ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

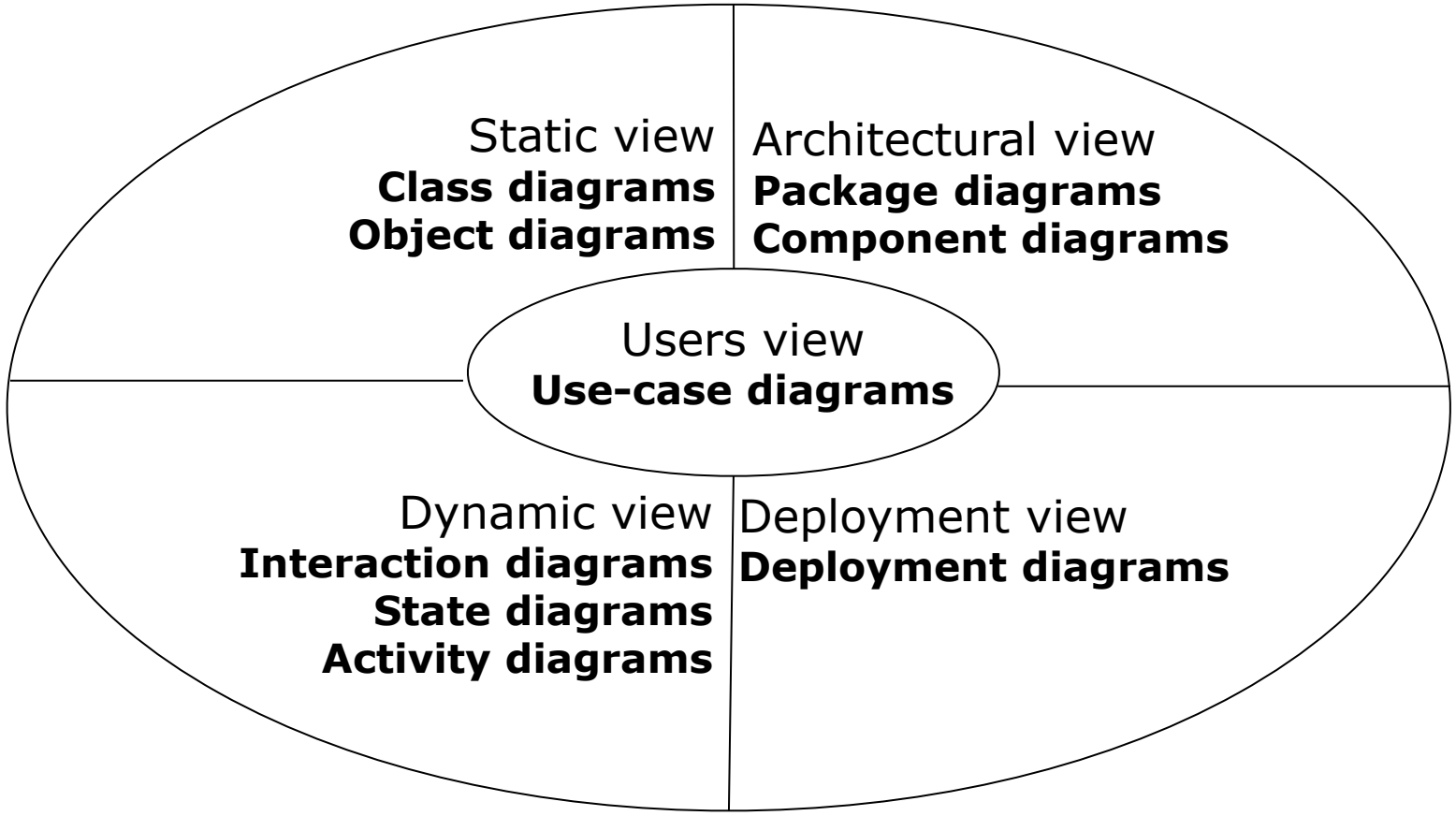
PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG PHẦN MỀM

Lê Viết Trương
Khoa Khoa học máy tính

Mô hình hóa cấu trúc tĩnh

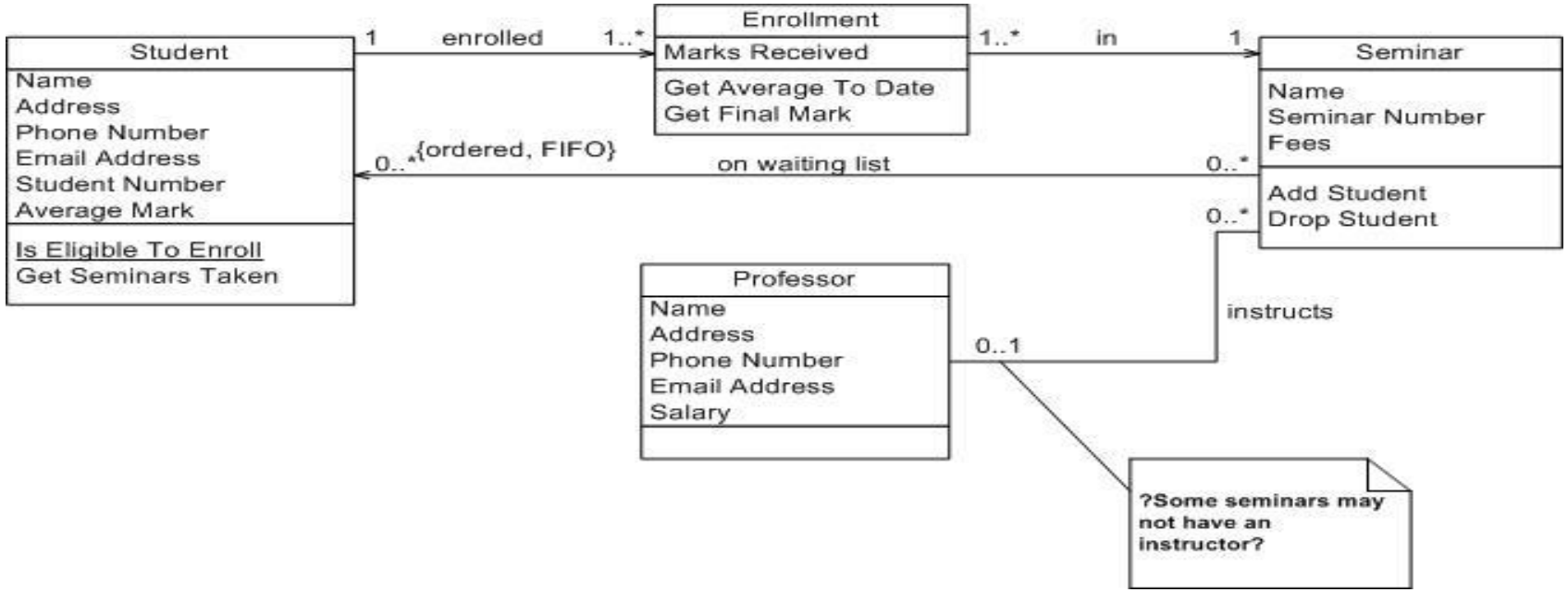
- Biểu đồ lớp
- Biểu đồ đối tượng

Khung nhìn


Static view
Class diagrams
Object diagrams
Architectural view
Package diagrams
Component diagrams
Users view
Use-case diagrams
Dynamic view
Interaction diagrams
State diagrams
Activity diagrams
Deployment view
Deployment diagrams

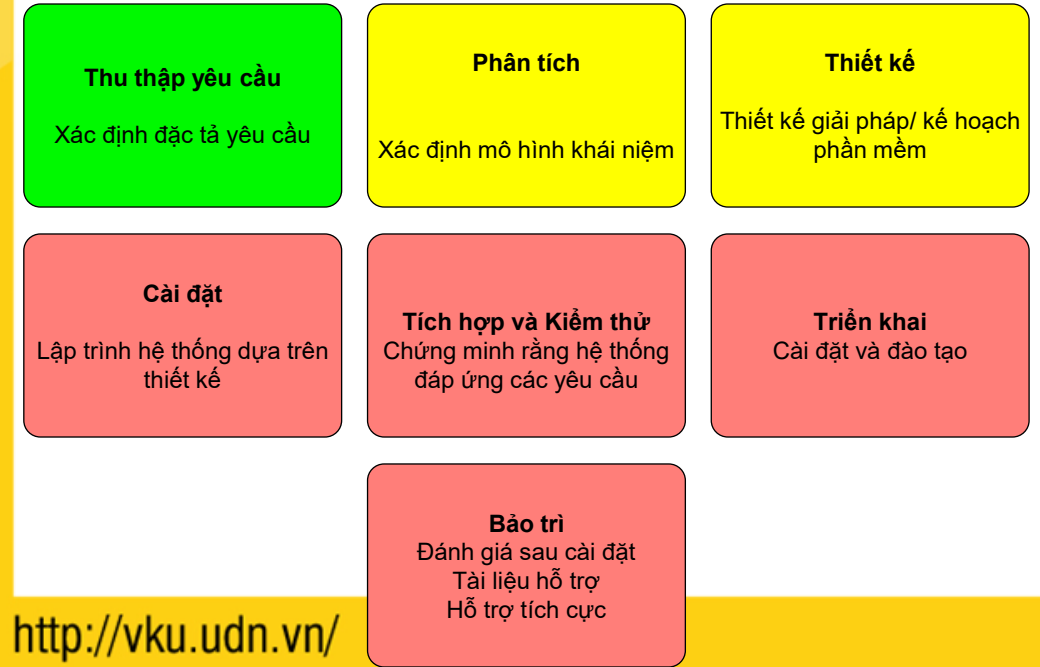
Biểu đồ lớp

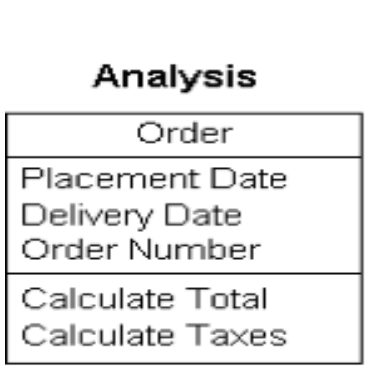
- Biểu đồ lớp
  - Bao gồm một tập các lớp, giao diện và các quan hệ của chúng
  - Diễn tả **khung nhìn tĩnh** của hệ thống
  - Có thể xây dựng **khung** của hệ thống
- Mô hình hóa biểu đồ lớp là bước thiết yếu trong thiết kế hướng đối tượng

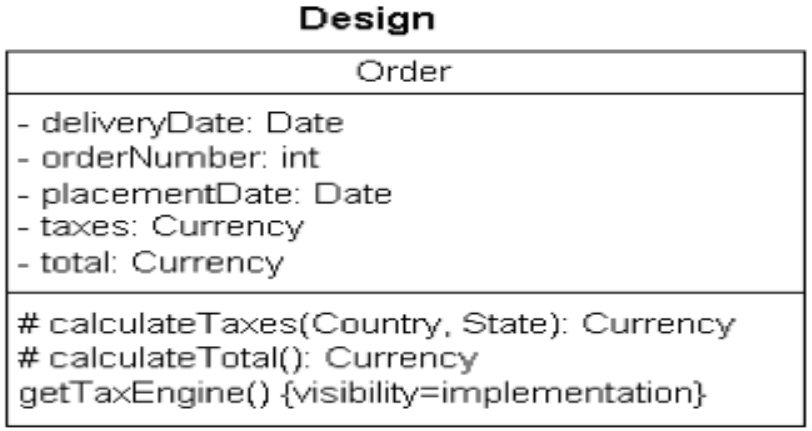


Biểu đồ lớp phân tích và Biểu đồ lớp thiết kế

- Có hai kiểu biểu đồ lớp
  - Biểu đồ lớp khái niệm / phân tích (mô hình miền)
    - được phát triển trong giai đoạn phân tích
    - mô tả hệ thống từ "quan điểm của người dùng"
  - Biểu đồ lớp thiết kế
    - được phát triển trong giai đoạn thiết kế
    - mô tả hệ thống từ "quan điểm của người phát triển phần mềm"

