Seminar team iOS

Nội dung: Một số design pattern thường dùng trong iOS

Tác giả: Nguyễn Đức Hà

Ngày thực hiện: 23/06/2016

1. Giới thiệu về design pattern

Trong khi thiết kế các framework của Cocoa và Cocoa Touch, Apple đã áp dụng rất nhiều các design pattern. Một số design pattern tiêu biểu như MVC, Singleton, Delegation,... Tuy nhiên không phải ai cũng nhận ra những dòng code mình sử dụng đang áp dụng design pattern gì. Bài viết này sẽ trình bày các kiến thức cơ bản về design pattern nói chung và các design pattern thường được áp dụng trong lập trình iOS nói riêng, qua đó giúp lập trình viên hiểu và áp dụng các design pattern vào dự án của mình.

1. Khái niệm design pattern

Design pattern được nhắc đến lần đầu tiên trong cuốn sách *Design Patterns: Elements of Reusable Object-Oriented Software* của bốn tác giả: Erich Gamma, Richard Helm, Ralph Johnson và John Vlissides (thường được biết đến với tên “Gang of Four”). Định nghĩa vắn tắt của design pattern được đề cập trong cuốn sách là “Giải pháp cho một vấn đề trong một ngữ cảnh”. Ngữ cảnh ở đây được hiểu là tình huống thường xảy ra, có thể áp dụng design pattern vào đó. Vấn đề là mục tiêu cần đạt được trong ngữ cảnh đó. Như vậy có thể định nghĩa design pattern như sau: design pattern là giải pháp đã được chứng minh tính hiệu quả, có thể áp dụng cho các vấn đề chung, lặp lại trong một ngữ cảnh nhất định.

1. Các lợi ích của design pattern

* Sử dụng design pattern giúp cho việc trao đổi về dự án ngắn gọn và dễ hiểu hơn. PM chỉ cần nói sử dụng singleton là dev sẽ hiểu mà không cần giải thích quá nhiều.
* Tiết kiệm thời gian nghiên cứu với những vấn đề chung thường gặp phải. Design pattern đã cung cấp giải pháp cho những vấn đề đó, giải pháp đó đã được chứng minh là hiệu quả.
* Design pattern là giải pháp ở tầng thiết kế, nó có thể được áp dụng vào bất kì ngôn ngữ hướng đối tượng nào.
* Giúp cho quá trình cài đặt và bảo trì thuận tiện và dễ dàng hơn.

1. Phân loại design pattern

Design pattern được chia làm 3 loại, tương ứng với chức năng của nó: creational patterns, structural patterns, behavior patterns.

1. Creational patterns

Các pattern thuộc loại này được dùng để khởi tạo một đối tượng nhưng che giấu phần logic khởi tạo và không sử dụng hàm khởi tạo trực tiếp (alloc init hoặc new). Sử dụng design pattern này giúp cho việc tạo đối tượng được linh hoạt hơn.

Một số creational pattern phổ biến: singleton, factory, abstract factory, builder,…

1. Structural patterns

Đây là các pattern liên quan đến quan hệ giữa các đối tượng. Nó giúp cho việc xác định quan hệ giữa các đối tượng trở nên đơn giản hơn.

Một số structural pattern phổ biến: adapter, facade, decorator, bridge,..

1. Behavior patterns

Các pattern này liên quan đến việc giao tiếp giữa các đối tượng, làm tăng tính linh hoạt khi thực hiện giao tiếp.

Một số behavior pattern phổ biến: observer, command, memento,…

1. Một số design pattern thường dùng trong iOS

Phần này của bài viết sẽ giới thiệu một số design pattern thường được sử dụng trong iOS, bao gồm định nghĩa, cách cài đặt và ý nghĩa của pattern đó.

1. Singleton pattern

Singleton pattern đảm bảo rằng một class chỉ có duy nhất một instance và có thể được truy cập một cách toàn cục (các class khác đều sử dụng chung một instance đó). Class luôn theo dõi instance duy nhất mà nó tạo ra và đảm bảo rằng không có thêm instance nào được tạo ra nữa.

Một số thư viện Cocoa Touch sử dụng singleton pattern: NSFileManager, NSURLSession, UIApplication, NSUserDefaults,…

Trường hợp có thể áp dụng singleton pattern: khi cần truy cập đến các tài nguyên toàn cục như setting, session.

Hạn chế khi sử dụng singleton: instance được tạo khi sử dụng singleton không thể được copy, retain và release.

Sơ đồ UML của singleton pattern:

Tạo singleton sử dụng Objective-C:

Tạo singleton sử dụng Swift 2:

1. Factory & Abstract Factory pattern
2. Facade
3. Decorator
4. Adapter
5. Observer