

Phân tích thiết kế hướng đối tượng

Bài 17: Các mẫu thiết kế (Design Patterns)

TS. Nguyễn Hiếu Cường

Bộ môn CNPM, Khoa CNTT

Trường ĐH GTVT

cuonggt@gmail.com

Mẫu thiết kế (Design patterns)

- Là những giải pháp để giải quyết một vấn đề nào đó
 - Không trình bày các ý tưởng thiết kế mới
 - Những tri thức, nguyên tắc trong các thiết kế đã có
- Mỗi mẫu thiết kế gồm:
 - Tên mẫu
 - Vấn đề (Problem)
 - Giải pháp (Solution)

GRASP Patterns

- Pattern = Problem/Solution
- GRASP - **G**eneral **R**esponsibility **A**ssignment **S**oftware **P**atterns
 - Các nguyên tắc cơ bản trong thiết kế và gán trách nhiệm các đối tượng được thể hiện dưới dạng các mẫu thiết kế
- Một số mẫu GRASP
 - Low coupling
 - High cohesion
 - Expert
 - Creator
 - Polymorphism
 - ...

Coupling

- Sự phụ thuộc của một lớp vào các đối tượng của các lớp khác
- Kết nối là cần thiết nếu muốn các đối tượng có thể truyền thông điệp cho nhau
- Nếu các lớp có kết nối mạnh
 - Dễ bị tác động bởi các lớp liên quan
 - Khó hiểu, khó bảo trì và khó sử dụng lại
- Làm sao để duy trì kết nối yếu?

Low coupling

- Problem
 - Làm sao để hỗ trợ kết nối giữa các thành phần là yếu?
- Solution
 - Gán các trách nhiệm sao cho một lớp cần phải biết ít lớp khác
- Ví dụ:
 - Có 3 lớp dưới đây, cần tạo các liên kết thế nào để có thể thanh toán

Payment

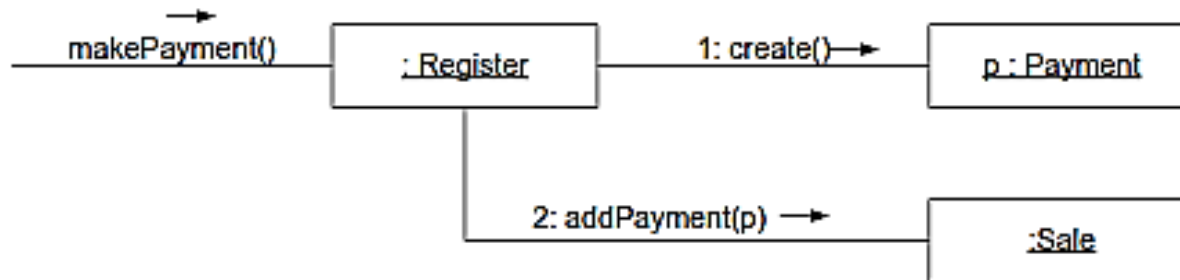
Register

Sale

Ví dụ

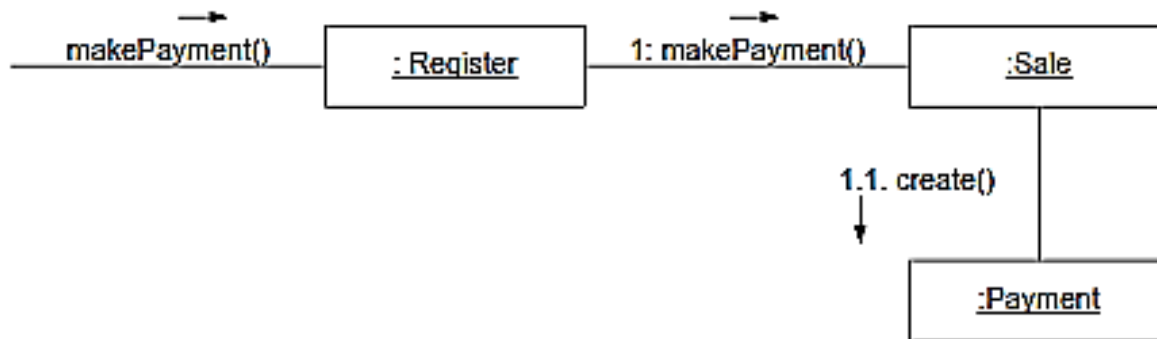
- Phương án 1

- Thêm kết nối giữa Register và Payment → làm tăng thêm kết nối



- Phương án 2

- Kết nối yếu!