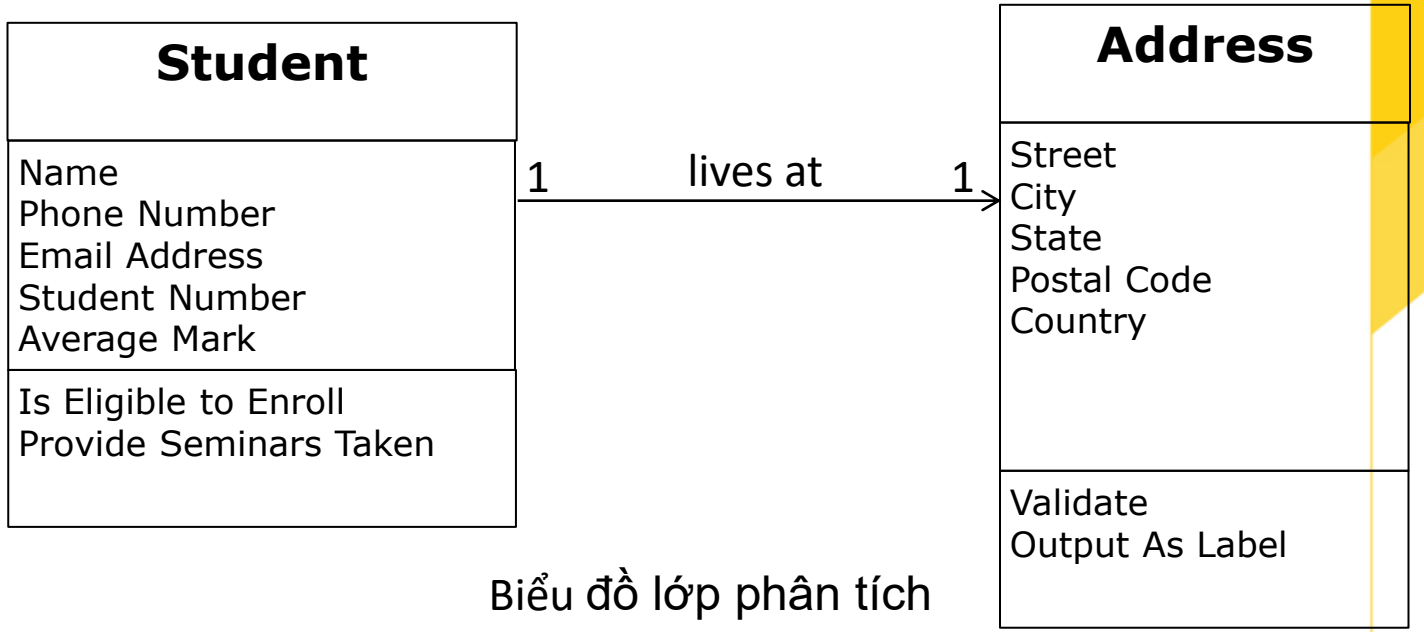






Biểu đồ lớp phân tích

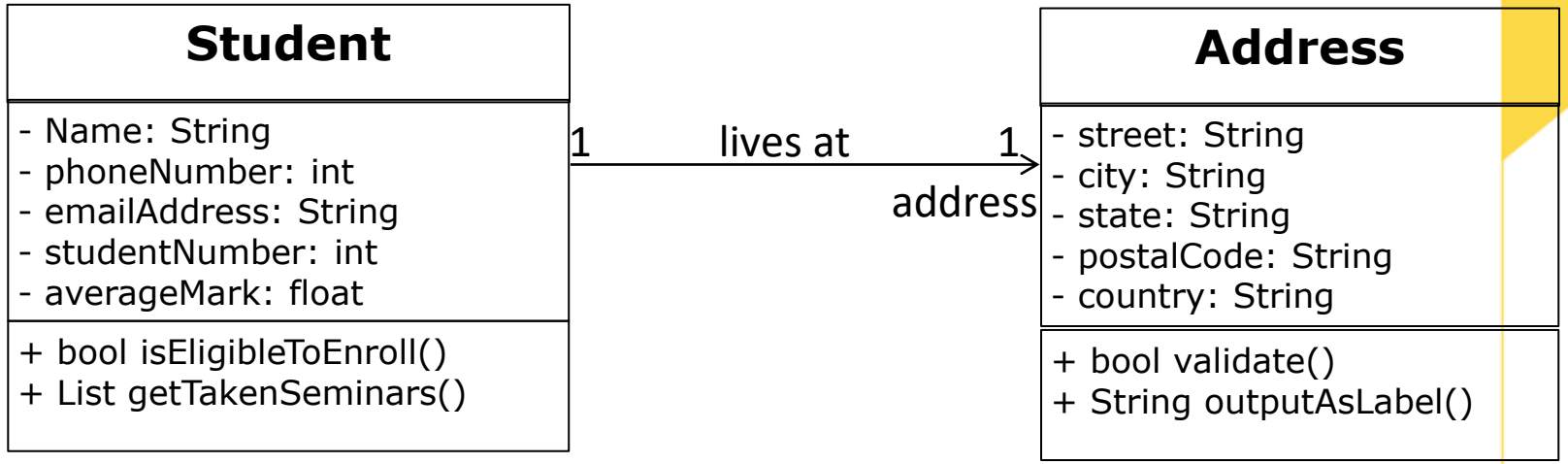
- Biểu đồ lớp khái niệm / phân tích (mô hình miền)
  - được xây dựng trong giai đoạn phân tích
  - nắm bắt các khái niệm được người dùng / khách hàng / bên liên quan công nhận không chứa thông tin về cách hệ thống phần mềm sẽ được triển khai



Biểu đồ lớp phân tích

Biểu đồ lớp thiết kế

- Biểu đồ lớp thiết kế
  - được xây dựng trong giai đoạn thiết kế
  - một phiên bản chi tiết của biểu đồ lớp phân tích
    - một lớp phân tích có thể tương ứng với một số lớp thiết kế
  - chứa thông tin về cách hệ thống phần mềm nên được triển khai
    - khả năng hiển thị của thuộc tính và phương thức
    - tên thuộc tính và phương thức phù hợp với ngôn ngữ lập trình đích



Biểu đồ lớp thiết kế (cài đặt Java)

Biểu đồ lớp phân tích và Biểu đồ lớp thiết kế

Thu thập yêu cầu
Xác định đặc tả yêu cầu

Phân tích
Xác định mô hình khái niệm

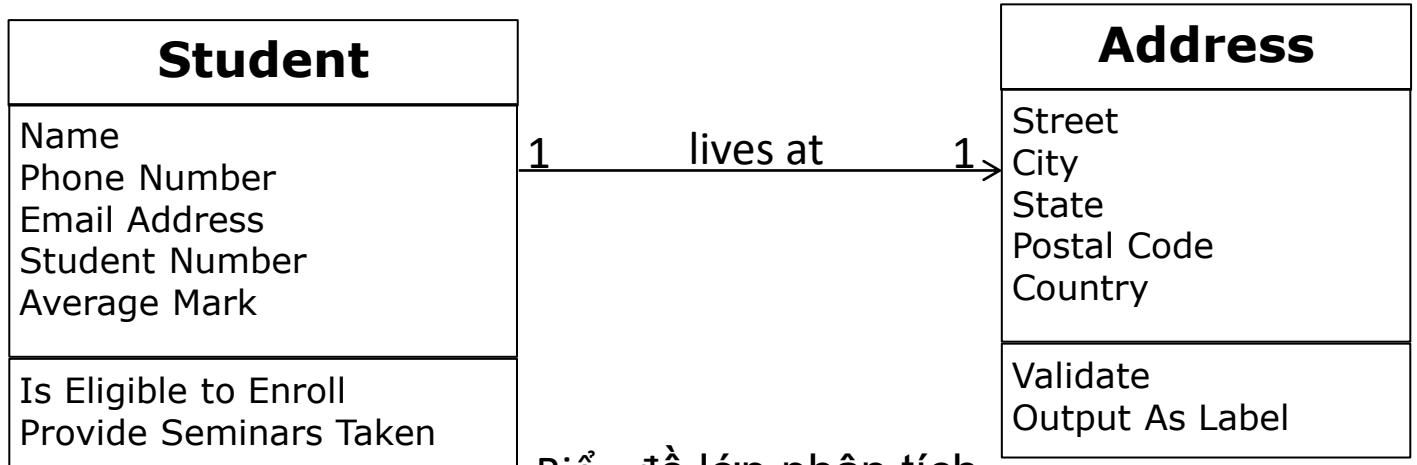
Thiết kế
Thiết kế giải pháp/ kế hoạch phần mềm

Cài đặt
Lập trình hệ thống dựa trên thiết kế

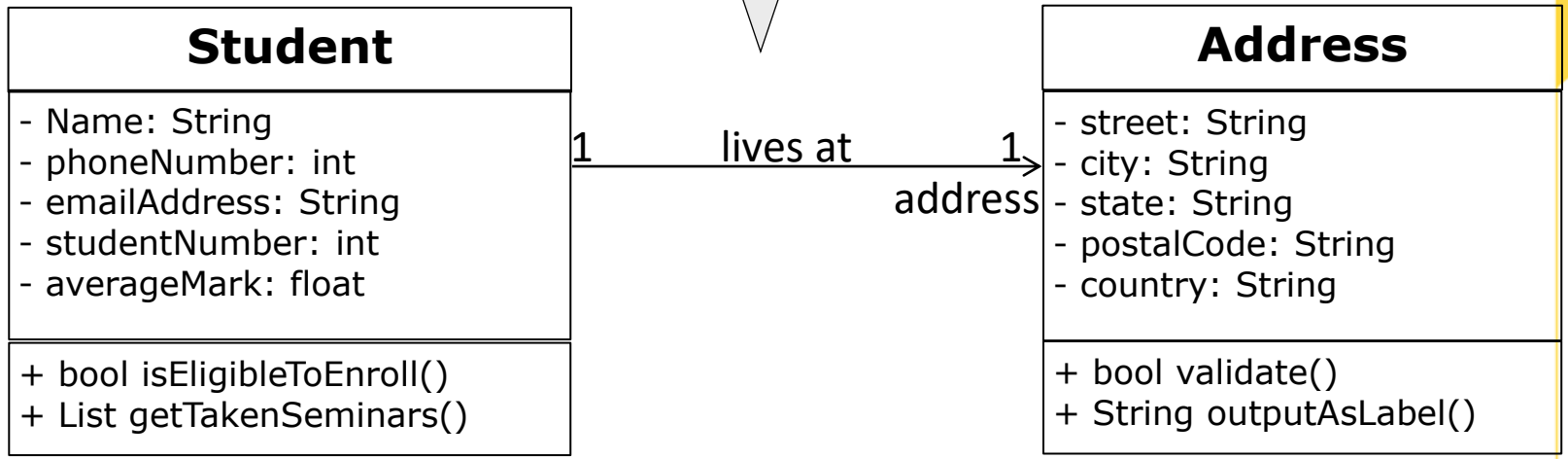
Tích hợp và Kiểm thử
Chứng minh rằng hệ thống đáp ứng các yêu cầu

Triển khai
Cài đặt và đào tạo

Bảo trì
Đánh giá sau cài đặt
Tài liệu hỗ trợ
Hỗ trợ tích cực



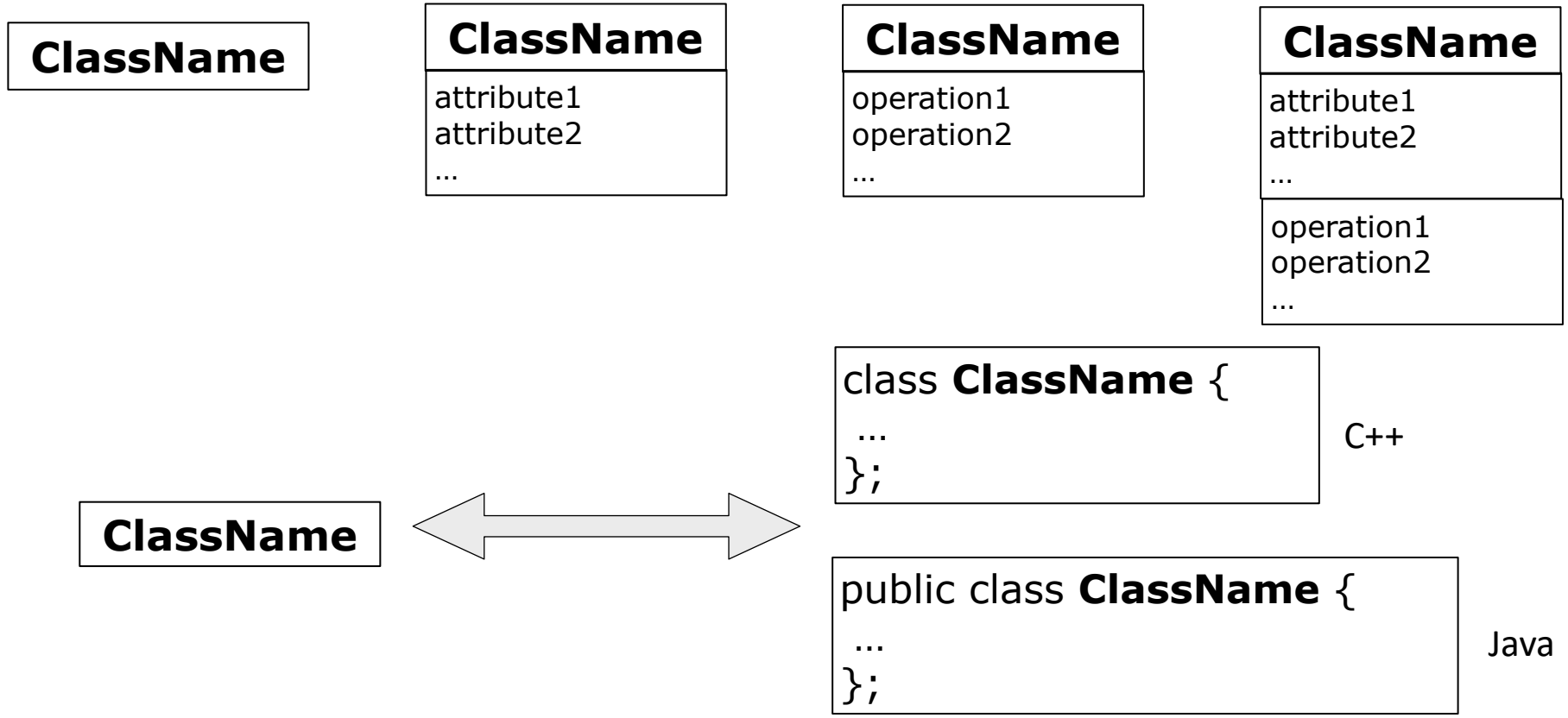
Biểu đồ lớp phân tích



Biểu đồ lớp thiết kế (cài đặt Java)

Lớp

- Lớp UML
  - đại diện cho khái niệm lớp hoặc giao diện của ngôn ngữ lập trình hướng đối tượng
  - bao gồm một tập các thuộc tính và phương thức
  - có thể được biểu diễn bằng đồ họa dưới một số dạng



