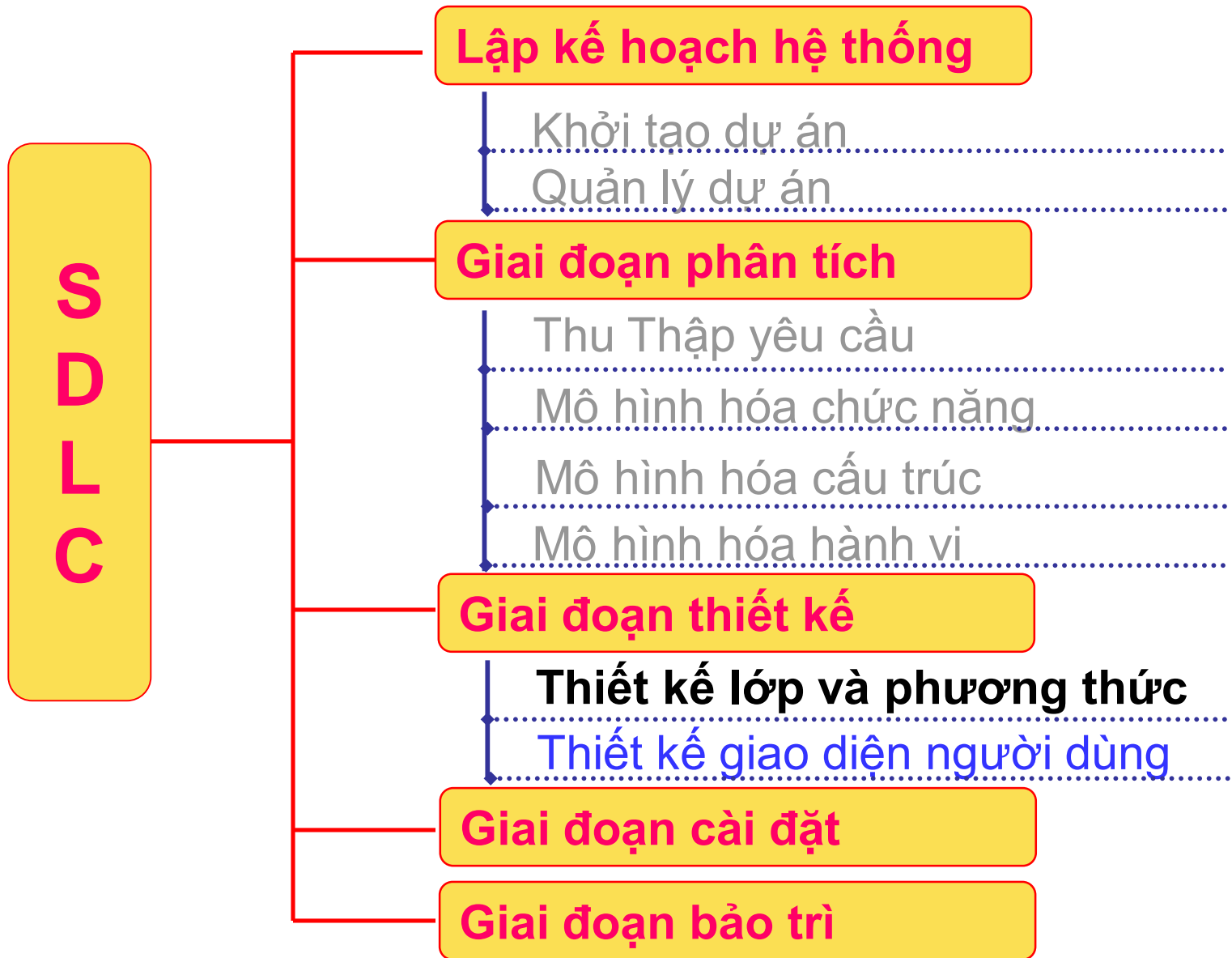


## Chương 8

# THIẾT KẾ LỚP VÀ PHƯƠNG THỨC

1. Một số đặc điểm cơ bản của hướng đối tượng
2. Các chuẩn thiết kế
3. Hoạt động thiết kế đối tượng
4. Các ràng buộc và contract
5. Sự đặc tả phương thức
6. Áp dụng vào Công ty CD Selections
7. Tóm tắt



# Một số đặc điểm cơ bản của hướng đối tượng

---

- Lớp, đối tượng, phương thức và thông điệp.
- Sự đóng gói và che dấu thông tin.
- Tính đa hình.
- Kế thừa.

