**Chương 9: Class and Method Design**

**Tổng quan**

- **lớp, đối tượng, phương thức thiết kế ...**

**- tối ưu hóa thiết kế đối tượng**

**- việc tái sử dụng các lớp, thư viện …**

**- xác định những rằng buộc và hợp đồng.**

**- đặc tả phương thức**

**- kỹ thuật hữu ích cho thiết kế phương thức: hợp đồng và thông số kỹ thuật.**

**Classes, Objects, Methods, and Messages**

- khối xây dựng cơ bản của một hệ thống là đối tượng, đối tượng là các trường hợp của một lớp-sử dụng khuôn mẫu để định nghĩa một đối tượng

- đối tượng có các thuộc tính mô tả dữ liệu về đối tượng, và các trạng thái, phương thức

- Một thông điệp là cốt lõi một hàm hoặc gọi thủ tục từ một đối tượng đến một đối tượng khác.

Encapsulation and Information Hiding - Đóng gói và Ẩn thông tin

- Đóng gói là cơ chế kết hợp các quy trình và dữ liệu vào một đối tượng duy nhất.

- Ẩn thông tin chỉ cho thấy các thông tin cần thiết để sử dụng một đối tượng có sẵn bên ngoài đối tượng.

- các đối tượng giao tiếp với nhau thông qua các phướng thức của đối tượng

**Đa hình và kết tập: ...**

**Thừa kế: ...**

**Lạm dụng kế thừa**

Coupling-Khớp nối

- dùng để chỉ các mô-đun (các lớp, đối tượng, và phương thức) nằm trong một hệ thống. Sự phụ thuộc lẫn nhau càng cao thì sự thay đổi nhiều hơn

- một phần của thiết kế có thể làm thay đổi được yêu cầu trong các phần khác của thiết kế

- khớp nối có thể không phải là một điều tốt

- các loại tương tác khớp nối: hình 9-7

**Hoạt động thiết kế đối tượng**

**Adding Specifications - thêm thông số kỹ thuật**

- trong sự phát triển của hệ thống, điều quan trọng là phải xem xét các mô hình chức năng, cấu trúc và hành vi hiện tại.

+ Thứ nhất, chúng ta phải đảm bảo rằng các lớp là cần thiết và đủ để giải quyết vấn đề tiềm ẩn. Để làm được điều này, chúng ta cần phải chắc chắn rằng không có thuộc tính hoặc phương thức nào thiếu và không có các thuộc tính hoặc phương thức bổ sung hoặc không sử dụng trong mỗi lớp.

+ Thứ hai, chúng ta cần phải hoàn thiện khả năng hiển thị (ẩn hoặc nhìn thấy được) của các thuộc tính và meth-ods trong mỗi lớp học. Tùy thuộc vào ngôn ngữ lập trình hướng đối tượng được sử dụng, điều này có thể được xác định trước.

+ Thứ ba, chúng ta cần quyết định chữ ký (tên phương thức) của mọi phương thức trong mỗi lớp. Các Signa-ture của một phương thức bao gồm ba phần: tên của phương thức này, các tham số hoặc argu-ments đó phải được truyền cho phương thức, trong đó có loại đối tượng của họ, và loại giá trị mà phương thức này sẽ trả lại với phương thức gọi đến. Chữ ký của một phương thức có liên quan đến hợp đồng của phương thức.

+ Thứ tư, chúng ta cần xác định bất kỳ ràng buộc nào phải được bảo vệ bởi các đối tượng (ví dụ: thuộc tính của một đối tượng chỉ có thể có giá trị trong một dải nhất định). Có ba loại ràng buộc khác nhau: điều kiện tiên quyết, điều kiện hậu và bất biến.

- Những thứ này được ghi lại trong hình thức hợp đồng và các khẳng định được thêm vào thẻ CRC và sơ đồ lớp. Chúng ta cũng phải quyết định làm thế nào để xử lý một sự vi phạm của một hạn chế. Nên hệ thống đơn giản là hủy bỏ? Nếu hệ thống tự động hoàn tác thay đổi gây ra sự vi phạm? Nếu hệ thống cho phép người dùng cuối xác định cách tiếp cận để sửa đổi sự viola-tion? Nói cách khác, nhà thiết kế phải thiết kế các lỗi mà hệ thống mong muốn han-dle. Tốt nhất là không để các loại quyết định thiết kế này để lập trình viên giải quyết.

**Identifying Opportunities for Reuse - Xác định cơ hội để tái sử dụng**

- Trong thiết kế, ngoài việc sử dụng các mẫu phân tích, có cơ hội sử dụng các mẫu thiết kế, framework-khuôn khổ, thư viện, và các thành phần. Các cơ hội sẽ khác nhau, tùy thuộc vào lớp nào đang được xem xét

- Một framework bao gồm một tập các lớp được thực hiện có thể được sử dụng như là một cơ sở để thực hiện một ứng dụng, cho phép chúng ta tạo lớp con để kế thừa từ các lớp trong framework. khi kế thừa từ các lớp trong một framework, chúng ta đang tạo ra một sự phụ thuộc (nghĩa là gia tăng sự nối kết thừa kế từ lớp con đến tầng lớp cha). Vì vậy, nếu chúng tôi sử dụng một framework và nhà cung cấp làm thay đổi framework, chúng ta sẽ phải có ít nhất biên dịch lại hệ thống khi nâng cấp lên phiên bản mới của framework.