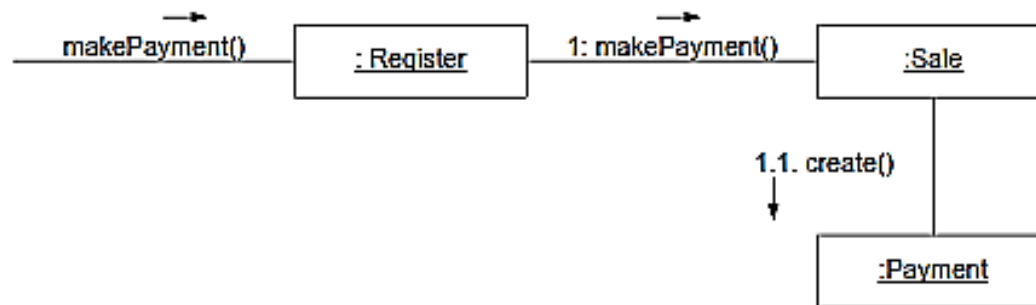


Cohesion

- Xác định mức độ liên hệ giữa các trách nhiệm trong một lớp.
- Một lớp có sự cố kết yếu
 - Có nhiều trách nhiệm không liên quan đến nhau hoặc có quá nhiều trách nhiệm
 - Lớp khó hiểu, khó bảo trì và khó sử dụng lại

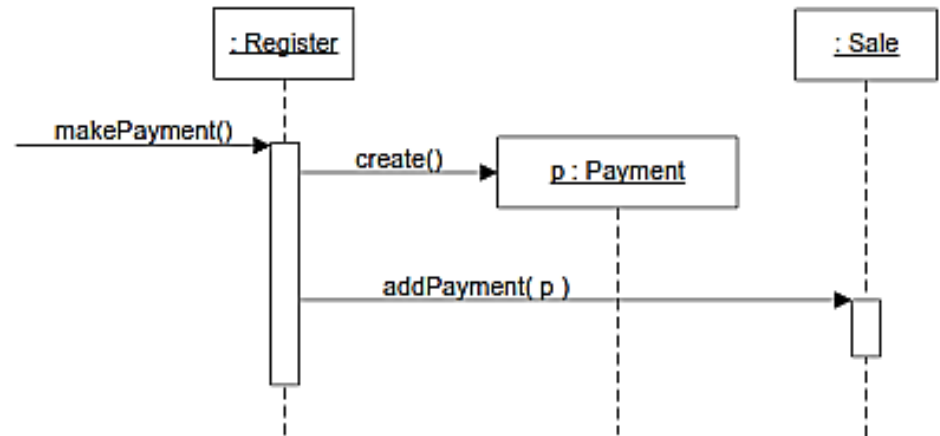
High cohesion

- Problem
 - Làm sao giữ cho lớp được cố kết mạnh?
- Solution
 - Gán trách nhiệm sao cho các lớp không phải làm quá nhiều việc hoặc nhiều việc khác nhau
- Ví dụ:
 - Tạo thêm một đối tượng Payment?