# Một số đặc điểm cơ bản của hướng đối tượng

## - Lớp, đối tượng, phương thức và thông điệp

- **Đối tượng** là thực thể của hệ thống, của CSDL và được xác định thông qua ID của chúng.
- Mỗi đối tượng có tập các đặc trưng, là các **thuộc tính** và các **phương thức**, các **thao tác** trên các dữ liệu để xác định hành vi của đối tượng đó.

# Một số đặc điểm cơ bản của hướng đối tượng

## - Lớp, đối tượng, phương thức, thông điệp (tt)

- **Lớp** là một khuôn mẫu của tất cả các đối tượng có những đặc trưng giống nhau. Đối tượng chính là thể hiện của một lớp xác định.
- Để đối tượng thực hiện một phương thức, thì gửi đến đối tượng đó một **thông điệp** - là một **hàm** hay **thủ tục** gọi từ đối tượng này đến đối tượng khác.

# Một số đặc điểm cơ bản của hướng đối tượng

## - Sự đóng gói và che dấu thông tin

- Sự đóng gói:
  - Là cơ chế kết nối các phương thức và dữ liệu vào trong một đối tượng riêng lẻ.
- Sự che dấu thông tin:
  - Chỉ có thông tin được cần thiết (yêu cầu sử dụng) mới available, cách mà đối tượng lưu trữ dữ liệu và thực thi hành động là không quan trọng.
  - Đối tượng được xem như một hộp đen

# Một số đặc điểm cơ bản của hướng đối tượng

## - Tính đa hình

- Tính đa hình
  - khả năng thực hiện rất nhiều form. Hệ thống hướng đối tượng có thể gửi một thông điệp chung đến nhiều đối tượng và mỗi đối tượng sẽ xử lý thông điệp theo một cách riêng
- Liên kết động
  - dạng liên kết các hàm, chức năng khi chương trình thực hiện các lời gọi hàm, chức năng đó.

# Một số đặc điểm cơ bản của hướng đối tượng

## - Kế thừa

- Sự kế thừa:
  - cho phép người thiết kế định nghĩa nhiều lớp bằng cách sử dụng lại các lớp đã định nghĩa trước như một cơ sở cho lớp mới.
  - Các lớp con kế thừa các thuộc tính và các phương thức thích hợp từ lớp cha. Sự kế thừa giúp định nghĩa lớp đơn giản hơn.
- Có nhiều dạng kỹ thuật kế thừa, kỹ thuật kế thừa phổ biến nhất là hình thức kế thừa đơn và kế thừa bội.



# Tiêu chuẩn thiết kế

---

- Theo Coad và Yourdon, một trong những thiết kế tốt là sự cân bằng giảm đến mức tối thiểu tổng chi phí của hệ thống trên toàn bộ vòng đời của nó. Các tiêu chuẩn này bao gồm:
  - Sự kết hợp (Coupling)
  - Sự kết dính (Cohesion)
  - Cùng sinh (Connascence)

# Tiêu chuẩn thiết kế - sự kết hợp

---

- Sự kết hợp (Coupling) nói đến mức độ sự phụ thuộc hoặc mối quan hệ giữa các module (lớp, đối tượng, phương thức) trong hệ thống.
- Đối với hệ thống hướng đối tượng, Coad và Yourdon đưa ra hai loại kết hợp:
  - Sự kết hợp tương tác
  - Sự kết hợp kế thừa

## Tiêu chuẩn thiết kế - sự kết hợp (tt)

- **Sự kết hợp tương tác:** đề cập đến sự kết hợp giữa các phương thức và đối tượng qua sự truyền thông điệp.
- Có 6 loại kết hợp tương tác sau sắp xếp theo mức từ tốt đến xấu:
  - kết hợp không trực tiếp, dữ liệu, nhãn, điều khiển, thông thường hoặc toàn cục, hợp lý hoặc không hợp lý

