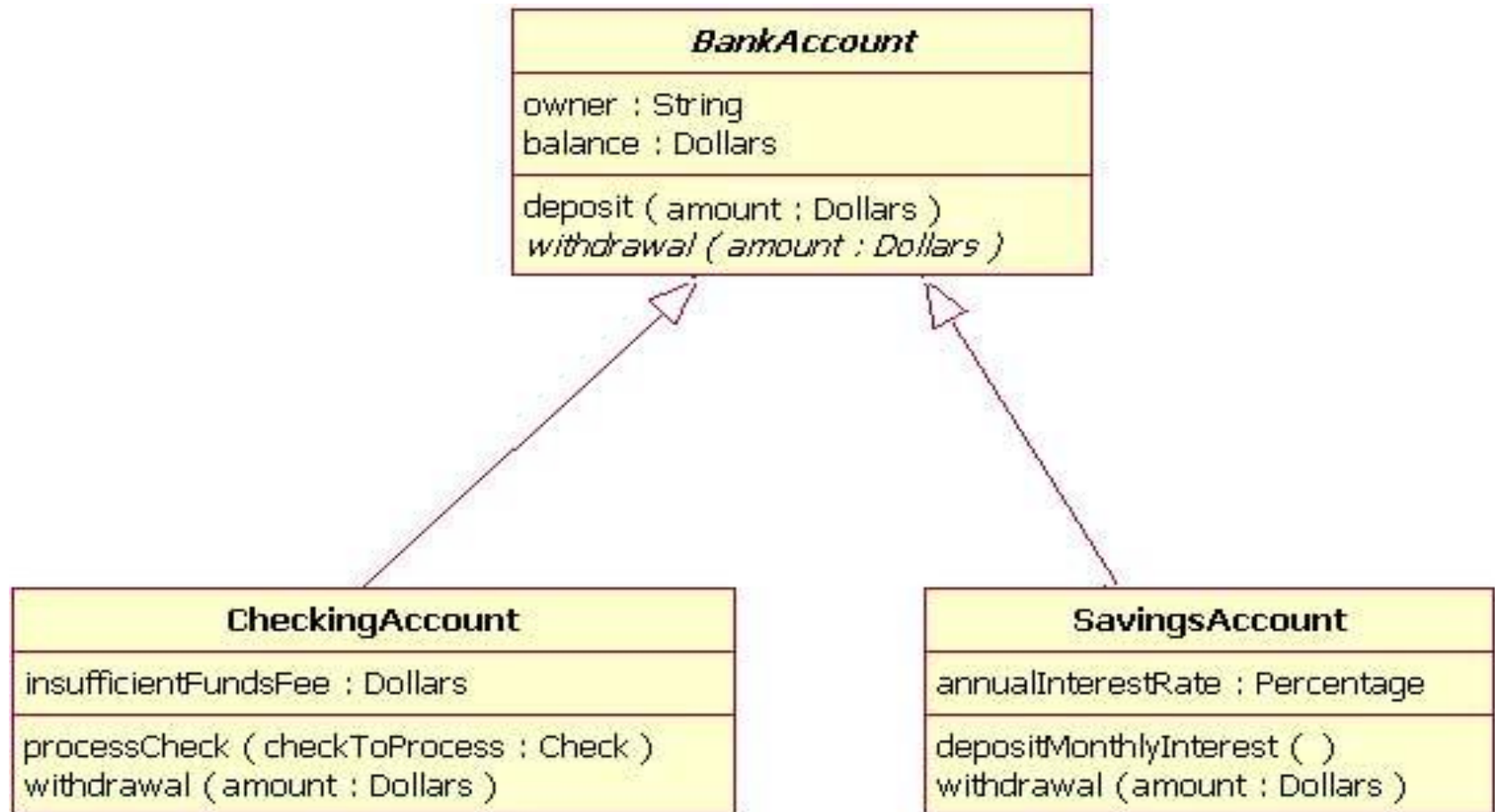
# Contoh: Multiplicity