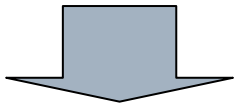
Thuộc tính

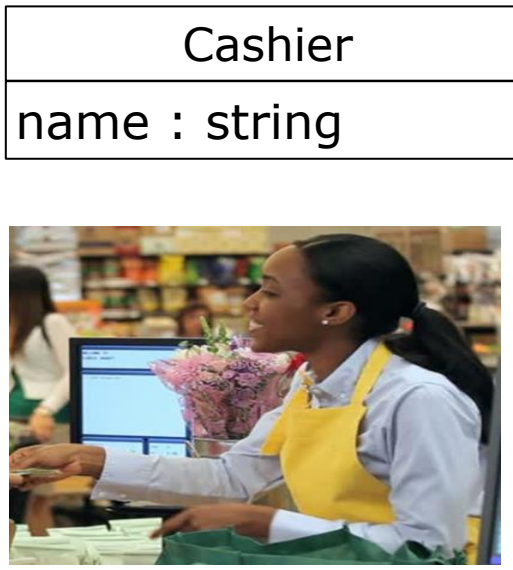
- Các thuộc tính đại diện cho dữ liệu cần thiết của các thể hiện lớp
- Các thuộc tính có thể có
  - Một kiểu
    - Kiểu đơn giản
      - number : integer
      - length : double
      - text : string
    - Kiểu phức tạp
      - center : Point
      - date : Data
  - Một giá trị mặc định
    - number : integer = 10
  - Một danh sách giá trị có thể
    - color : Color = red {red, blue, purple, yellow}

Person
name : string firstName : string dateOfBirth : Date nbChildren : integer = 0 married : Boolean = false profession : string = « not defined »

Thuộc tính

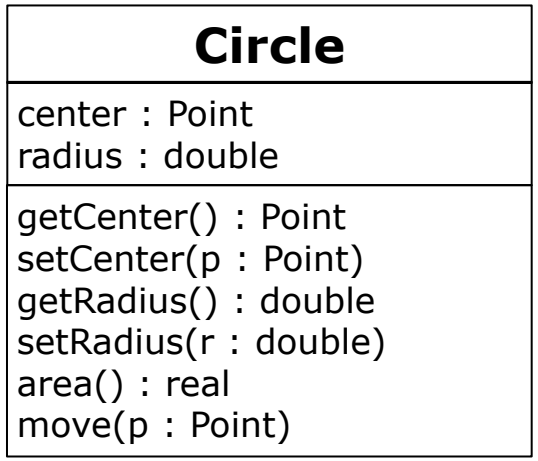
- Thuộc tính chỉ đại diện cho dữ liệu liên quan đến lớp có thuộc tính này
- Ví dụ

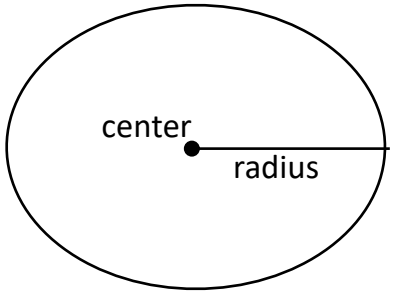




Phương thức

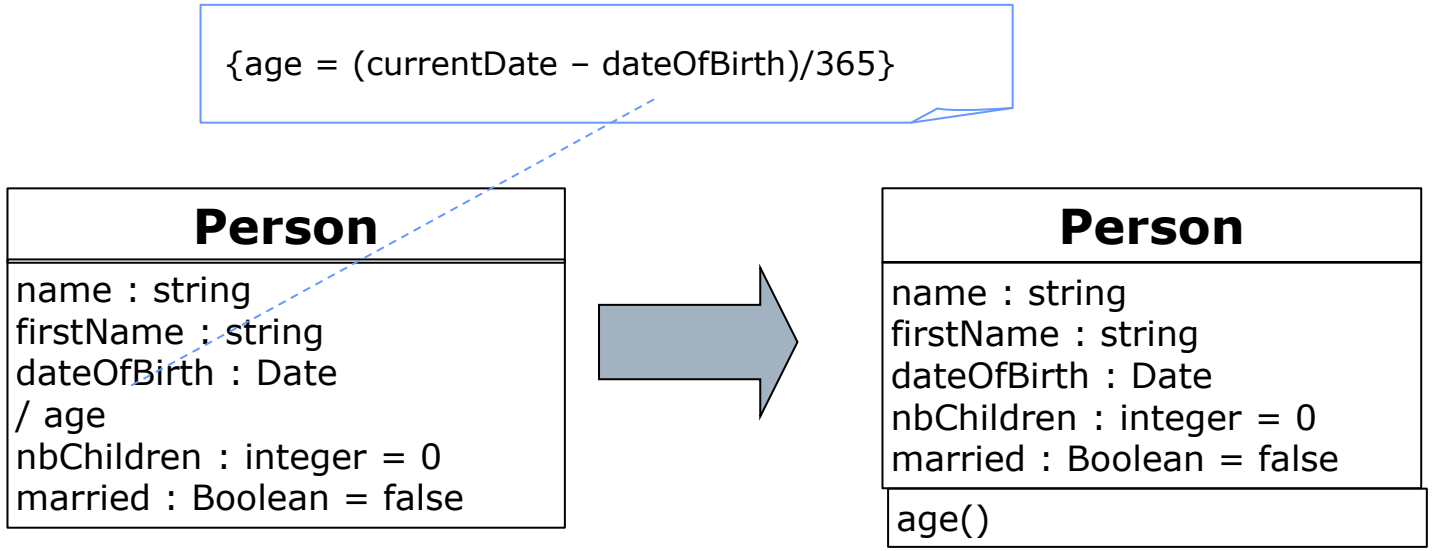
- Phương thức đại diện cho các **hành vi** của thể hiện của lớp
- Hành vi của một lớp bao gồm
  - Các **getters** và **setters** thao tác dữ liệu của các thể hiện lớp
  - Một số nhiệm vụ nhất định gắn với **trách nhiệm** của lớp
- Phương thức có thể có
  - Một tên
    - area, calculate, …
  - Một kiểu trả về
    - area() : double
  - Các tham số với kiểu
    - move(p : Point)





Thuộc tính suy diễn

- Các thuộc tính có thể được suy diễn từ các thuộc tính khác
- Ví dụ: age/tuổi của một người có thể được tính từ dateOfBirth/ngày sinh

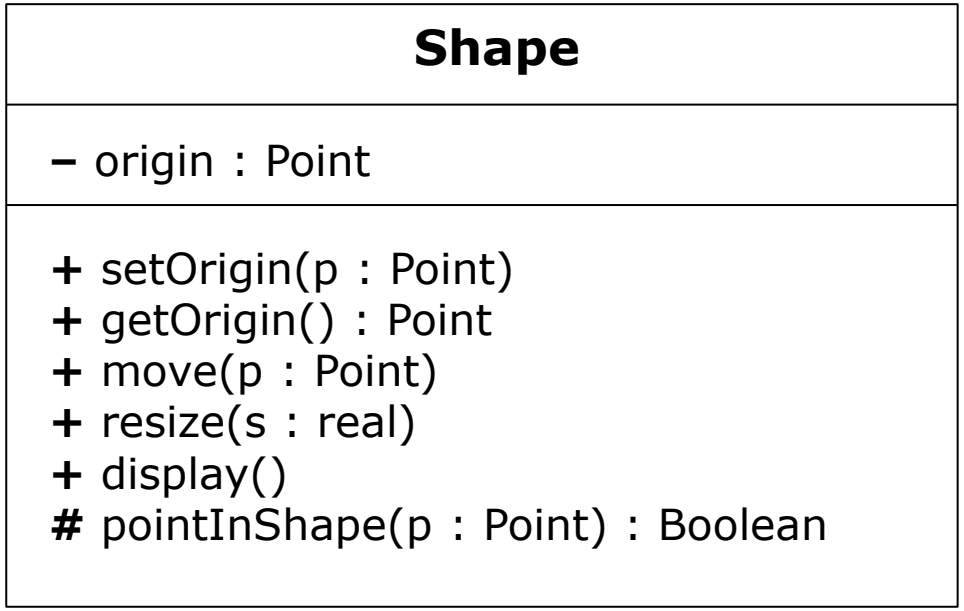


Thiết kế mức cao

Thiết kế chi tiết

Hiển thị

- Các thuộc tính và phương thức có khả năng hiển thị
  - Public
    - có thể nhìn thấy bên ngoài lớp
    - Ký hiệu “ + ”
  - Protected
    - chỉ hiển thị cho các đối tượng của cùng một lớp và các đối tượng của các lớp con
    - Ký hiệu “ # “
  - Private
    - chỉ hiển thị cho các đối tượng của lớp
    - Ký hiệu “ - “



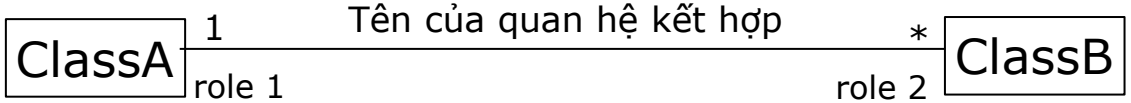
Các kiểu quan hệ

Mối quan hệ giữa các lớp:

- Quan hệ kết hợp/Association
  - Mối quan hệ ngữ nghĩa giữa các lớp
- Quan hệ tổng quát hóa/Generalization
  - Một lớp có thể kế thừa một hoặc nhiều lớp
- Quan hệ kết nhập/Aggregation
  - Một quan hệ kết hợp biểu diễn một lớp là một phần của lớp khác
- Quan hệ hợp thành/Composition
  - Một dạng đặc biệt của quan hệ kết nhập
- Quan hệ phụ thuộc/Dependency
  - Biểu diễn sự phụ thuộc giữa các lớp

Quan hệ kết hợp

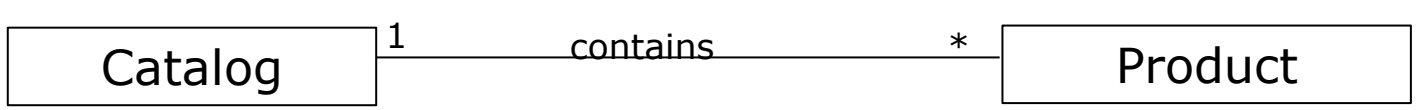
- Một kết hợp
  - được sử dụng để hiển thị cách hai lớp được liên kết với nhau
  - thể hiện kết nối ngữ nghĩa hai chiều giữa các lớp
  - là sự trừu tượng hóa các liên kết giữa các thể hiện của các lớp
  - Ký hiệu

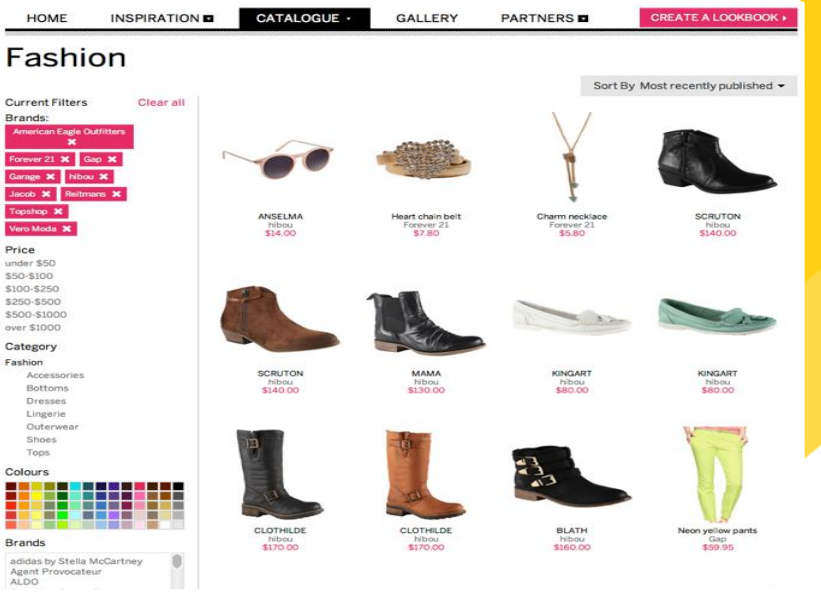


  - Mỗi phần cuối của một quan hệ kết hợp được gọi là một **vai trò/role**
    - Một vai trò thể hiện mục đích của quan hệ kết hợp
    - Một vai trò có thể có
      - một cái tên
      - một biểu hiện của **bội số/multiplicity**

Quan hệ kết hợp

- Bội số
  - xác định có bao nhiêu thể hiện của lớp A được kết hợp với một thể hiện của lớp B

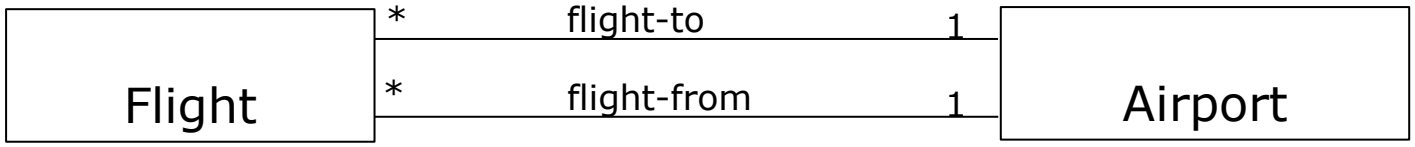




- Các biểu hiện khác nhau của **bội số**
  - 1 : một và chỉ một
  - 0..1 : Không hoặc chỉ một
  - m..n : từ m đến n (integer, n >= m >= 0)
  - n : chính xác n (integer, n >= 0)
  - * : Không hoặc nhiều
  - 1..* : từ một đến nhiều