- thư viện lớp tương tự như một framework trong đó nó thường có một bộ thực hiện các lớp học được thiết kế để tái sử dụng …

**Restructuring the Design-tái cấu trúc thiết kế**

- Một khi các lớp và phương thức cá nhân đã được chỉ định và các thư viện lớp, khung công việc và các thành phần đã được kết hợp với thiết kế đang phát triển, chúng ta nên sử dụng Factoring để cơ cấu lại thiết kế.

Factoring là quá trình tách biệt ngoại cảnh của một phương thức hoặc lớp thành một phương thức hay lớp học mới để đơn giản hóa thiết kế tổng thể

**Optimizing the Design-tối ưu hóa thiết kết**

- tăng tính dễ hiểu của một thiết kế thường tạo ra một thiết kế không hiệu quả. Ngược lại, tập trung vào vấn đề hiệu quả sẽ cung cấp một thiết kế đó là khó hiểu hơn.

Một số phương thức tối ưu hóa thiết kế (Chỉ mang tính chất gợi ý): slide-338

Mapping Problem-Domain Classes to Implementation Languages - Lập bản đồ miền giá trị cho lớp cho các ngôn ngữ thục hiện

**CONSTRAINTS AND CONTRACTS - Rằng buộc và hợp đồng**

- một hợp đồng là một tập hợp các ràng buộc và đảm bảo

**Types of Constraints**

- Có ba loại Constraints khác nhau thường được nắm bắt trong thiết kế hướng đối tượng: điều kiện tiên quyết (tiền điều kiện), điều kiện sau (hậu điều kiện) và bất biến.

- Các hợp đồng được sử dụng chủ yếu để thiết lập các tiền điều kiện và hậu điều kiện

+ tiền điều kiện là một hạn chế cần phải đáp ứng cho một phương thức để thực hiện

+ hậu điều kiện là một con-straint phải được đáp ứng sau khi phương thức được thực hiện, hoặc hiệu quả của việc thực hiện phương thức phải được hoàn tác

+ bất biến - các ràng buộc mô hình phải luôn luôn đúng đối với tất cả các thể hiện của một lớp, có thể đính kèm bất biến vào CRC thẻ hoặc biểu đồ lớp bằng cách thêm một tập hợp các khẳng định cho nó. Hình minh họa - Hình 9-17.

- Ý tưởng sử dụng hợp đồng trong thiết kế phát triển từ kỹ thuật "Thiết kế theo hợp đồng" do Bertrand phát triển.

**Elements of a Contract - Các yếu tố của hợp đồng**

- Về mặt kỹ thuật nói, cần tạo một hợp đồng cho mỗi tin nhắn được gửi và nhận bởi mỗi đối tượng; một cho mỗi tương tác. Tuy nhiên, sẽ có một số trùng lắp nếu điều này là làm xong. Trong thực tế, hợp đồng được tạo cho mỗi phương thức có thể nhận tin nhắn từ các đối tượng khác (tức là một đối tượng cho mỗi phương thức hiển thị).

- Một hợp đồng nên chứa các thông tin cần thiết cho một lập trình để hiểu phương thức là làm gì (nghĩa là chúng có tính khai báo). Thông tin này bao gồm tên phương thức, tên lớp, số ID, đối tượng khách hàng, trường hợp sử dụng liên quan, mô tả, các đối số đã nhận, loại dữ liệu được trả về, và các điều kiện trước và sau.

- một hợp đồng được đặt ra cho các thông tin cần thiết cho nhà phát triển của một đối tượng khách hàng biết những thông điệp nào có thể được gửi tới các đối tượng máy chủ và những gì khách hàng có thể mong đợi.

**METHOD SPECIFICATION - Đặc tả phương thức**

- Một khi nhà phân tích đã truyền đạt được bức tranh toàn cảnh về cách hệ thống cần phải được đưa ra cùng với nhau, người đó cần mô tả các lớp học và phương thức cá nhân với đầy đủ chi tiết rằng các lập trình viên có thể tiếp quản và bắt đầu viết code

- Phương thức trên thẻ CRC, lớp sơ đồ và hợp đồng được mô tả bằng cách sử dụng chi tiết kỹ thuật. Thông số kỹ thuật là tài liệu văn bản bao gồm hướng dẫn rõ ràng về cách viết mã để thực hiện phương thức. Thông thường, các thành viên nhóm dự án viết một đặc tả cho từng phương thức và sau đó vượt qua tất cả cùng với các lập trình, những người viết mã trong quá trình thực hiện các dự án