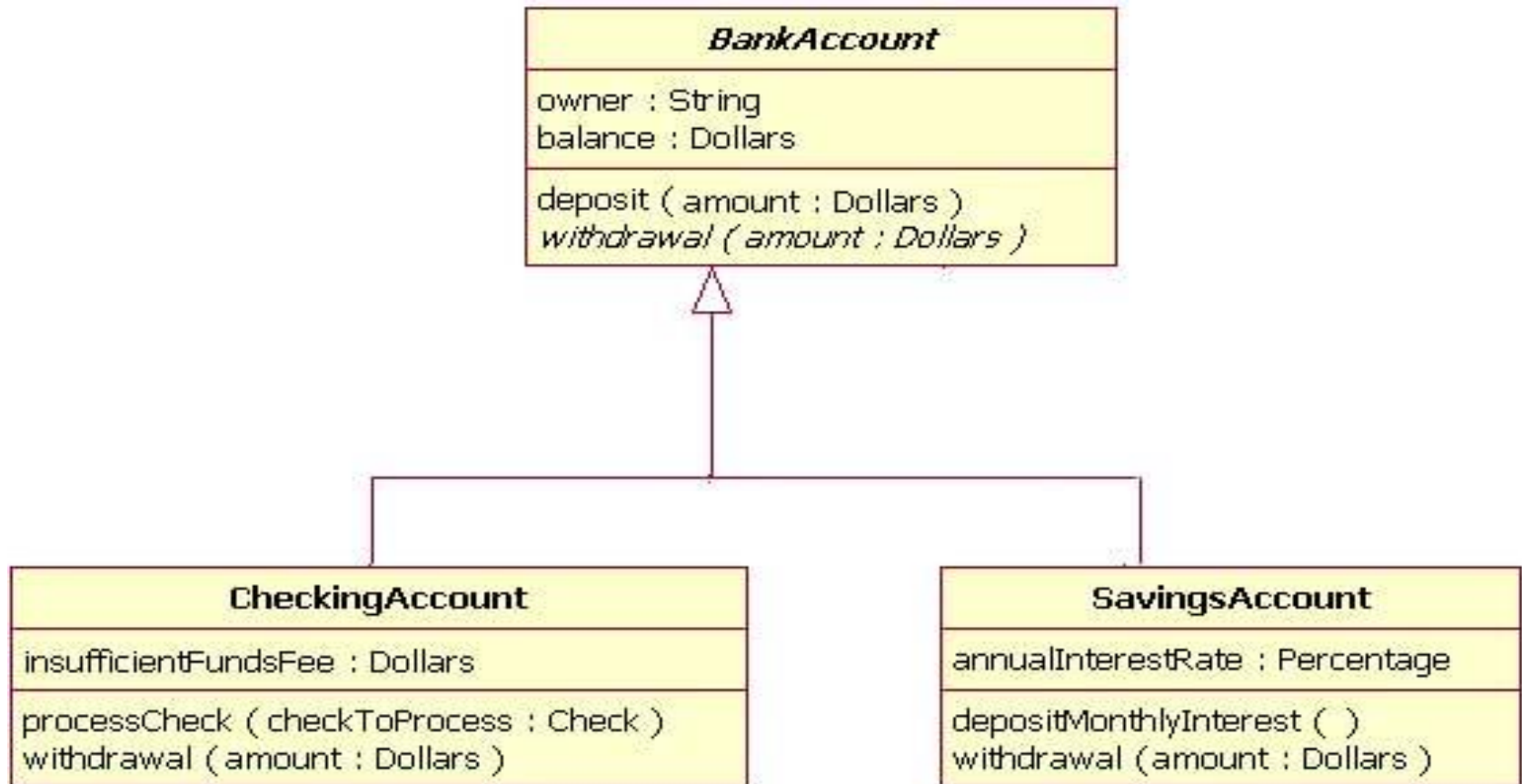
# Inheretance



# Contoh : Inheritance BankAccount