## Tiêu chuẩn thiết kế - sự kết hợp (tt)

- **Sự kết hợp tương tác:** đề cập đến sự kết hợp giữa các phương thức và đối tượng qua sự truyền thông điệp.
- Có 6 loại kết hợp tương tác sau: kết hợp không trực tiếp, dữ liệu, nhân, điều khiển, thông thường hoặc toàn cục, hợp lý hoặc không hợp lý

## Tiêu chuẩn thiết kế - sự kết hợp (tt)

---

- **Kết hợp không trực tiếp:** Các phương thức không có mối quan hệ với phương thức khác.
- **Dữ liệu:** Gọi các phương thức qua biến.
- **Nhãn:** Gọi phương thức qua biến phức (đối tượng), nhưng gọi phương thức chỉ sử dụng một phần của đối tượng để thực hiện chức năng của nó.

## Tiêu chuẩn thiết kế - sự kết hợp (tt)

- **Điều khiển:** Gọi phương thức qua biến điều khiển.
- **Hợp lệ hoặc không hợp lệ:** Một phương thức của một đối tượng đề cập đến thuộc tính ẩn của đối tượng khác.
- **Sự kết hợp kế thừa:** đề cập đến sự phụ thuộc vốn có trong sử dụng kế thừa trong hệ thống hướng đối tượng.

# Tiêu chuẩn thiết kế - sự kết dính

- Sự kết dính (Cohesion) đề cập đến một Module như thế nào là duy nhất trong hệ thống. Coad và Yourdon đã đưa ra 3 loại của sự kết dính cho hệ thống hướng đối tượng:
  - Sự kết dính phương thức.
  - Sự kết dính lớp.
  - Sự kết dính tổng quát hóa/chuyên môn hóa.

## Tiêu chuẩn thiết kế - sự kết dính (tt)

- **Sự kết dính phương thức:** đề cập đến sự kết dính bên trong một phương thức riêng lẻ, tức là, một phương thức như thế nào là duy nhất. Mỗi phương thức chỉ nên thực hiện một chức năng.
- Nói chung, sự kết dính phương thức nên là cực đại hóa (tăng đến mức tối đa).



## Tiêu chuẩn thiết kế - sự kết dính (tt)

- Có 7 loại kết dính phương thức sau sắp xếp từ tốt đến xấu:
  - Chức năng: Phương thức chỉ thực hiện một nhiệm vụ.
  - Tuần tự: Phương thức kết hợp 2 chức năng, mà output từ chức năng thứ nhất là input cho chức năng thứ hai.
  - Truyền thông: Phương thức kết hợp hai chức năng nhưng sử dụng cùng các thuộc tính để thực hiện.

# Tiêu chuẩn thiết kế - sự kết dính (tt)

- Thủ tục: Phương thức hỗ trợ nhiều mối quan hệ chức năng thấp.
- Temporal hoặc kinh điển: Phương thức hỗ trợ nhiều mối quan hệ chức năng in time.
- Logic:
- Trùng khớp: Mục đích của phương thức không thể định nghĩa.

# Tiêu chuẩn thiết kế - sự kết dính (tt)

- **Sự kết dính lớp:** là mức kết dính giữa các thuộc tính và phương thức của một lớp, có nghĩa là như thế nào là một lớp duy nhất. Một lớp chỉ nên biểu diễn một vấn đề, ví dụ, một employee, một department,... Có 3 loại:
  - Mixed-Role
  - Mixed-Domain
  - Mixed-Instance

## Tiêu chuẩn thiết kế - sự kết dính (tt)

- Glenford Meyers đề xuất một lớp kết dính nên:
  - Gồm nhiều phương thức có thể thấy bên ngoài lớp và mỗi phương thức hiện rõ chỉ nên thực hiện một chức năng.
  - Có những phương thức chỉ đề cập đến các thuộc tính hoặc các phương thức khác định nghĩa với lớp hoặc lớp cha của nó, và
  - Không có bất kỳ sự cố kết luồng sự kiện nào giữa các phương thức.