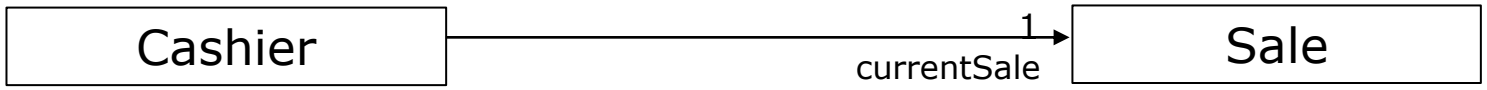
Quan hệ kết hợp

- Nhiều kết hợp
  - Hai lớp có thể có một số kết hợp giữa chúng

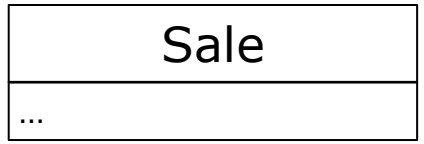


Quan hệ kết hợp

- Kết hợp theo hướng và thuộc tính
  - Theo mặc định, các quan hệ kết hợp có hai hướng
  - Tuy nhiên, các quan hệ kết hợp có thể được định hướng
  - Ví dụ

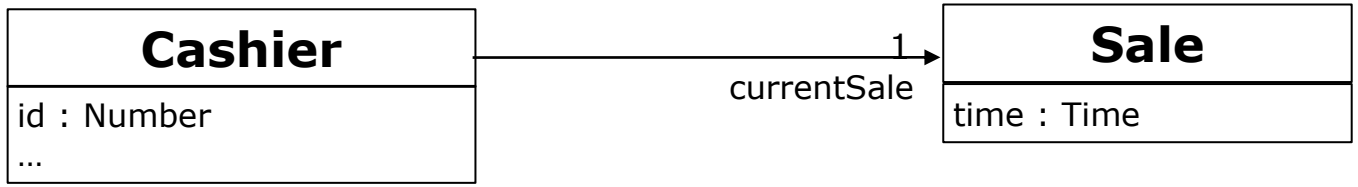


    - Khả năng điều hướng trỏ từ *Cashier* đến *Sale* cho thấy rằng một thuộc tính có kiểu *Sale*
    - Thuộc tính này được gọi là *currentSale*
  - Một dạng biểu diễn khác: sử dụng các thuộc tính



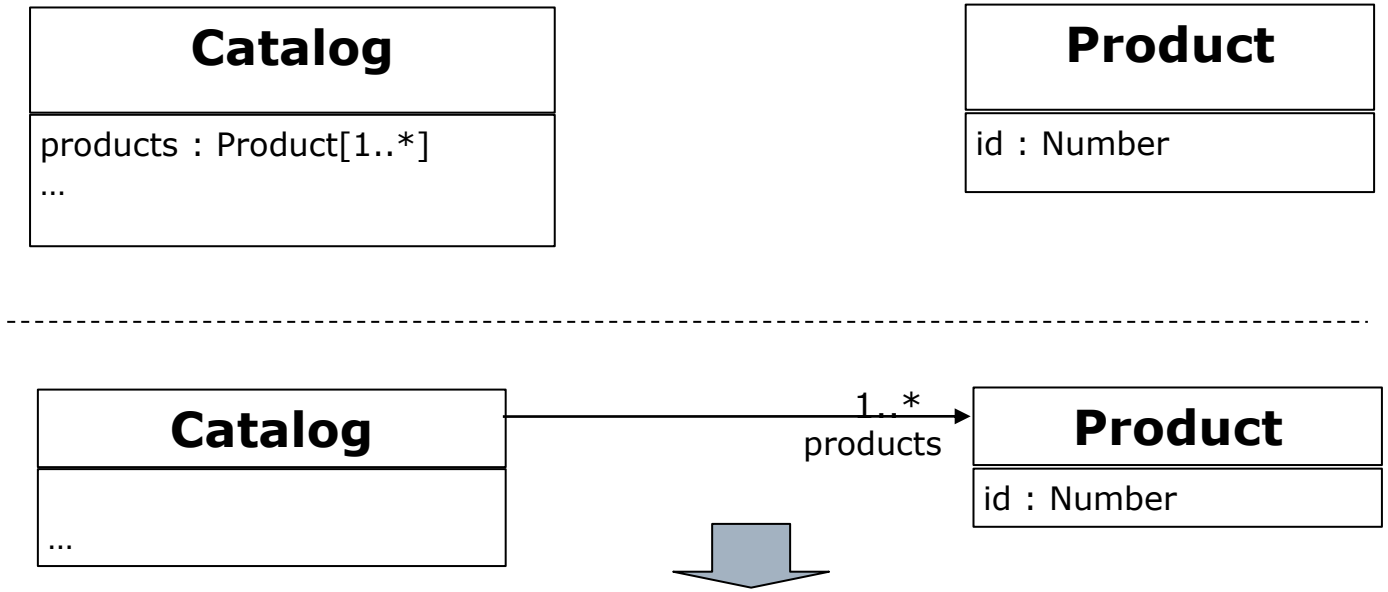
Quan hệ kết hợp

- Kết hợp theo hướng và thuộc tính
  - Khi nào chúng ta sử dụng kết hợp định hướng hoặc thuộc tính?
  - Chúng tôi sử dụng thuộc tính cho các kiểu dữ liệu “nguyên thủy”, chẳng hạn như Boolean, Time, Real, Integer,…
  - Chúng tôi sử dụng kết hợp định hướng cho các lớp khác
    - Để thấy rõ hơn mối liên hệ giữa các lớp
  - Nó chỉ là để thể hiện tốt hơn, hai cách này tương đương nhau về mặt ngữ nghĩa
    - Ví dụ

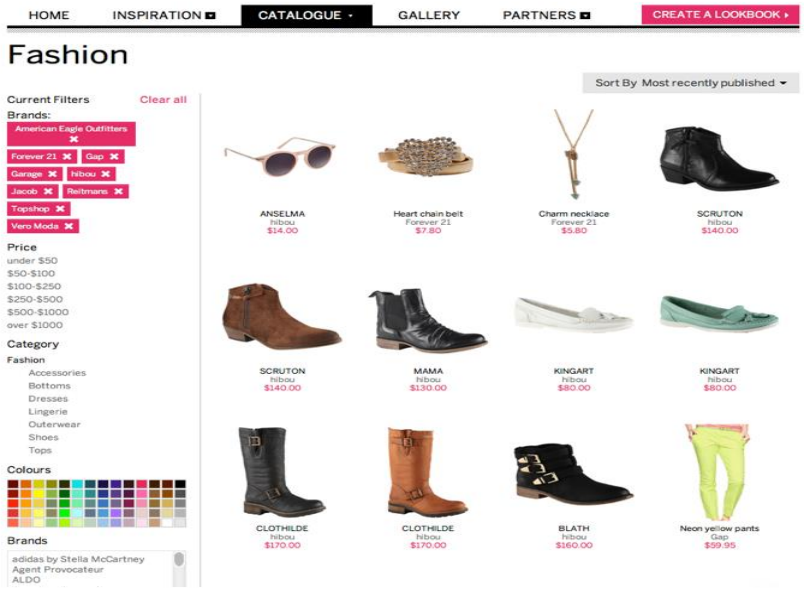


Quan hệ kết hợp

- Kết hợp theo hướng và thuộc tính
  - Ví dụ khác

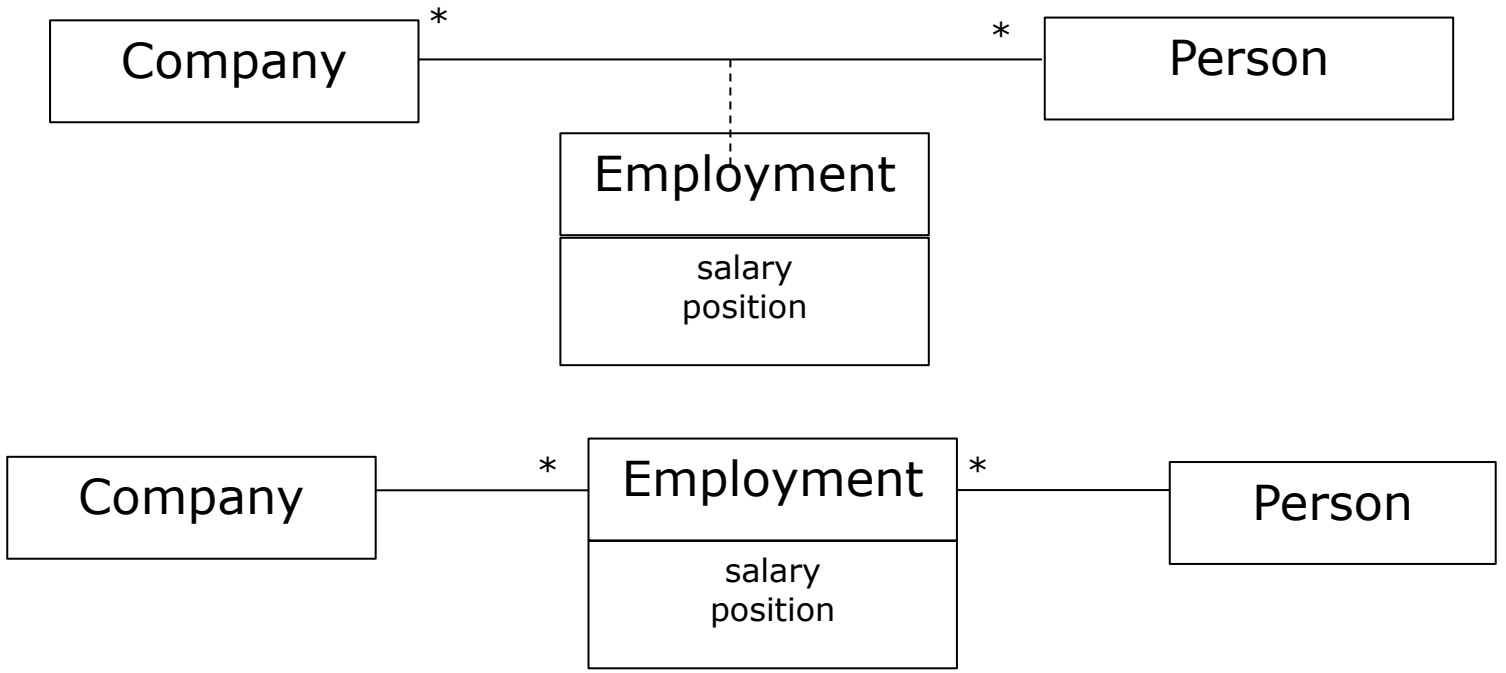



```
public class Catalog {
        private List<Product> products =
                        new ArrayList<Product>();
// ...
}
```



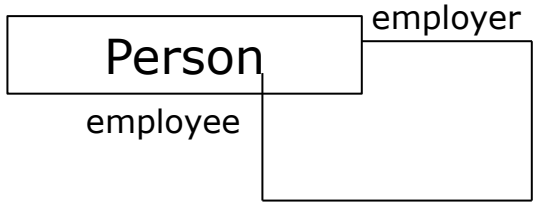
Quan hệ kết hợp

- Lớp kết hợp
  - Một lớp kết hợp cho phép một kết hợp được coi là một lớp. Khi một thuộc tính **không thể được gắn vào** bất kỳ hai lớp của một kết hợp
  - Ví dụ



Quan hệ kết hợp

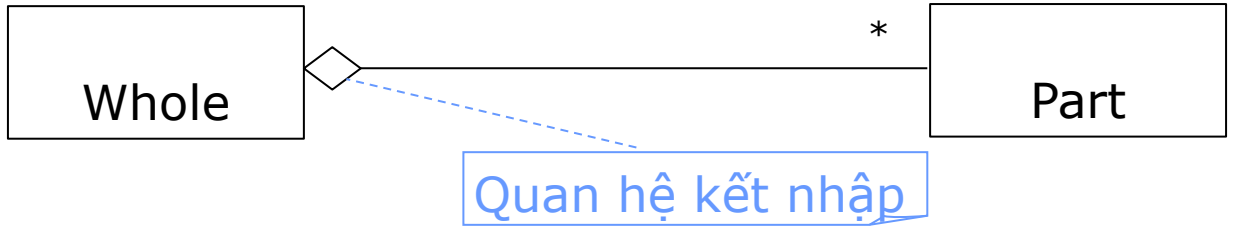
- Một lớp có thể kết hợp với chính nó
  - Ví dụ

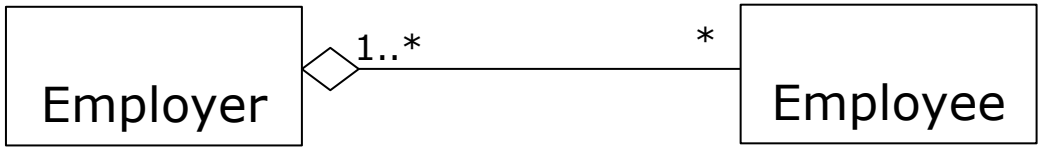


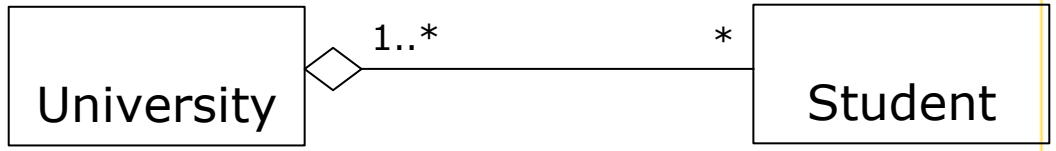
Employee Employer

Quan hệ kết nhập

- Quan hệ kết nhập là một dạng của quan hệ kết hợp thể hiện sự kết hợp mạnh mẽ hơn giữa các lớp (hơn quan hệ kết hợp bình thường)
- Quan hệ kết nhập được sử dụng giữa hai lớp
  - chủ và tớ: "thuộc về“
  - toàn bộ và một phần: "là một phần của“
- Ký hiệu
  - Biểu tượng biểu thị quan hệ kết nhập ở phía tổng hợp
- Ví dụ

