**Chương 9: Class and Method Design**

**Classes, Objects, Methods, and Messages – Lớp học, đối tượng, phương thức và thông điệp**

- 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

**DESIGN CRITERIA – Tiêu chuẩn thiết kế**

-Khi xem xét việc thiết kế một hệ thống hướng đối tượng, một bộ tiêu chí tồn tại có thể được sử dụng để xác định liệu thiết kế là tốt hay xấu

**Coupling - Khớp nối**

- 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

- xác định hai loại khớp nối để xem xét: tương tác và thừa kế.

**-**  Khớp nối tương tác đề cập đến sự ghép nối giữa các phương thức và các đối tượng thông qua message đi qua

- Về cơ bản, luật giảm thiểu số lượng các đối tượng có thể nhận tin nhắn từ một đối tượng nhất định. Luật quy định rằng một đối tượng nên gửi message chỉ với một trong những điều sau đây:

+ Chính nó (một đối tượng có thể gửi message đến chính nó)

+ Một đối tượng được chứa trong một thuộc tính của đối tượng hoặc một trong các siêu lớp của nó.

+ Một đối tượng được truyền như một tham số cho phương thức

+ Một đối tượng được tao ra bằng phương thức

+ Một đối tượng được lưu trữ trong một biến toàn cục

- Khớp nối thừa kế, như tên của nó ngụ ý, đề cập đến sự chặt chẽ của các lớp học trong một hệ thống phân cấp thừa kế.

- Từ góc độ thiết kế, nên tránh các biến toàn cục. Hầu hết các chương trình hướng đối tượng thuần túy lan-guages ​​không hỗ trợ rõ ràng các biến toàn cục

**Cohesion – Sự gắn kết**

**-** Sự gắn kết đề cập đến làm thế nào duy nhất minded một mô-đun (lớp học, đối tượng, hoặc phương thức) là trong một hệ thống. Một lớp hay một đối tượng chỉ nên đại diện cho một điều và một phương thức chỉ nên giải quyết một nhiệm vụ duy nhất

**-** Các phương thức nên làm một và chỉ có một điều. Một phương thức thực sự thực hiện nhiều chức năng khó hiểu hơn – và do đó để thực hiện và duy trì tốt hơn là chỉ thực hiện một chức năng duy nhất

- Sự gắn kết lớp học là mức độ gắn kết giữa các thuộc tính và phương thức của một lớp

**Connascence – Sự Sự khép kín**

- Sự Sự khép kín khái quát hoá các ý tưởng về sự gắn kết và khớp nối, và nó kết hợp chúng với các lập luận cho đóng gói

- Mức đóng gói mức 0 đề cập đến số lượng đóng gói được nhận ra trong mỗi dòng mã riêng lẻ, mức đóng gói cấp 1 là mức độ đóng gói đạt được bởi các dòng mã kết hợp vào một phương pháp và mức đóng gói mức 2 đạt được bằng cách tạo ra các lớp chứa cả phương pháp và thuộc tính. Sự gắn kết phương pháp và địa chỉ khớp nối tương tác chủ yếu là mức độ đóng gói mức 1. Sự gắn kết lớp học, khái quát hóa / chuyên môn hoá, và địa chỉ liên kết kế thừa chỉ đóng gói cấp độ 2

**-** Sự khép kín, như là một sự tẩy xóa liên kết và khớp nối gen, giải quyết cả việc đóng gói mức 1 và cấp 2

**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 – các loại rằng buộc**

- 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

**Events – Sự kiện**

- Sự kiện là một điều xảy ra hoặc diễn ra. Nhấp chuột tạo ra một sự kiện chuột, nhấn một phím tạo ra một sự kiện keystroke-trên thực tế, hầu như mọi thứ người dùng tạo ra một sự kiện xảy ra.

- chương trình hướng sự kiện bao gồm các phương pháp được thực hiện để đáp ứng với một sự kiện do người dùng, hệ thống hoặc một phương pháp khác khởi xướng. Sau khi khởi tạo, hệ thống sẽ đợi một sự kiện xảy ra. Khi đó, một phương pháp được kích hoạt để thực hiện công việc thích hợp và sau đó hệ thống sẽ đợi một lần nữa.  
- có nhiều lập trình viên vẫn sử dụng các thông số kỹ thuật của phương pháp khi lập trình bằng ngôn ngữ lập trình dựa trên sự kiện và chúng bao gồm phần sự kiện trên mẫu để nắm bắt khi phương pháp được gọi

**Message Passing – Thông qua**

**-** Các phần tiếp theo của đặc tả phương pháp mô tả thông điệp đi qua và từ phương thức, được xác định trên sơ đồ trình tự và phối hợp. Các lập trình viên sẽ cần phải hiểu những gì các đối số đang được chuyển vào, từ và trả về bởi phương thức bởi vì các đối số cuối cùng sẽ dịch thành các thuộc tính và cấu trúc dữ liệu trong phương pháp thực tế.

**Algorithm Specification – Thuật toán**

- Thông số thuật toán có thể được viết bằng Tiếng Anh có cấu trúc, một số dạng giả mã, hoặc một số loại ngôn ngữ chính thức.