## Tiêu chuẩn thiết kế - sự kết dính (tt)

---

- Sự kết dính tổng quát hóa/chuyên môn hóa: đề cập đến như thế nào là một thứ tự kế thừa tốt. Đó là chỉ support ngữ nghĩa của tổng quát hóa, chuyên môn hóa.

# Tiêu chuẩn thiết kế - cùng sinh

- Cùng sinh (connascence) tổng quát hóa ý tưởng của sự kết hợp (cohesion) và sự kết dính (coupling) và kết hợp chúng thành gói. Để thực hiện, có ba mức đóng gói được xác định
  - Mức 0 đề cập đến tổng số lượng gói thực hiện trong một dòng lệnh riêng lẻ.
  - Mức 1 là mức của gói đạt được bởi việc nối các dòng lệnh thành một phương thức.
  - Mức 2 đạt được bởi việc thiết kế các lớp bao gồm cả phương thức và thuộc tính.

# Tiêu chuẩn thiết kế - cùng sinh (tt)

---

- Cùng sinh
  - Hai đơn vị (lớp và phương thức) có sự liên kết chặt chẽ với nhau (cùng sinh).

## Các ràng buộc và giao ước

---

- Một **ràng buộc (contract)** hình thức hóa sự tương tác giữa đối tượng client và server, nơi mà một đối tượng client là một thể hiện của lớp gửi thông điệp đến đối tượng server để thực thi một phương thức.
- Contract là một tập các **ràng buộc** và các giao ước (**guarantee**). Nếu gặp một ràng buộc, đối tượng server sẽ đảm bảo hành vi chắc chắn.



## Các ràng buộc và giao ước (tt)

- Các ràng buộc có thể được viết dưới ngôn ngữ tự nhiên (như English), Structured English, mã giả hoặc ngôn ngữ chính thức (như ngôn ngữ ràng buộc đối tượng của UML).
- Có ba loại ràng buộc điển hình trong thiết kế hướng đối tượng:
  - Pre-condition
  - Post-condition
  - Invariant (bất biến)

# Các ràng buộc và giao ước (tt)

- Pre-condition
  - Là một ràng buộc mà phải thỏa đối với một phương thức để thực hiện.
  - Ví dụ, các tham số truyền qua phương thức phải hợp lệ đối với phương thức, nếu không, phải chấm dứt ngoại lệ.

# Các ràng buộc và giao ước (tt)

- Post-condition

- Là một ràng buộc mà phải thỏa sau khi thực hiện phương thức hoặc kết quả của sự thực hiện phương thức phải được gỡ bỏ.
- *Ví dụ:* một phương thức không thể thực hiện khi bất kỳ một thuộc tính nào của đối tượng nhận một giá trị không phù hợp.

# Các ràng buộc và giao ước (tt)

- Invariant

- Mô hình pre-condition và post-condition ràng buộc trên các phương thức riêng lẻ thì mô hình ràng buộc invariant phải luôn luôn đúng cho tất cả các thể hiện của lớp.
- Chúng ta có thể gắn sự bất biến vào thẻ **CRC** hoặc **sơ đồ lớp** bởi việc thêm vào tập các khẳng định.

# Các ràng buộc và giao ước

## - Các thành phần của một ràng buộc

|                                  |             |     |
|----------------------------------|-------------|-----|
| Method Name:                     | Class Name: | ID: |
| Clients (Customers)              |             |     |
| Associated Use Cases             |             |     |
| Description of Responsibilities: |             |     |
| Arguments Received:              |             |     |
| Type of Value Returned:          |             |     |
| Pre-condition:                   |             |     |
| Post-condition:                  |             |     |

# Sự đặc tả phương thức

- **Cú pháp**

- Không có một cú pháp chính thức nào cho đặc tả phương thức, do đó nhiều tổ chức sử dụng dạng mẫu của chính họ.
- Một mẫu đặc tả phương thức điển hình gồm bốn thành phần:
  - Thông tin tổng quát
  - Sự kiện
  - Thông điệp
  - Sự đặc tả thuật toán