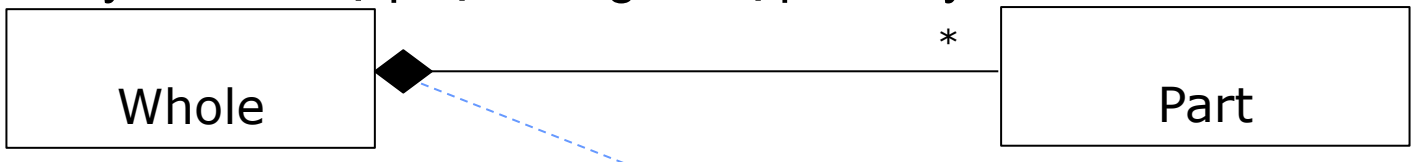




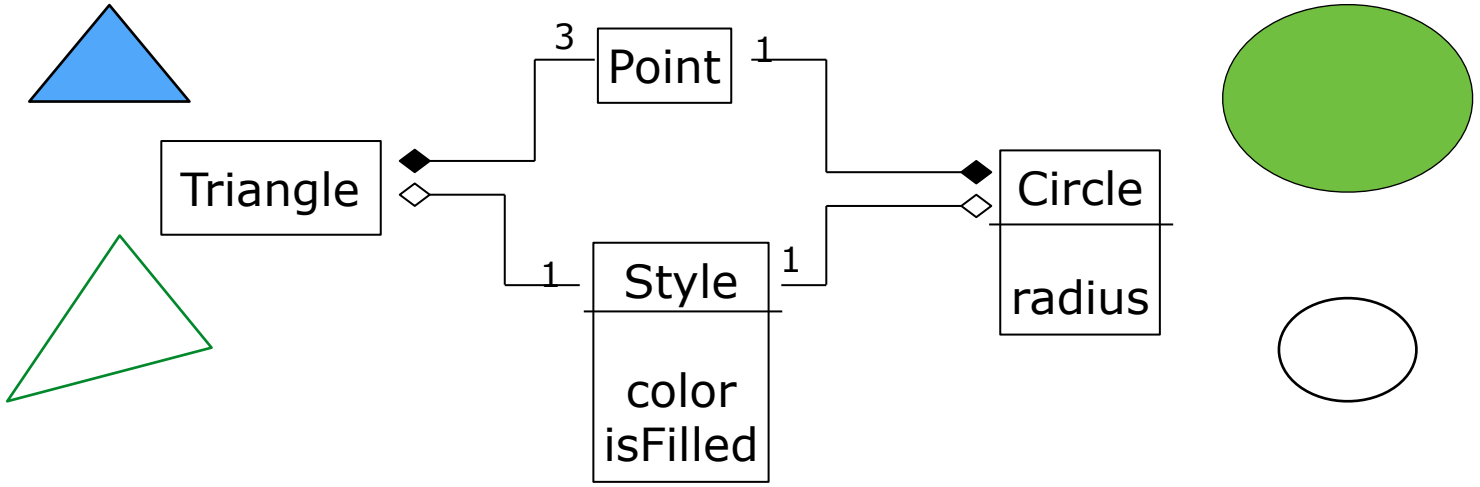


Quan hệ hợp thành

- Quan hệ hợp thành là một dạng mạnh của quan hệ kết nhập
- Quan hệ hợp thành cũng là một quan hệ "toàn bộ và một phần " nhưng quan hệ hợp thành mạnh hơn
- Nếu toàn bộ bị phá hủy thì các bộ phận cũng sẽ bị phá hủy

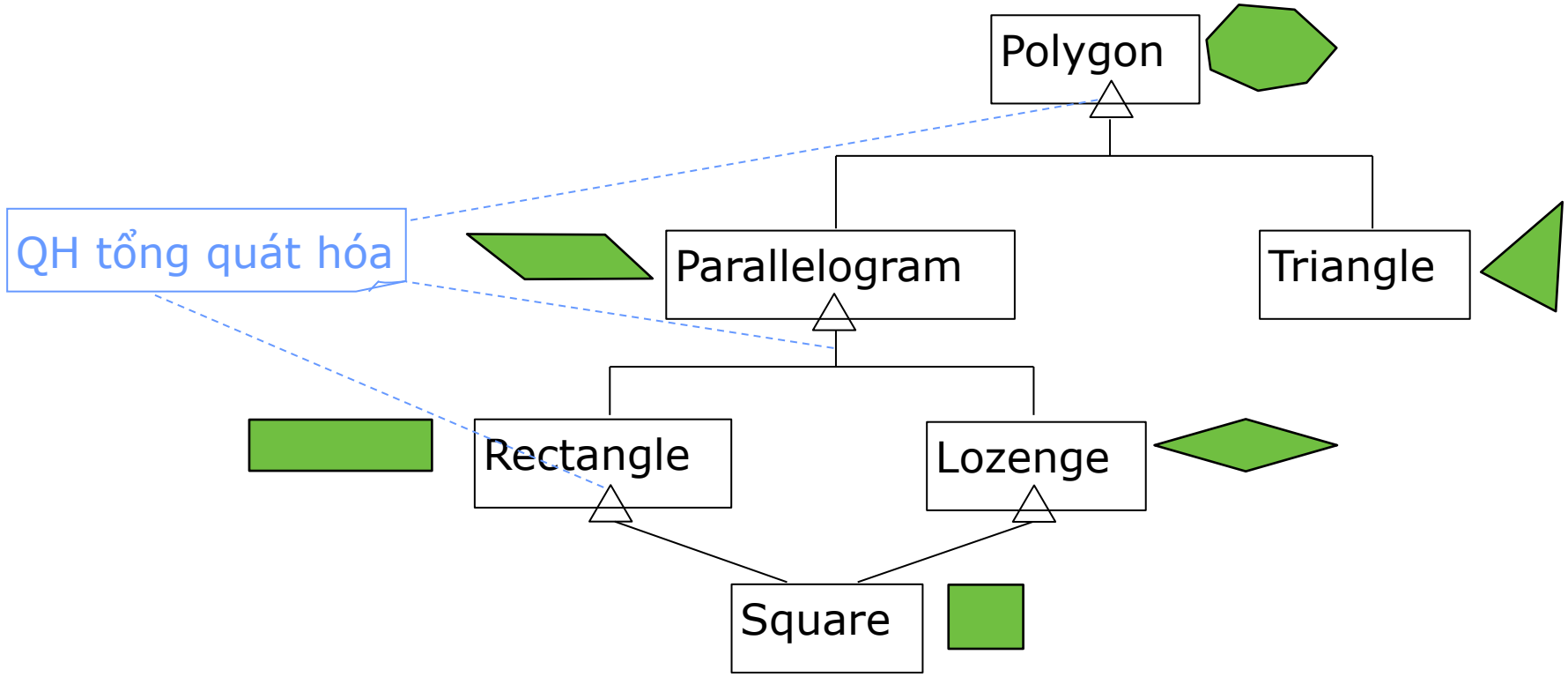


- Ví dụ



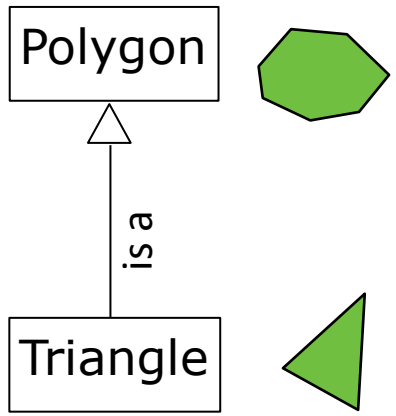
Quan hệ tổng quát hóa

- Một lớp có thể có vài lớp con



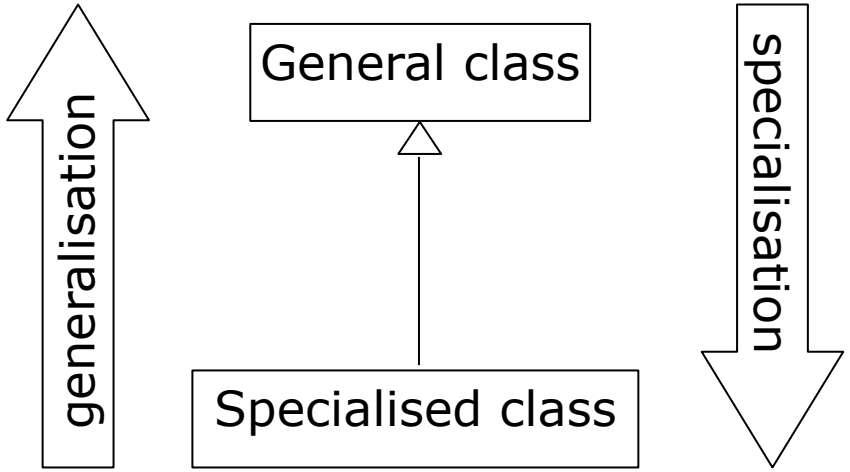
Quan hệ tổng quát hóa

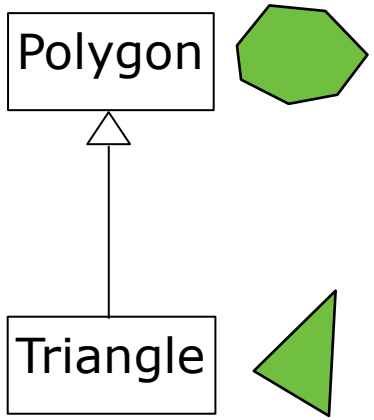
- Nguyên tắc thay thế
  - Tất cả các đối tượng của lớp con có thể đóng vai trò là một đối tượng của lớp cha của nó
  - Một đối tượng của một lớp con có thể ghi đè một đối tượng của lớp cha của nó
- Không chính thức: Lớp con là một loại lớp cha
- Ví dụ
  - Tam giác là một đa giác



Quan hệ tổng quát hóa

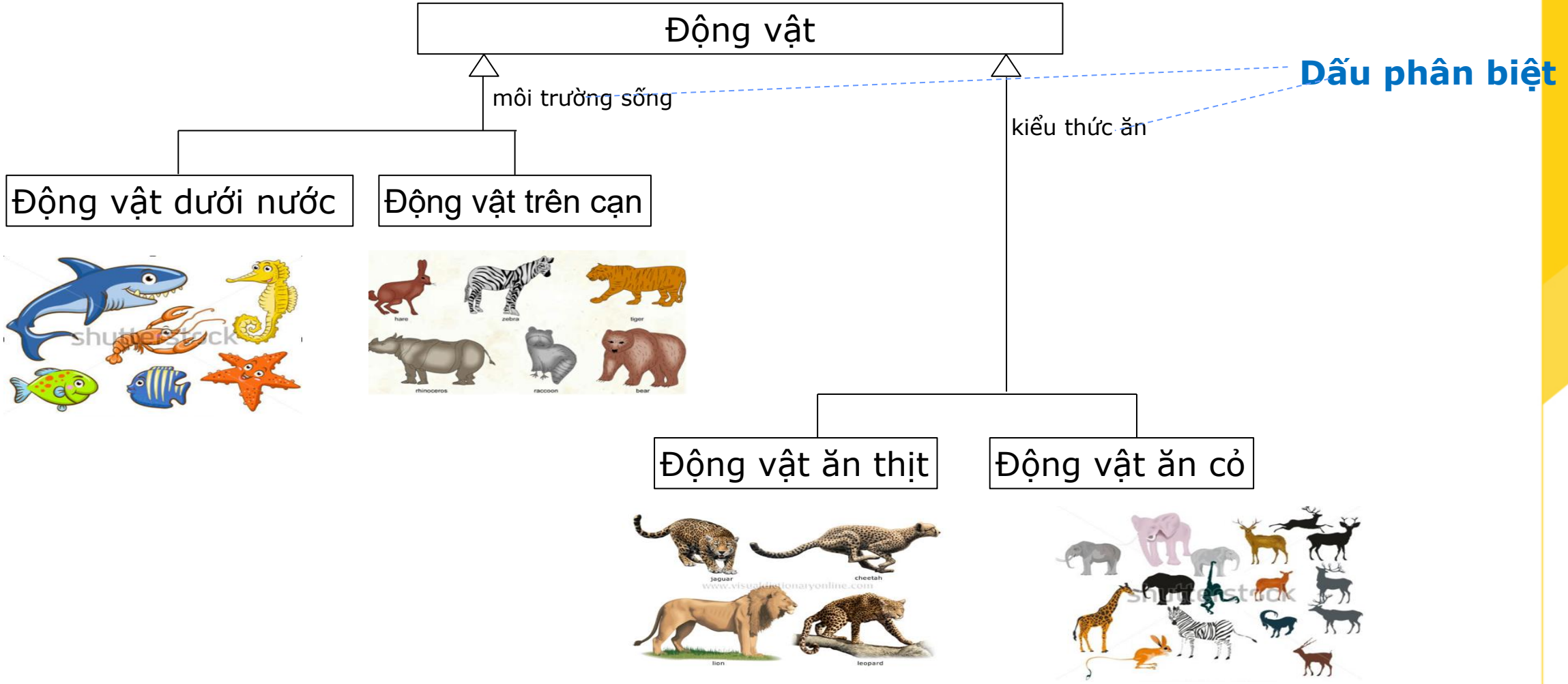
- Các lớp con còn được gọi là các **lớp chuyên biệt**
- Lớp cha còn được gọi là **lớp chung**
- Sự kế thừa còn được gọi là sự **chuyên biệt hóa** hoặc **tổng quát hóa**