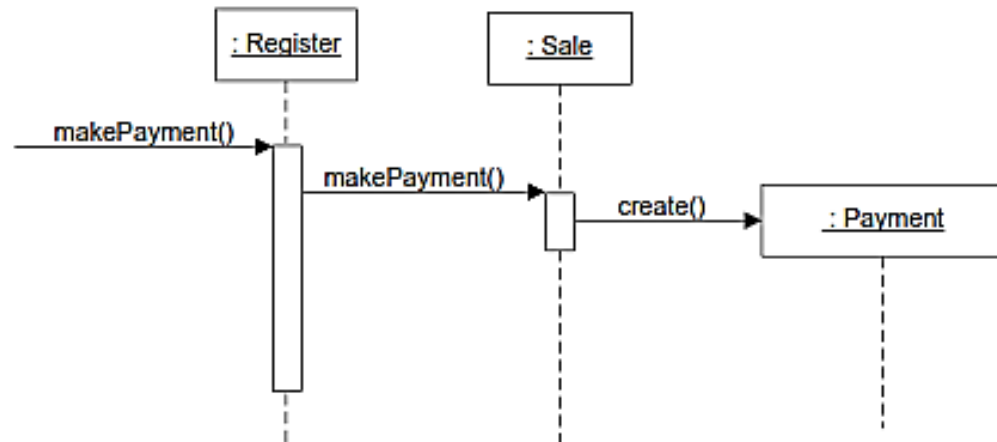


Ví dụ

- Phương án 1
 - Lớp Register làm nhiều việc



- Phương án 2
 - Ủy thác cho lớp Sale
 - Mỗi lớp có tính cố kết cao hơn



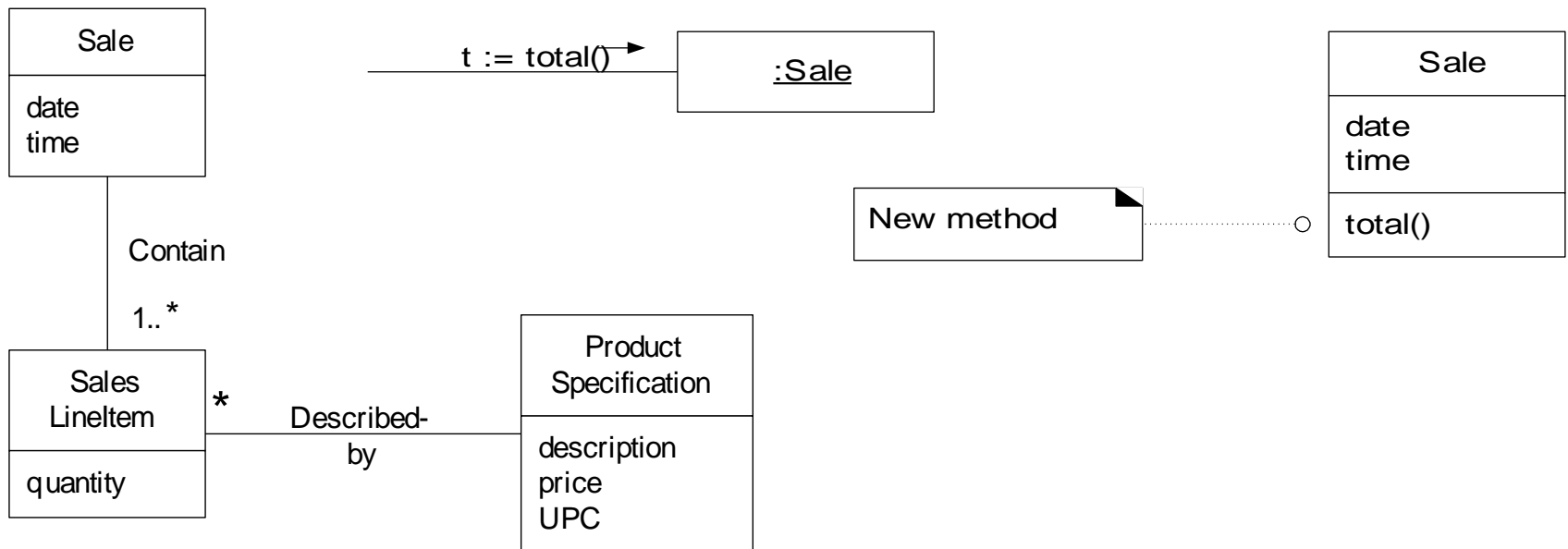
Expert

- Problem
 - Nguyên tắc chung để gán các trách nhiệm cho các đối tượng là gì?
- Solution
 - Gán trách nhiệm cho lớp có đủ *thông tin cần thiết* để hoàn thành trách nhiệm đó