# Sự đặc tả phương thức (tt)

|   |             |           |
|---|-------------|-----------|
| Method Name:                                      | Class Name: | ID        |
| Contract ID:                                      | Programmer: | Date Due: |
| Programming Language:<br>q Visual Basic<br>q Java |             |           |
| Triggers/Events:                                  |             |           |
| Arguments Received:                               | Notes:      |           |
| Date Type:  |             |           |
|   |             |           |
|   |             |           |
| Messages Sent & Arguments Passed:                 | Date Type:  | Notes:    |
| ClassName.MethodName:                             |             |           |
|   |             |           |
|   |             |           |
|   |             |           |
|   |             |           |
|   |             |           |
|   |             |           |
|   |             |           |

|  |             |           |
|--|-------------|-----------|
| Method Name:   | Class Name: | ID        |
| Contract ID:   | Programmer: | Date Due: |
| Programming Language:<br>q Visual Basic<br>q Java          |             |           |
| Triggers/Events:   |             |           |
| Arguments Received:<br>Date Type:                          | Notes:      |           |
|  |             |           |
|  |             |           |
|  |             |           |
|  |             |           |
|  |             |           |
| Messages Sent & Arguments Passed:<br>ClassName.MethodName: | Date Type:  | Notes:    |
|  |             |           |
|  |             |           |
|  |             |           |
|  |             |           |
|  |             |           |
|  |             |           |
| Argument Returned:<br>Data Type:                           | Notes:      |           |
|  |             |           |
| Algorithm Specification:                                   |             |           |
| Misc.Notes   |             |           |



# Sự đặc tả phương thức (tt)

---

- Thông tin tổng quát, gồm
  - Tên phương thức
  - Tên lớp
  - Số ID
  - Contact ID
  - Programmer
  - Date due
  - Ngôn ngữ lập trình

# Sự đặc tả phương thức (tt)

---

- Sự kiện:
  - Một sự kiện là một hành động xảy ra hoặc takes place.