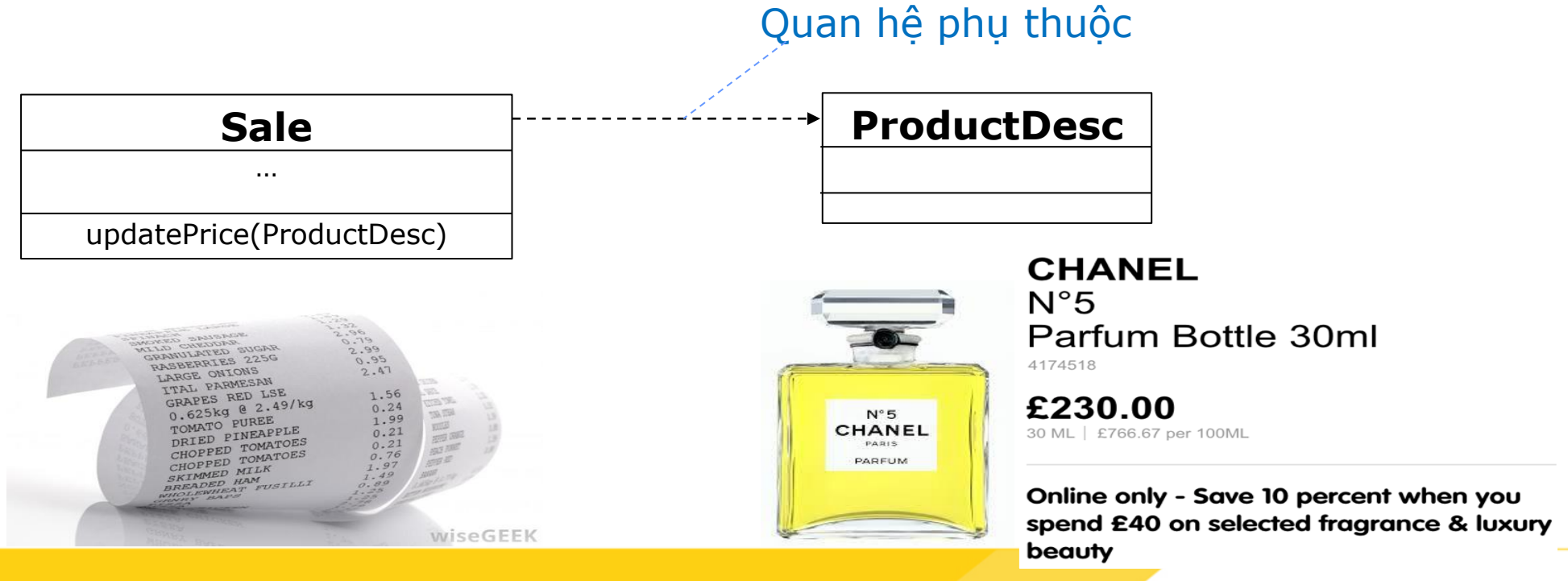
Quan hệ tổng quát hóa

- **Dấu phân biệt** (tùy chọn) là một nhãn mô tả tiêu chí mà **chuyên biệt hóa** dựa vào



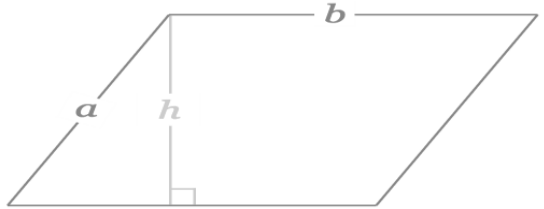
Quan hệ phụ thuộc

- Một lớp có thể phụ thuộc vào lớp khác
- Quan hệ phụ thuộc giữa các lớp có thể được cài đặt theo những cách khác nhau
  - Có một thuộc tính với kiểu của một lớp khác
  - Gửi thông điệp bằng thuộc tính, biến cục bộ, biến toàn cục của một lớp khác hoặc các phương thức tĩnh
  - Nhận một tham số có kiểu lớp khác
- Ví dụ

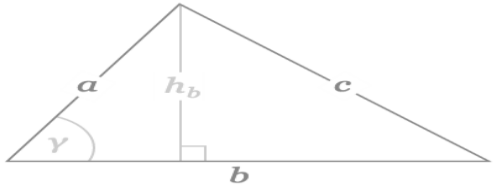


Lớp trừu tượng

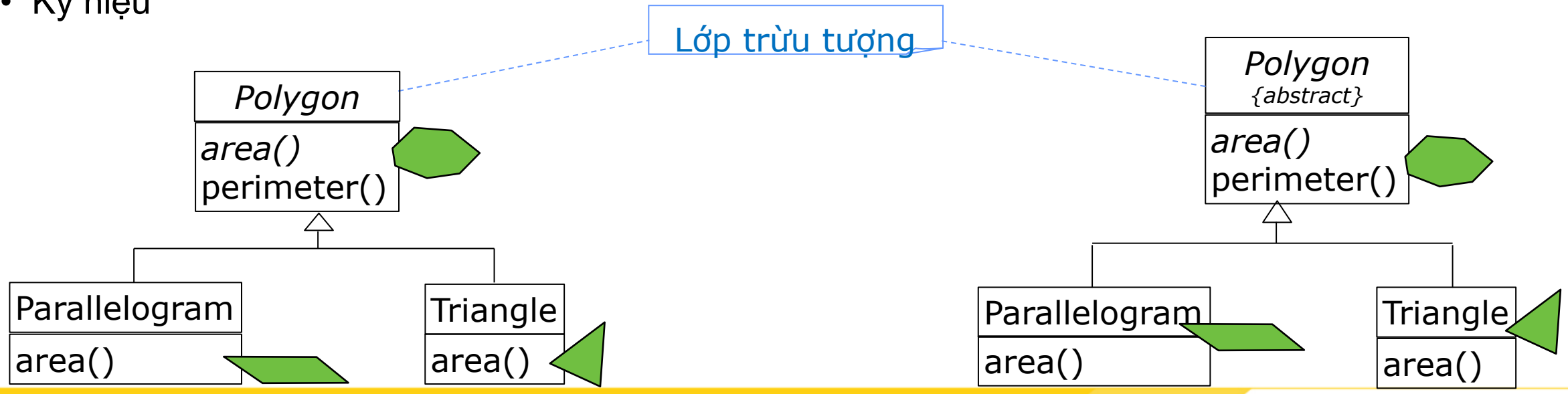
- Một lớp trừu tượng là một lớp không có thể hiện /đối tượng
  - Thừa kế: *area()*, perimeter()
  - Đa hình: *area()*
    - Parallelogram = b * h



Triangle = (h * b) / 2

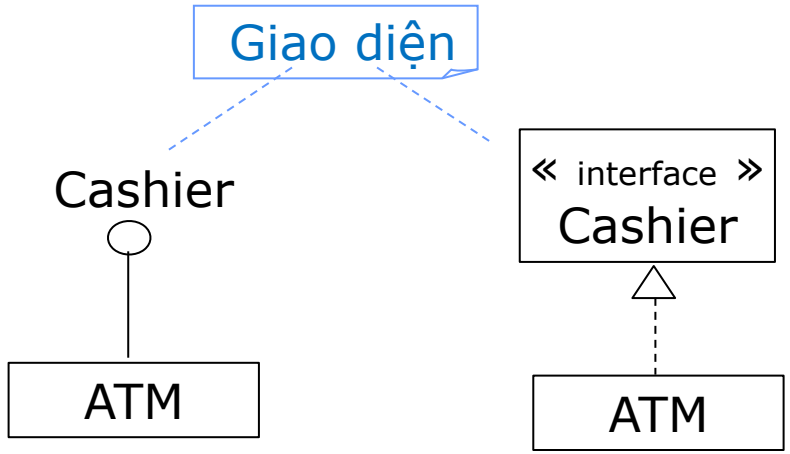


- Ký hiệu



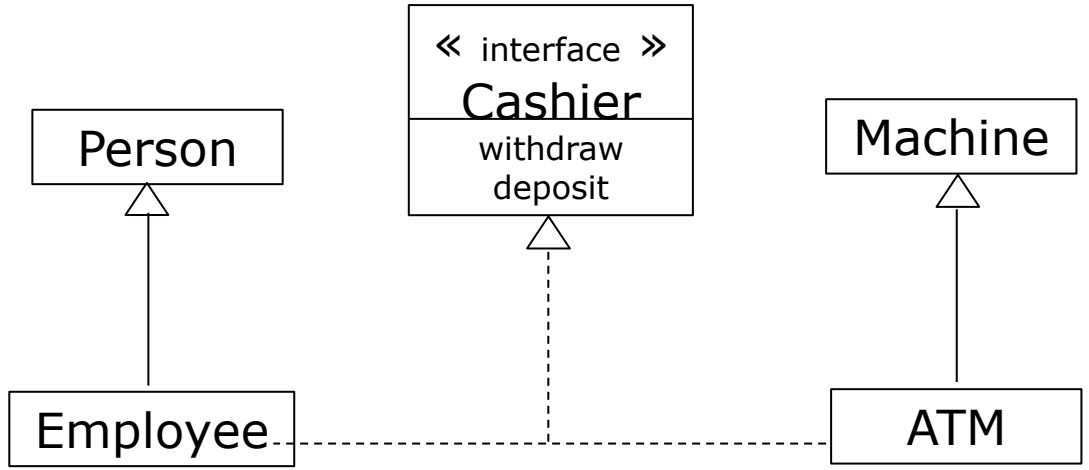
Giao diện

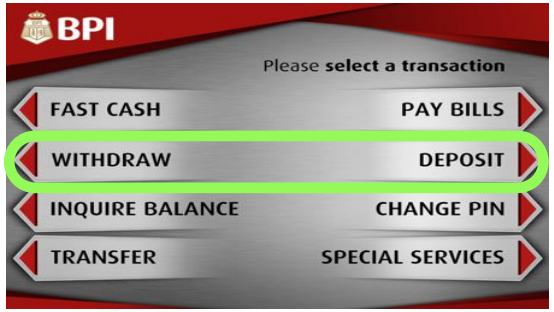
- Giao diện
  - mô tả một phần hành vi hiển thị của một tập các đối tượng
  - rất giống với một lớp trừu tượng chỉ chứa các thao tác trừu tượng
  - chỉ xác định các thao tác mà không cài đặt
- Hai ký hiệu



Giao diện

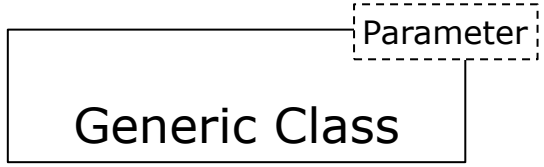
- Ví dụ





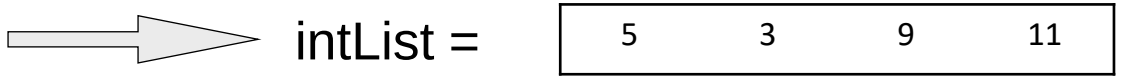
Lớp tham số hóa/Generic class

- Một lớp chung (hoặc tham số hóa) cho phép xem các kiểu dữ liệu như tham số
- Các lớp chung thường được sử dụng cho các loại lớp tập hợp: vectơ, bảng, ngăn xếp,…
- Ký hiệu



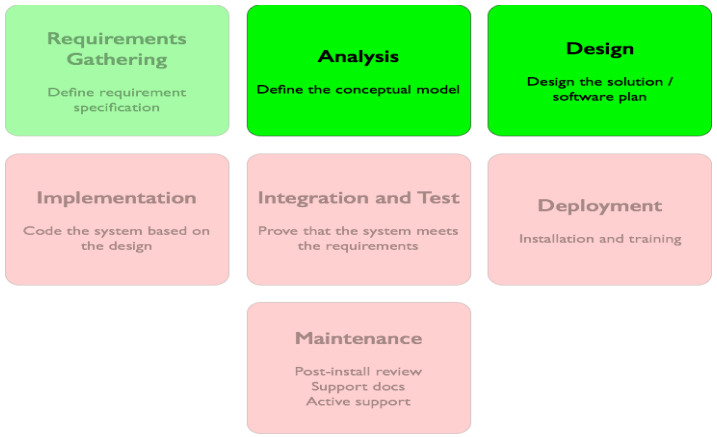
- Ví dụ

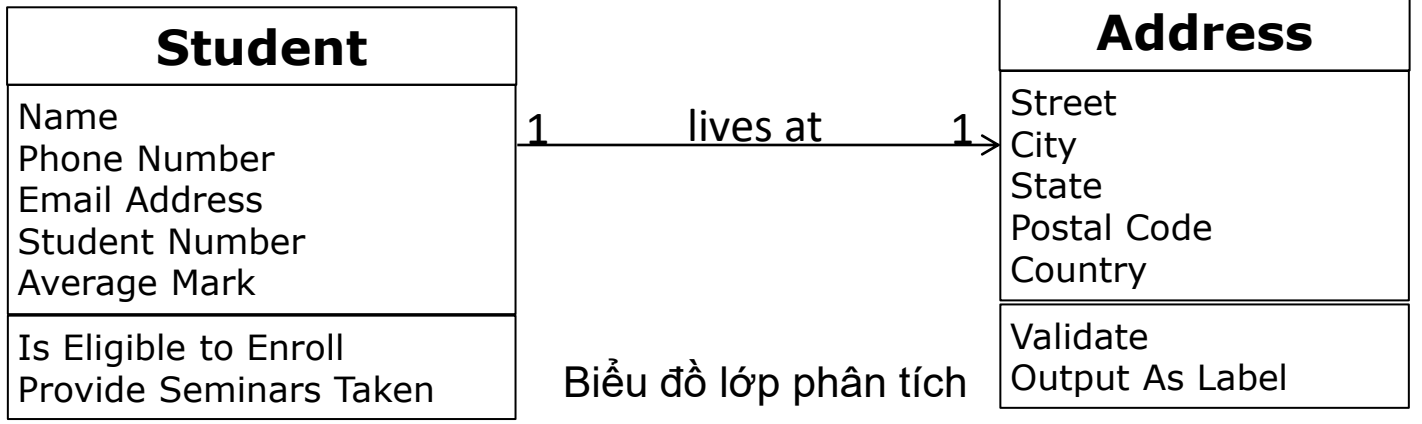
- “template” in C++
- Kiểu Generic trong Java
  - List<Integer> **intList** = new ArrayList<Integer>();