Ví dụ

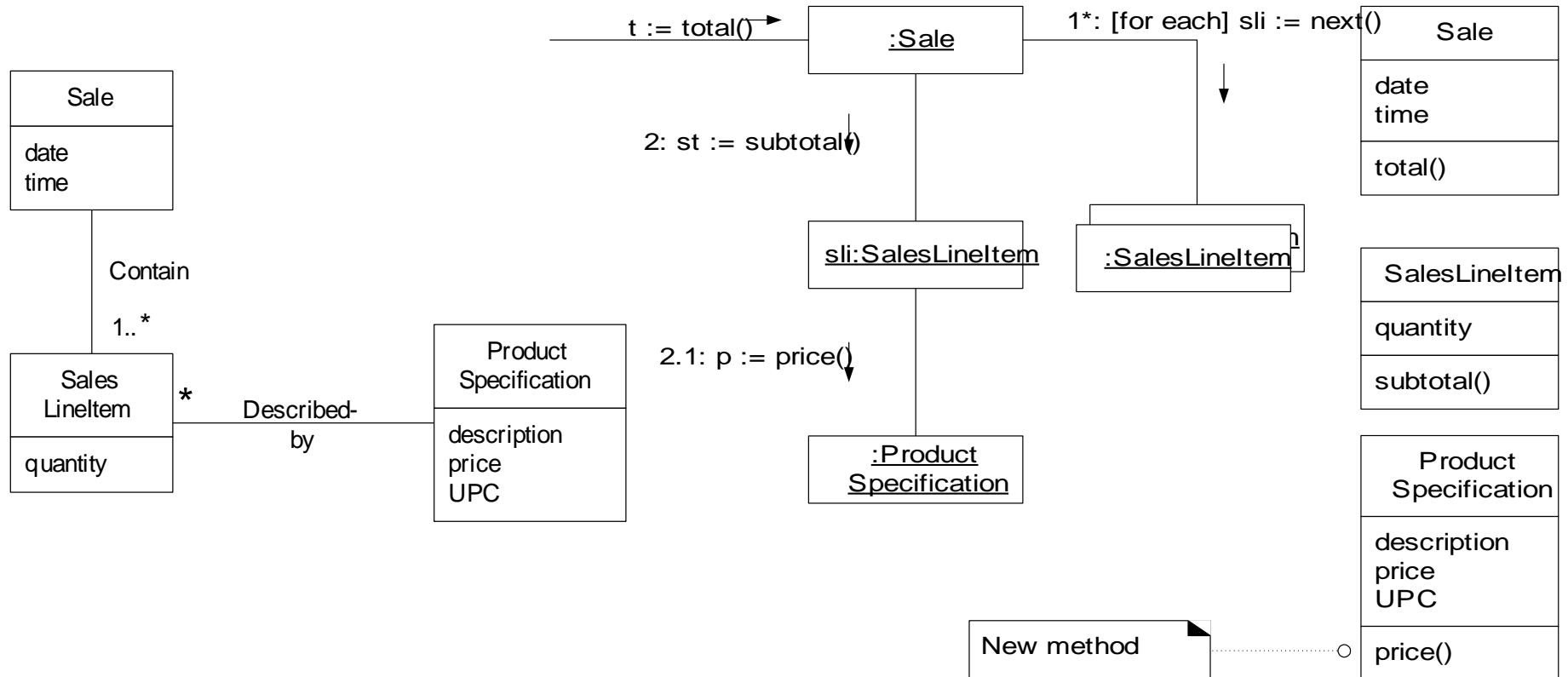
Lớp đối tượng nào có trách nhiệm (đủ thông tin) để biết **tổng số tiền** bán hàng trong ứng dụng?

Sale có đủ thông tin để tính tổng số tiền (Sale là “information expert”)



Ví dụ

Lớp nào biết **giá của mỗi sản phẩm**? Trong mỗi phiên cần tính số tiền (subtotal). Lớp nào có trách nhiệm biết **số tiền bán của mỗi phiên**?