# Sự đặc tả phương thức (tt)

---

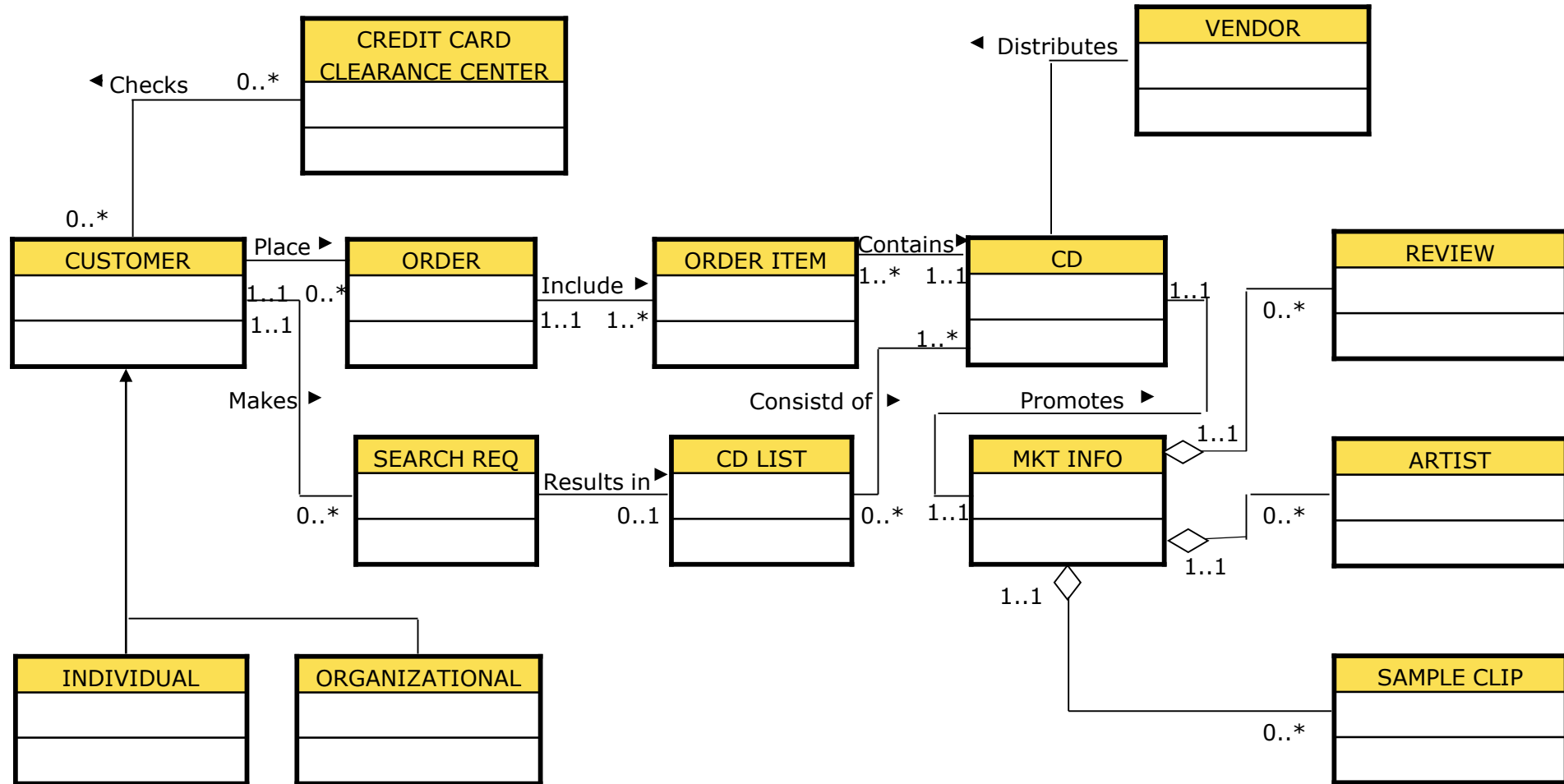
- Message Passing:
  - Giai đoạn tiếp theo của sự đặc tả phương thức mô tả message passing đi hoặc đến phương thức, được xác định trên sơ đồ cộng tác và sơ đồ tuần tự.
  - Những thông tin đề cập đến signature của phương thức, dữ liệu nhận, dữ liệu chuyển đến phương thức khác và dữ liệu trả lại được lưu lại.

# Sự đặc tả phương thức (tt)

---

- Đặc tả thuật toán:
  - Sự đặc tả thuật toán có thể được viết trong Structured English, form giải mã, hoặc ngôn ngữ formal.

# Ví dụ



Revised CD Selections Internet Sales System Class Diagram(Place Order Use-Case View)