+ Tiếng Anh cấu trúc chỉ đơn giản là một cách chính thức để viết các hướng dẫn mô tả các bước của quy trình. Bởi vì đây là bước đầu tiên hướng tới thực hiện phương pháp, nó trông giống như một ngôn ngữ lập trình đơn giản. Tiếng Anh cấu trúc sử dụng các câu ngắn mô tả rõ ràng chính xác công việc được thực hiện như thế nào  
dữ liệu

+ Pseudocode-mã giả là một ngôn ngữ chứa các cấu trúc hợp lý, bao gồm các mô tả trình tự, câu lệnh có điều kiện và lặp lại. Nó khác với tiếng Anh có cấu trúc trong pseudocode đó có chứa các chi tiết được lập trình cụ thể, chẳng hạn như các hướng dẫn khởi tạo hoặc liên kết, và nó cũng rộng hơn để người lập trình có thể viết mô đun này bằng cách phản chiếu các hướng dẫn giả mạo. Nói chung, mã giả là nhiều hơn như mã thực, và khán giả của nó là lập trình viên, trái với nhà phân tích. Định dạng của nó không phải là thông số kỹ thuật

**Thẻ đặc tả phương thức:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên phương thức: timKiemBaiViet | Tên lớp: BaViet | ID: 1 | |
| ID Hợp đồng: 2 | Lập trình viên: BK-FS | Hạn: 8/12/2017 | |
| Ngôn ngữ lập trình:  Visual Basic Smalltalk C++ Java Php | | | |
| Triggers / Sự kiện:  Nhập dữ liệu vào form tìm kiếm và nhấn nút tìm kiếm | | | |
| Đối số đã nhận:  Kiểu dữ liệu: | Ghi chú: | | |
| String | Nội dung người dùng nhập vào form tìm kiếm | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| Thông điệp đã gửi & Đối số truyền qua:  TenLop.TenPhuongThuc: | Kiểu dữ liệu: | | Ghi chú: |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
| Đối số trả về:  Kiểu dữ liệu: | Ghi chú: | | |
| Mảng | Mảng danh sách các đối tượng xuất hiện nội dung người dùng muốn tìm kiếm | | |
| Thuật toán:  Chuyển phần nội dung người dùng nhập và phần tiêu đê, tóm tắt, nội dung của dũ liệu tin tức trong cơ sở dữ liệu thành một chuỗi liền không dấu, không có chữ hoa. Sau đó so sánh phần nội dung nhập với dữ liệu tin tức mình đã chuyển đổi và đưa ra kết quả. | | | |
| Ghi chú khác: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên phương thức: taoBaiViet | Tên lớp: BaViet | ID: 1 | |
| ID Hợp đồng: 1 | Lập trình viên: BK-FS | Hạn: 8/12/2017 | |
| Ngôn ngữ lập trình:  Visual Basic Smalltalk C++ Java Php | | | |
| Triggers / Sự kiện:  Nhập dữ liệu vào form tạo bài viết và nhấn nút “tạo bài viết” | | | |
| Đối số đã nhận:  Kiểu dữ liệu: | Ghi chú: | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| Thông điệp đã gửi & Đối số truyền qua:  TenLop.TenPhuongThuc: | Kiểu dữ liệu: | | Ghi chú: |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
| Đối số trả về:  Kiểu dữ liệu: | Ghi chú: | | |
| Boolean | Hoàn thành hoặc thất bại | | |
| Thuật toán: | | | |
| Ghi chú khác: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên phương thức: hienThiBaiViet | Tên lớp: BaViet | ID: 3 | |
| ID Hợp đồng: 3 | Lập trình viên: BK-FS | Hạn: 8/12/2017 | |
| Ngôn ngữ lập trình:  Visual Basic Smalltalk C++ Java Php | | | |
| Triggers / Sự kiện: | | | |
| Đối số đã nhận:  Kiểu dữ liệu: | Ghi chú: | | |
| int | Mã (id) bài viết | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| Thông điệp đã gửi & Đối số truyền qua:  TenLop.TenPhuongThuc: | Kiểu dữ liệu: | | Ghi chú: |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
| Đối số trả về:  Kiểu dữ liệu: | Ghi chú: | | |
| Đối tượng | Đối tượng một bài viết | | |
| Thuật toán: | | | |
| Ghi chú khác: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên phương thức: taoQuanTriVien | Tên lớp: QuanTriVien | ID: 4 | |
| ID Hợp đồng: 4 | Lập trình viên: BK-FS | Hạn: 8/12/2017 | |
| Ngôn ngữ lập trình:  Visual Basic Smalltalk C++ Java Php | | | |
| Triggers / Sự kiện: | | | |
| Đối số đã nhận:  Kiểu dữ liệu: | Ghi chú: | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| Thông điệp đã gửi & Đối số truyền qua:  TenLop.TenPhuongThuc: | Kiểu dữ liệu: | | Ghi chú: |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
| Đối số trả về:  Kiểu dữ liệu: | Ghi chú: | | |
| Boolean | Thành công hoặc thất bại | | |
| Thuật toán: | | | |
| Ghi chú khác: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên phương thức: traLoiLienHe | Tên lớp: LienHe | ID: 5 | |
| ID Hợp đồng: 5 | Lập trình viên: BK-FS | Hạn: 8/12/2017 | |
| Ngôn ngữ lập trình:  Visual Basic Smalltalk C++ Java Php | | | |
| Triggers / Sự kiện: | | | |
| Đối số đã nhận:  Kiểu dữ liệu: | Ghi chú: | | |
| int | Mã (id) liên hệ của người dùng, thành viên | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| Thông điệp đã gửi & Đối số truyền qua:  TenLop.TenPhuongThuc: | Kiểu dữ liệu: | | Ghi chú: |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
|  |  | |  |
|  |  | |  |
|  | |  |
|  | |  |
| Đối số trả về:  Kiểu dữ liệu: | Ghi chú: | | |
| Đối tượng | Đối tượng liên hệ bao gồm: nội dung, ngày tạo, tiêu đề | | |
| Thuật toán: | | | |
| Ghi chú khác: | | | |