Creator

- Problem

- “Ai” có trách nhiệm tạo ra một thực thể mới của một lớp đối tượng?

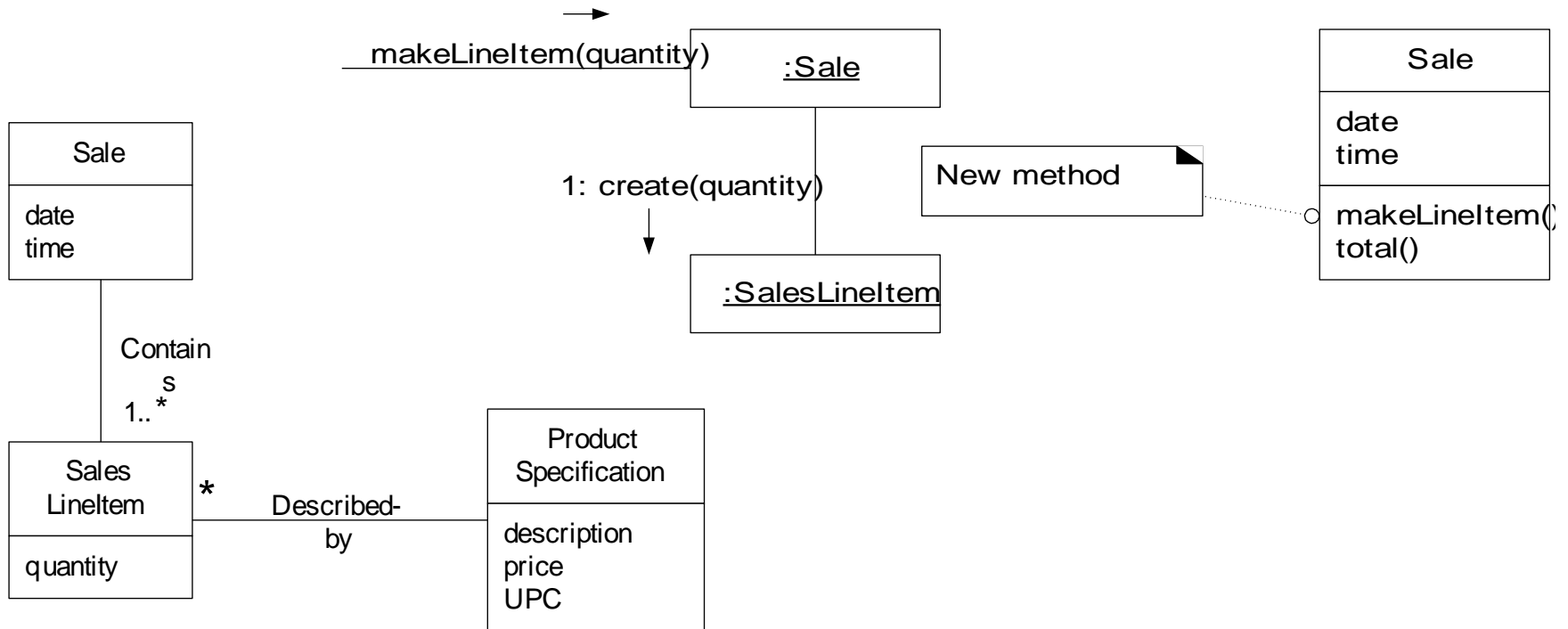
- Solution

- Lớp A có trách nhiệm tạo một thực thể của một lớp B dựa trên quan hệ giữa lớp tạo thực thể A (creator) và lớp được tạo thực thể B
 - A bao gồm các đối tượng B
 - A sử dụng các đối tượng B

Ví dụ

Lớp nào có trách nhiệm tạo các đối tượng của SalesLineItem?

Lớp **Sale** vì: Sale *contains* SalesLineItems



Tóm tắt

- Mẫu thiết kế (Design Patterns)
- Một số mẫu thiết kế GRASP