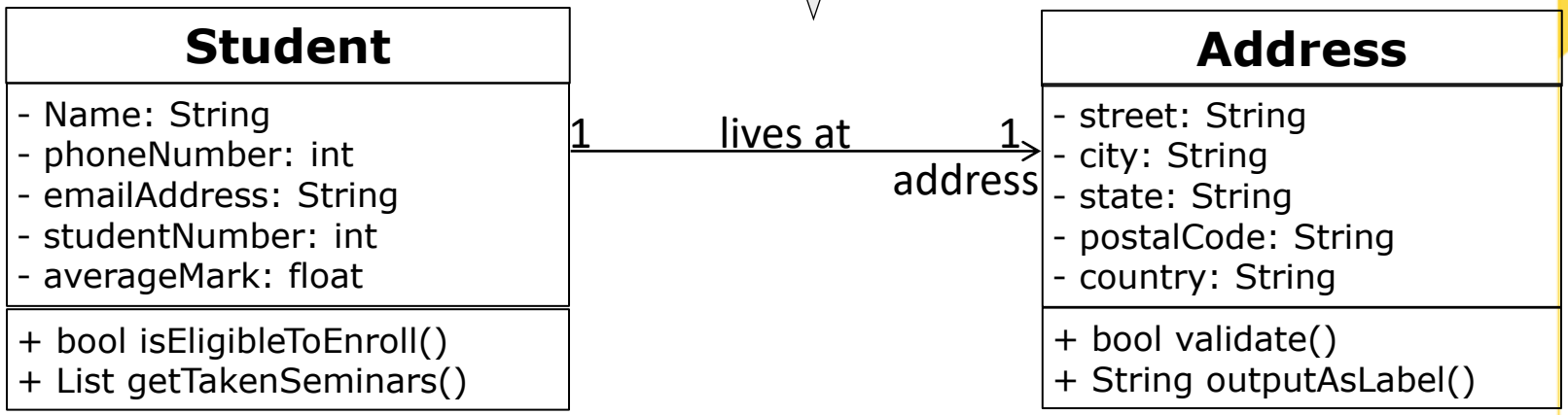
Xây dựng biểu đồ lớp

- Biểu đồ lớp được xây dựng dần dần ở các giai đoạn khác nhau của quá trình phát triển phần mềm





Biểu đồ lớp phân tích



Biểu đồ lớp thiết kế (cài đặt Java)

Xây dựng biểu đồ lớp

- Xác định các lớp
  - Câu hỏi "Làm thế nào để tìm lớp?“
  - Các khái niệm trong miền được nghiên cứu cũng có thể là các lớp
    - Các khái niệm này được gọi là **các lớp khái niệm**
    - Vì vậy, trước tiên chúng tôi xác định các lớp khái niệm, sau đó các lớp khác được thêm vào trong quá trình phát triển
- Các nguyên tắc để tìm các lớp khái niệm
  - Sử dụng **danh sách các danh mục**
  - Nhận dạng **danh từ**

Xây dựng biểu đồ lớp

- Xác định các lớp: Sử dụng danh sách các danh mục

Các danh mục của các lớp khái niệm	Ví dụ
giao dịch (của kinh doanh)	Reservation, Payment
sản phẩm hoặc dịch vụ liên quan đến giao dịch	Product, Flight
nơi giao dịch được ghi lại	Cash desk, Cash
tác nhân của các ca sử dụng	Cashier, Customer
địa điểm (dịch vụ, giao dịch)	Station, Store
sự kiện quan trọng	purchase
vật thể	Car
mô tả sự vật	Description of products
mục lục	Product catalog
vật chưa	Store
các hệ thống cộng tác khác	Bank, database
tổ chức	University
chính sách, nguyên tắc	Tax
...	

Xây dựng biểu đồ lớp

- Xác định các lớp: Nhận dạng **danh từ**
  - Xem xét các tài liệu bằng văn bản như đặc tả hoặc mô tả các ca sử dụng
  - Trích xuất các tên và xem chúng như các ứng viên của lớp khái niệm
  - Loại bỏ các danh từ
    - dư thừa
    - mơ hồ hoặc quá chung chung
    - không phải là các lớp khái niệm theo kinh nghiệm và kiến thức trong ngữ cảnh của ứng dụng

Xây dựng biểu đồ lớp

- Ví dụ

Các hành động của tác nhân	Các hành động của hệ thống
1. **Khách hàng** đưa các **mặt hàng** cần mua đến **quầy thu ngân**	
2. **Thu ngân** ghi nhận từng **mặt hàng**. Nếu một **mặt hàng** có số lượng nhiều hơn một, nhân viên **thu ngân** nhập vào số lượng	3. Hiển thị **mô tả** và giá của **mặt hàng**. Số này được hiển thị
4. Sau khi ghi nhận tất cả các **mặt hàng**, nhân viên **thu ngân** báo hiệu kết thúc việc ghi nhận mặt hàng	5. Tính toán và hiển thị tổng số tiền **khách hàng** phải thanh toán
6. **Thu ngân** thông báo tổng số tiền cho khách hàng	
7. **Khách hàng** trả tiền	
8. **Thu ngân** nhập số tiền **khách hàng thanh toán**	9. Hiển thị **số dư**

Xây dựng biểu đồ lớp

- Ví dụ (tiếp theo)

Các hành động của tác nhân	Các hành động của hệ thống
	10. In **phiếu thu**
11. **Thu ngân** đưa **tiền lẻ** cho **khách hàng** và **phiếu thu**	12. Lưu thông tin **phiên bán hàng**
13. **Khách hàng** rời **quầy thu ngân** với các **mặt hàng** đã mua	

Xây dựng biểu đồ lớp

- Các lớp ứng cử viên từ danh từ được xác định từ mô tả ca sử dụng: **khách hàng**, **quầy thu ngân**, **mặt hàng**, **thu ngân**, **phiên bán hàng, thanh toán**

Xây dựng biểu đồ lớp

- Xác định các **quan hệ** và **thuộc tính**
  - Bắt đầu với **các lớp trung tâm** của hệ thống
  - Xác định các thuộc tính của mỗi lớp và quan hệ kết hợp với các lớp khác
  - Tránh thêm quá nhiều thuộc tính hoặc quan hệ kết hợp vào một lớp. Để quản lý lớp tốt hơn

Xây dựng biểu đồ lớp

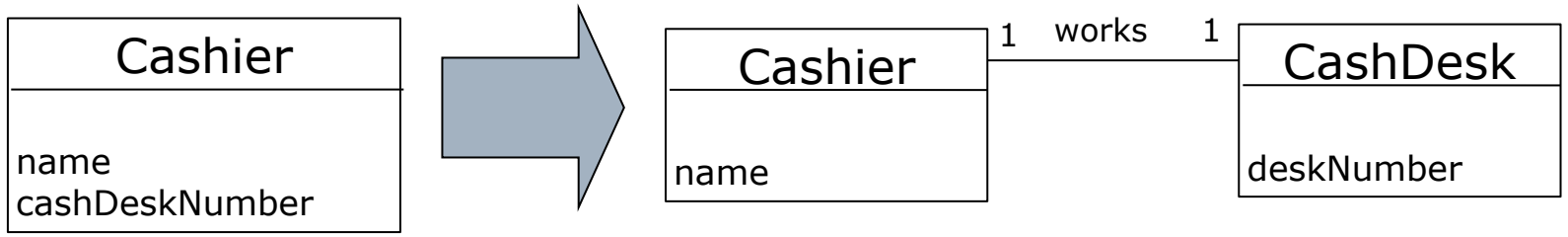
- Xác định các **quan hệ**
  - Một **quan hệ kết hợp** phải tồn tại giữa lớp A và lớp B, nếu
    - A là một dịch vụ hoặc sản phẩm của B
    - A là một phần của B
    - A là mô tả cho B
    - A là một thành viên của B
    - A được kết nối với B
    - A sở hữu B
    - A điều khiển B
    - …
  - Chỉ định **bội số** ở mỗi đầu của quan hệ kết hợp
  - Gán nhãn quan hệ kết hợp

Xây dựng biểu đồ lớp

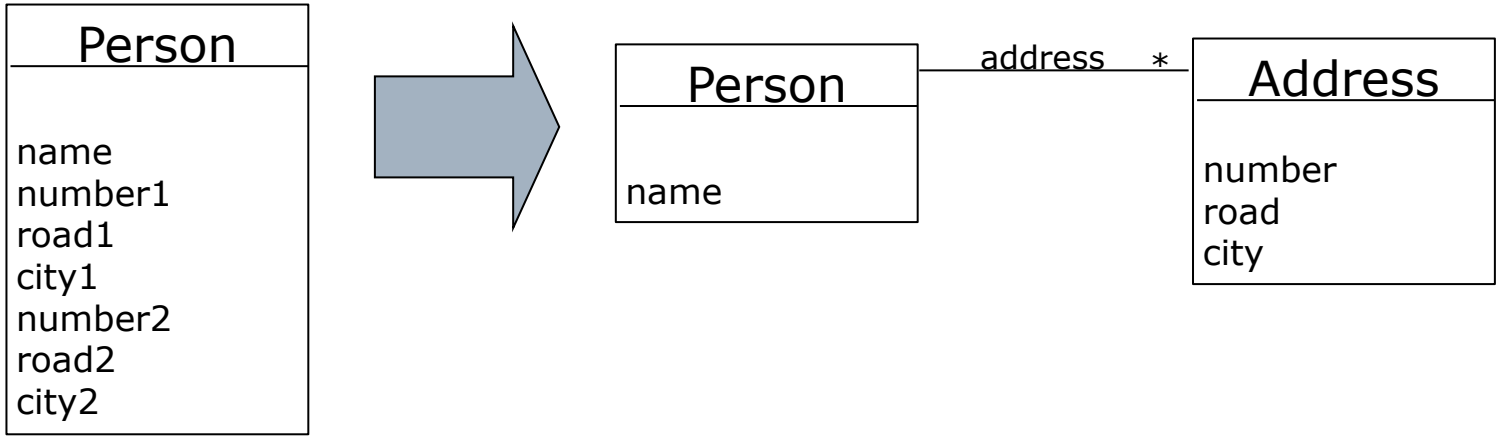
- Xác định thuộc tính
  - Đối với mỗi lớp, xác định thông tin cần thiết để lưu trữ theo đặc tả yêu cầu hoặc ca sử dụng
    - Ví dụ: Thu ngân cần số định danh, tên,…
  - Nguyên tắc xác định thuộc tính
    - Một thuộc tính chỉ đại diện cho dữ liệu liên quan đến lớp sở hữu thuộc tính
    - Nếu một tập con của các thuộc tính tạo thành một nhóm nhất quán, có thể tạo một lớp mới
  - Chỉ xác định tên của các thuộc tính ở giai đoạn này (tức là giai đoạn phân tích)

Xây dựng biểu đồ lớp

- Xác định thuộc tính
  - Ví dụ: Một thuộc tính chỉ đại diện cho dữ liệu liên quan đến lớp sở hữu thuộc tính

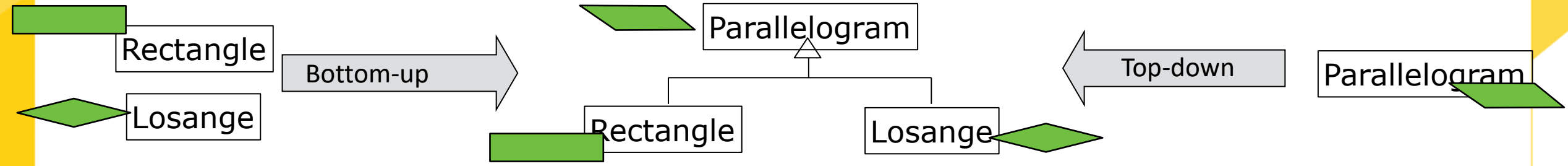


  - Nếu một tập con của các thuộc tính tạo thành một nhóm nhất quán, có thể tạo một lớp mới