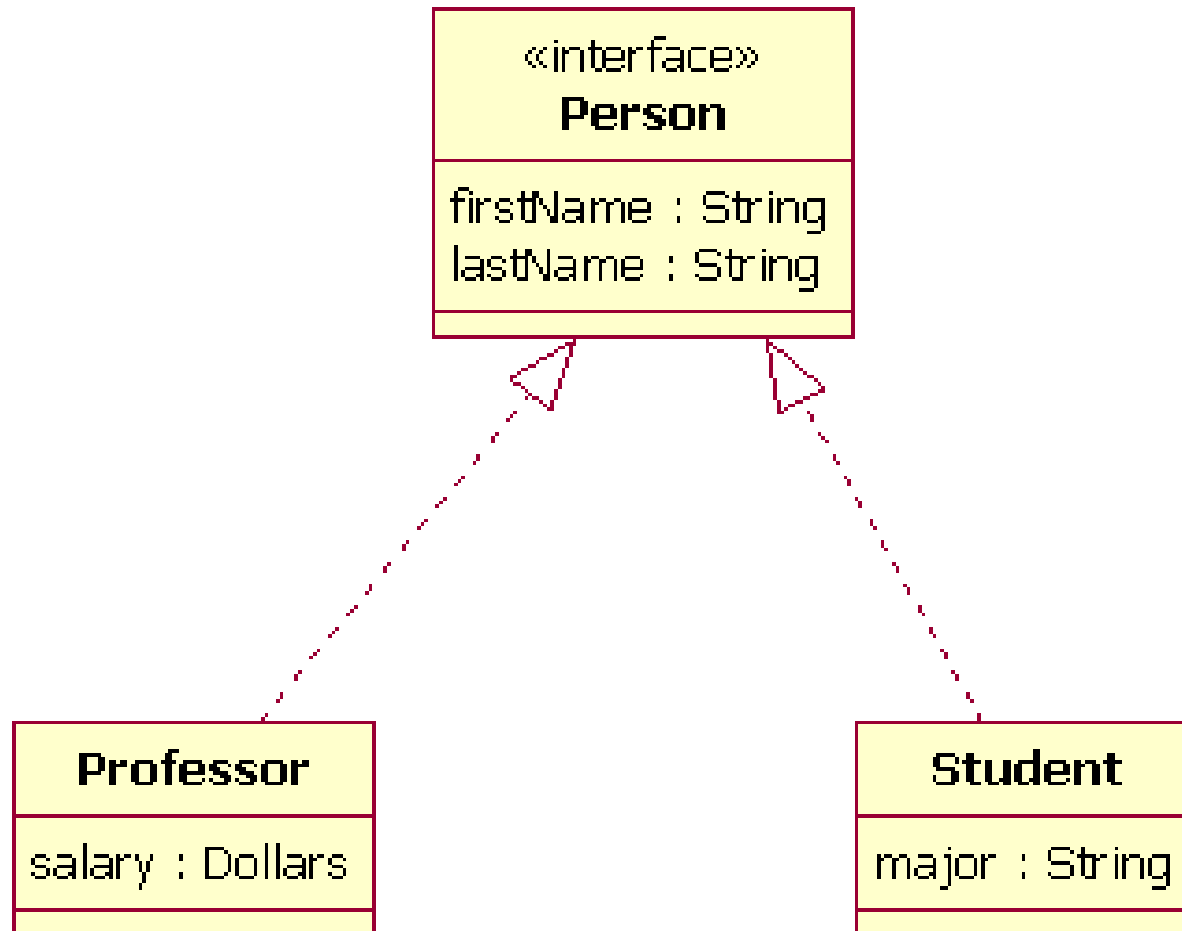


# Inheretance