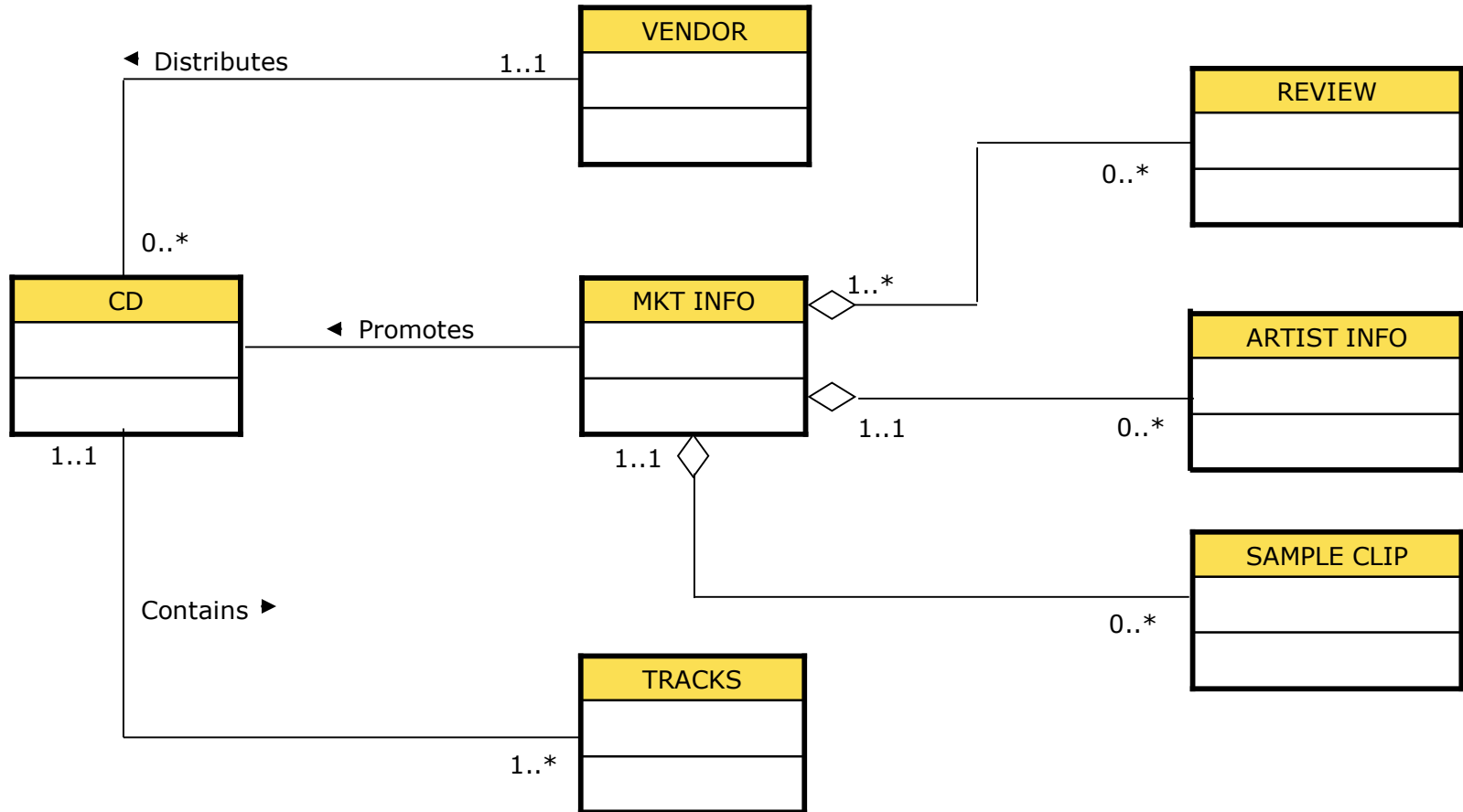
## Ví dụ (tt)

|   |             |     |
|---|-------------|-----|
| Method Name: GetReview()  | Class Name: | ID: |
| Clients (Customers): CD Detailed Report   |             |     |
| Associated Use Cases: Places Order  |             |     |
| Description of Responsibilities:<br>Return review objects for the Detailed Report Screen to display |             |     |
| Arguments Received:   |             |     |
| Type of Value Returned: List of Review objects  |             |     |
| Pre-condition: Review attribute not Null  |             |     |
| Post-condition:   |             |     |

Get Review Method Contract

|  |                        |                        |        |           |
|--|------------------------|------------------------|--------|-----------|
| Method Name: Get Review()  |                        | Class Name: MKT Info   |        | ID:435    |
| Contract ID: 89  |                        | Programmer: John Smith |        | Date Due: |
| Programming Language:<br><div>Visual Basic ;Java</div>                                       |                        |                        |        |           |
| Triggers/Events: Detail Button on Basic Report is pressed                                    |                        |                        |        |           |
| Arguments Received:  |                        | Notes:                 |        |           |
| Date Type:   |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
| Messages Sent & Arguments Passed:  |                        | Date Type:             | Notes: |           |
| ClassName.MethodName:  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
|  |                        |                        |        |           |
| Argument Returned:   |                        | Notes:                 |        |           |
| Data Type:   |                        |                        |        |           |
| List   | List of Review objects |                        |        |           |
| Algorithm Specification:<br>If Review Not Null<br>Return Review<br>Else Throw Null Exception |                        |                        |        |           |
| Misc.Notes   |                        |                        |        |           |

## Ví dụ (tt)



Revised Package Diagram for the CD Package on the PD Layer of CD Selection Internet Sales System