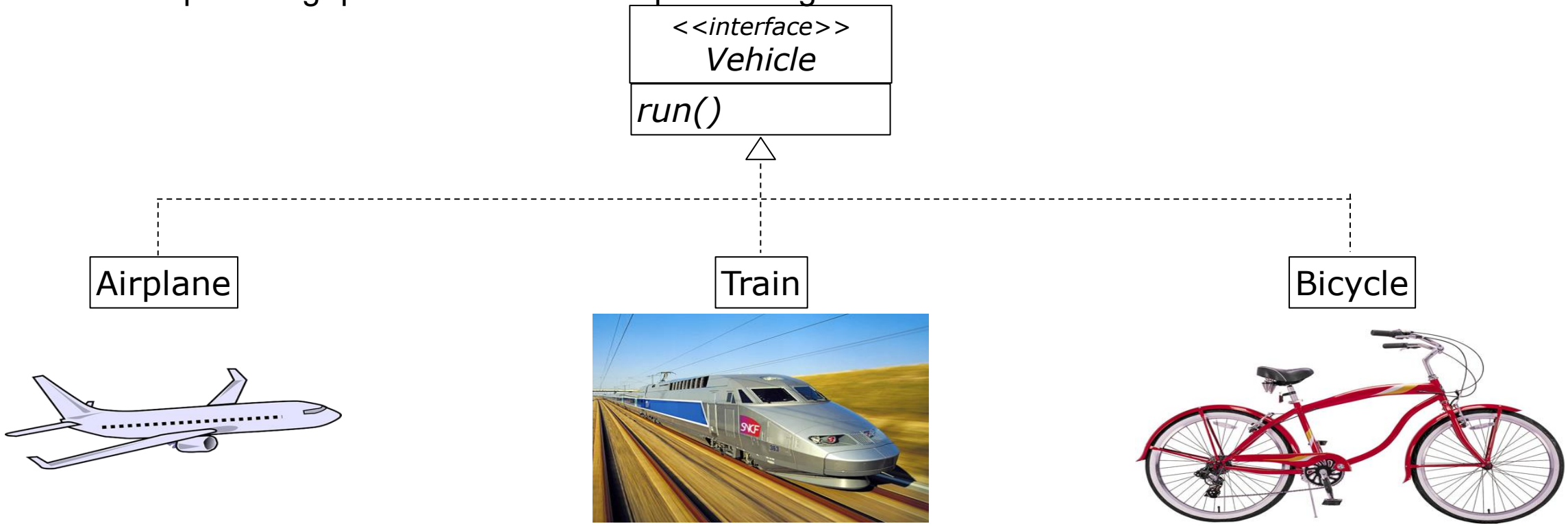
Xây dựng biểu đồ lớp

- Xác định **quan hệ thừa kế:** Hai cách tiếp cận
  - Dưới lên. Tổng quát hóa: nhóm các lớp tương tự để tạo siêu lớp
  - Trên xuống. Chuyên biệt hóa: xây dựng các lớp con từ các lớp chung hiện có



Xây dựng biểu đồ lớp

- Xác định giao diện
  - Tạo giao diện thay vì siêu lớp, nếu
    - Nó là cần thiết để nhận ra các cài đặt khác nhau của cùng một lớp
    - Hai lớp tạo ra chia sẻ các thao tác không giống nhau
    - Lớp để tổng quát hóa đã có siêu lớp của riêng nó



Xây dựng biểu đồ lớp

- Xác định trách nhiệm của các lớp
  - Trách nhiệm là một hoặc một số nhiệm vụ mà hệ thống phải thực hiện
  - **Mỗi yêu cầu chức năng phải được quy cho một trong các lớp**
    - Tất cả các trách nhiệm phải được quy cho một trong các lớp
    - Nếu một lớp có quá nhiều trách nhiệm thì phải chia thành nhiều lớp
    - Nếu một lớp không có trách nhiệm, nên xóa nó
    - Nếu không thể giao trách nhiệm cho bất kỳ lớp nào, một lớp mới có thể được tạo
  - Các trách nhiệm có thể được xác định bằng cách phân tích các hành động / động từ trong đặc tả ca sử dụng.

Xây dựng biểu đồ lớp

- Phát triển **biểu đồ lớp thiết kế**
  - Dựa trên biểu đồ lớp phân tích (mô hình miền)
  - Biểu đồ lớp phân tích chi tiết
    - Tạo các lớp mới, nếu cần. Ví dụ, một quan hệ kết hợp của lớp trở thành một lớp mới
    - Chi tiết các thuộc tính
    - Thêm và chi tiết các quan hệ
    - Xác định thao tác/ chức năng

Xây dựng biểu đồ lớp

- Chi tiết các thuộc tính
  - Xác định các loại thuộc tính
    - Sử dụng các kiểu nguyên thủy: boolean, int, real,…
    - Xác định kiểu mới cho một thuộc tính (lớp mới), nếu
      - Nó bao gồm một số phần
      - Nó có các thuộc tính khác
      - Nó được kết hợp với các thao tác khác
  - Xác định giá trị ban đầu nếu cần
  - Xác định khả năng hiển thị của các thuộc tính

Xây dựng biểu đồ lớp

- Chi tiết các quan hệ
  - Giới thiệu các quan hệ theo các lớp mới được thêm vào
  - Chỉ định xem một quan hệ kết hợp là một quan hệ kết nhập hay một quan hệ hợp thành
  - Đặt tên cho mối quan hệ
  - Đưa ra hướng

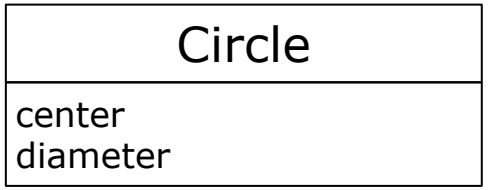
Xây dựng biểu đồ lớp

- Xác định các thao tác/chức năng của mỗi lớp
  - getters và setters
  - Các thao tác được sử dụng để đạt được các trách nhiệm đã xác định
  - Một trách nhiệm có thể được thực hiện bằng một số thao tác
  - Xác định khả năng hiển thị của các thao tác
    - Các thao tác thiết yếu thực hiện trách nhiệm được tuyên bố là " public“
    - Các thao tác chỉ phục vụ trong lớp được tuyên bố là "private" hoặc " protected " nếu lớp đó được kế thừa

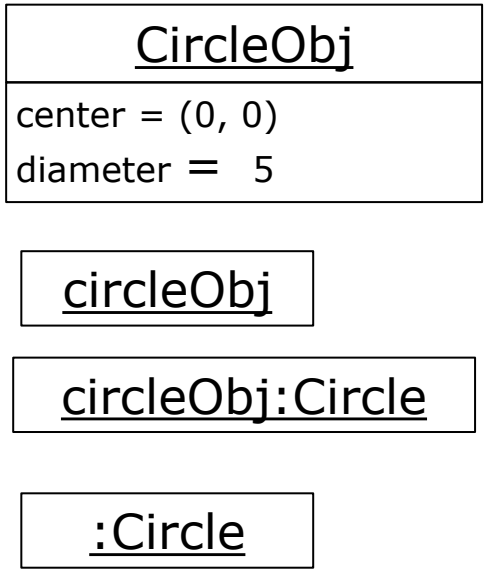
Biểu đồ đối tượng

- Đối tượng
  - Đối tượng là thể hiện của các lớp
  - Ký hiệu
    - Giá trị của các thuộc tính có thể được chỉ định
    - Tên của đối tượng được gạch dưới

Lớp

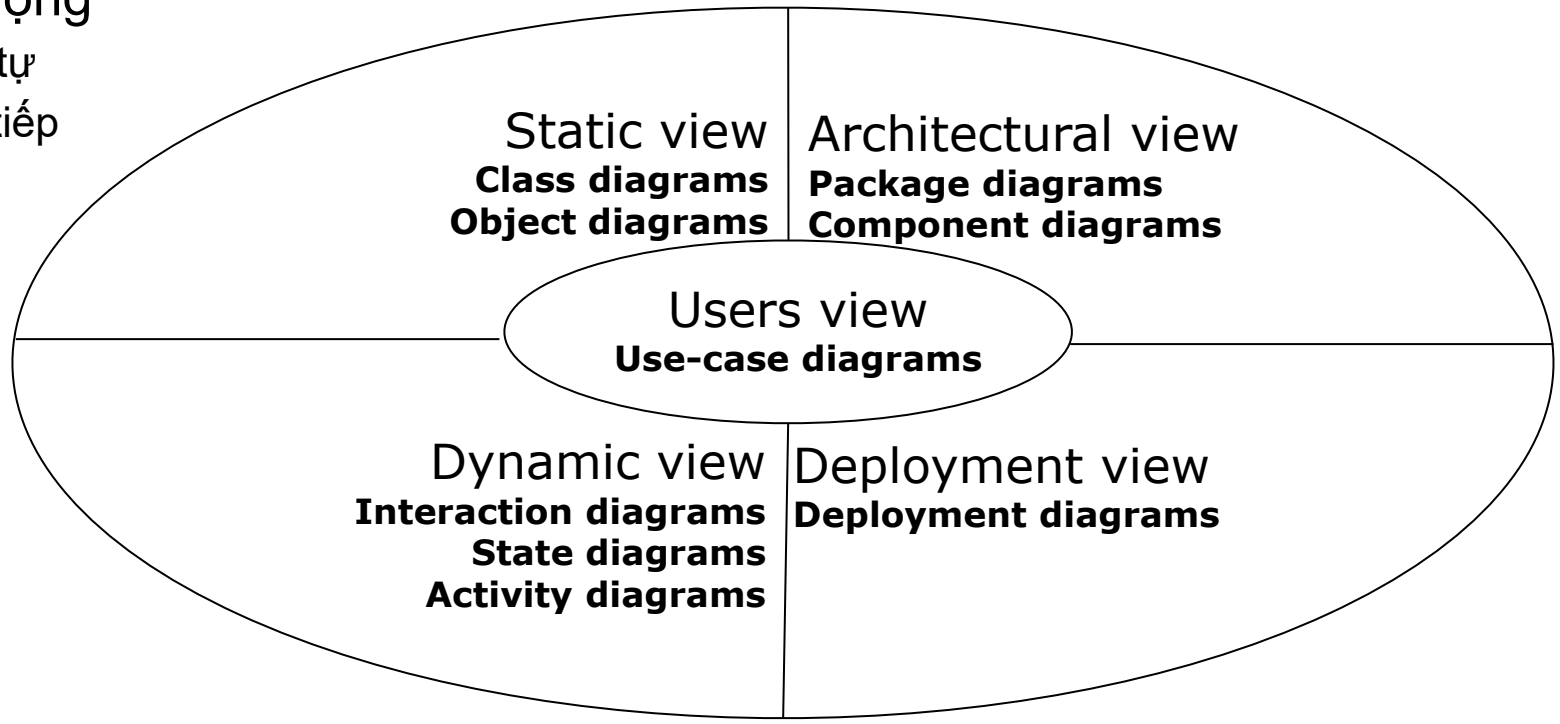


Đối tượng



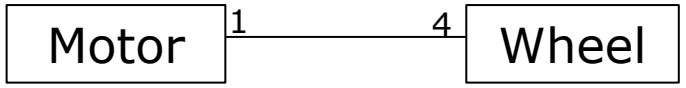
Biểu đồ đối tượng

- Đối tượng: Ba loại biểu đồ với các đối tượng
  - Chế độ xem tĩnh
    - Biểu đồ đối tượng
  - Chế độ xem động
    - Biểu đồ trình tự
    - Biểu đồ giao tiếp

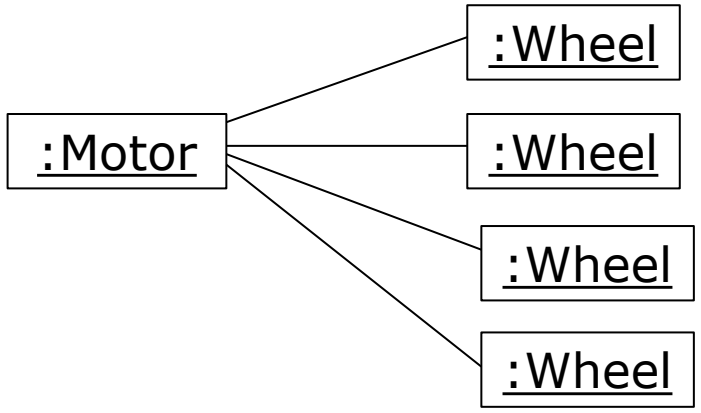


Biểu đồ đối tượng

- Biểu đồ đối tượng
  - đại diện cho một tập các đối tượng và liên kết giữa chúng
  - là các khung nhìn tĩnh các thể hiện của các phần tử xuất hiện trong biểu đồ lớp
- Biểu đồ đối tượng là một thể hiện của biểu đồ lớp



Biểu đồ lớp



Biểu đồ đối tượng

Đồ án (1)

- Chia nhóm 4-5 sinh viên
- Mỗi nhóm chọn một bài toán
- Xây dựng biểu đồ lớp phân tích: Chọn một trong các công cụ sau:
  - Microsoft Visio
  - StarUML: http://staruml.io/
  - Argo UML: https://argouml.jaleco.com/
  - Lucidchart: https://www.lucidchart.com/pages/examples/uml_diagram_tool

Đồ án (1)

- Chia nhóm 4-5 sinh viên
- Mỗi nhóm chọn một bài toán
- Xây dựng biểu đồ lớp thiết kế: Chọn một trong các công cụ sau:
  - Microsoft Visio
  - StarUML: http://staruml.io/
  - Argo UML: https://argouml.jaleco.com/
  - Lucidchart: https://www.lucidchart.com/pages/examples/uml_diagram_tool