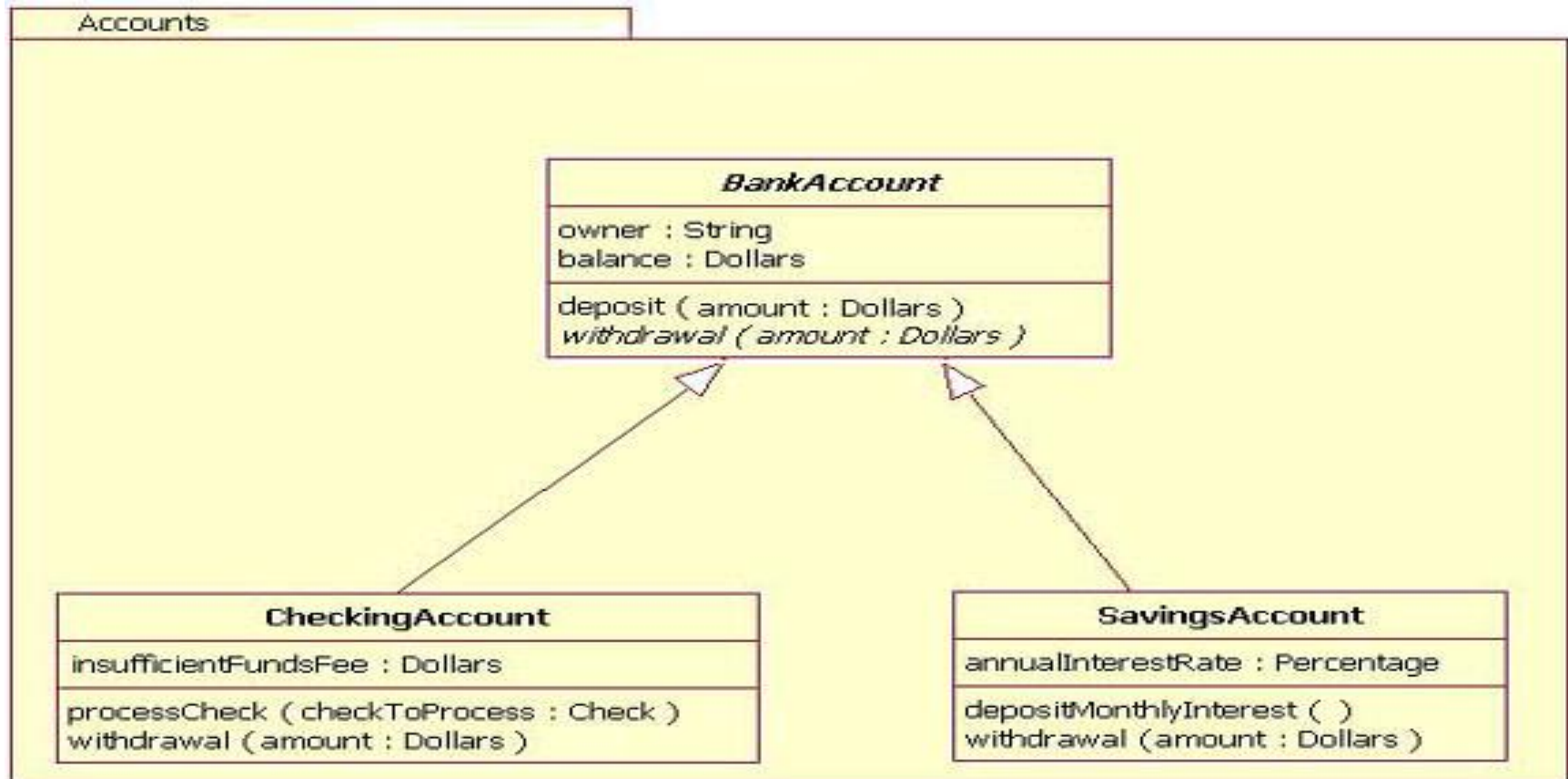


# Interfaces

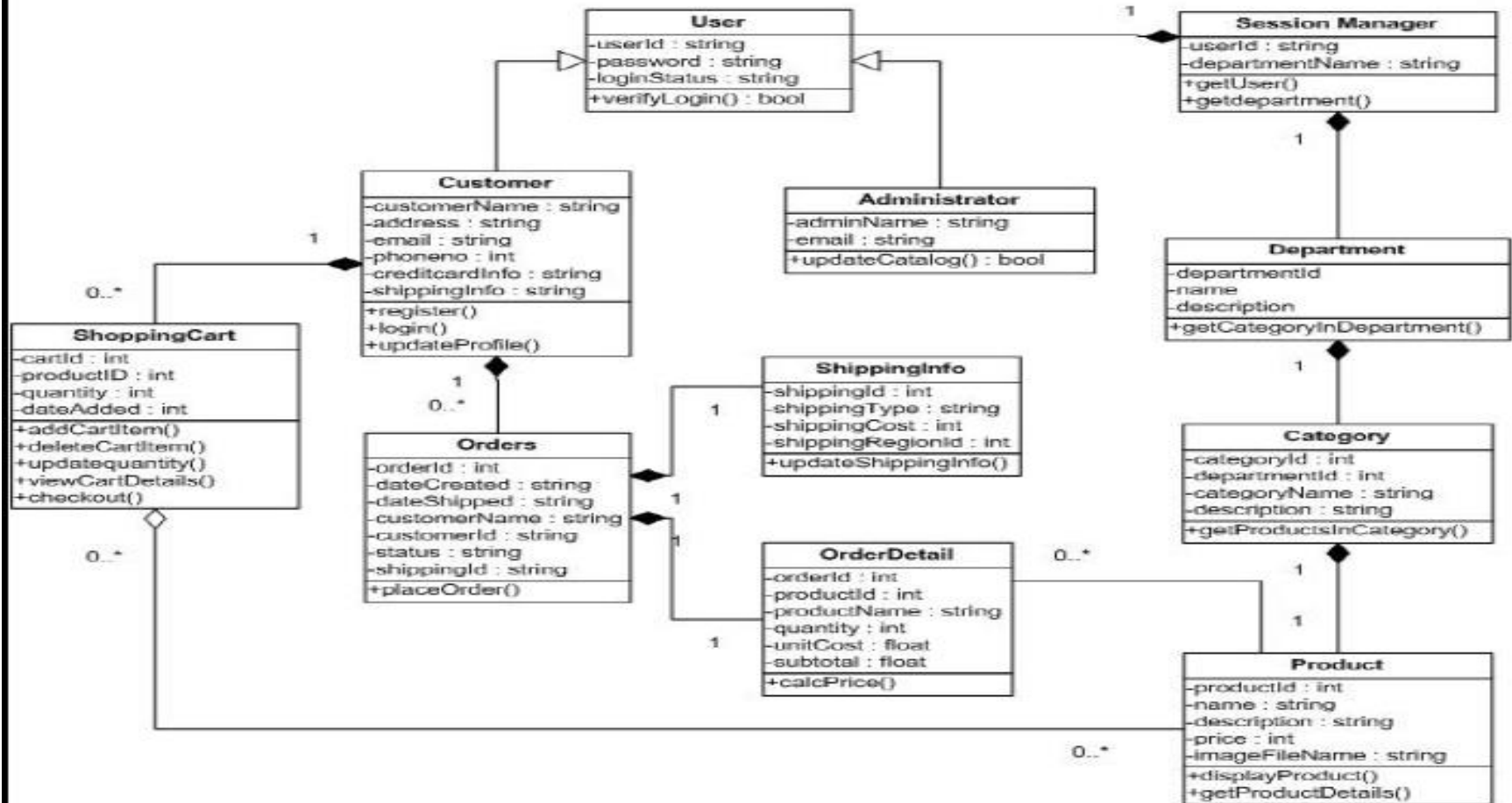
---



# Package



# Contoh : Class Diagram





## Contoh : Class Diagram

