Nhận xét của giáo viên

Nội dung

[1. Tổng quan 3](#_Toc469737536)

[1.1. Giới thiệu 3](#_Toc469737537)

[1.2. Phạm vi báo cáo 3](#_Toc469737538)

[1.3. Các khái niệm OOP được sử dụng 3](#_Toc469737539)

[2. Giới thiệu về mẫu thiết kế 3](#_Toc469737540)

[2.1. Khái niệm mẫu thiết kế: 3](#_Toc469737541)

[2.2. Lịch sử ra đời 3](#_Toc469737542)

[2.3. Phân loại 4](#_Toc469737543)

[2.4. Lợi ích khi sử dụng mẫu 4](#_Toc469737544)

[2.5. Cấu trúc trình bày mẫu 6](#_Toc469737545)

[3. Các mẫu Thiết kế GoF 7](#_Toc469737546)

[3.1. Mẫu Singleton 7](#_Toc469737547)

[3.2. Mẫu Proxy 7](#_Toc469737548)

[3.3. Mẫu Template 7](#_Toc469737549)

[3.4. Mẫu Façade 7](#_Toc469737550)

[3.5. Mẫu Observer 7](#_Toc469737551)

[4. Tổng kết 8](#_Toc469737552)

[5. Tài liệu tham khảo 9](#_Toc469737553)

1. **Tổng quan** 
   1. **Giới thiệu**

Design Pattern có vai trò cực kì quan trọng trong ngành công nghiệp phần mềm. Đã từ lâu, việc thiết kế các chương trình theo hướng đối

* 1. **Phạm vi báo cáo**

Trong phạm vi báo cáo này trình bày về 5 mẫu DP có tần suất sử dụng tương đối cao, được áp dụng trong nhiều lĩnh vực khác nhau, bao gồm : Façade, Template, Observe, Proxy và Singleton.

* 1. **Các khái niệm OOP được sử dụng**

Trong quá trình mô tả các mẫu thiết kế, có khá nhiều các khái niệm trong OOP (lập trình hướng đối tượng) được sử dụng lại. Một số thuật ngữ tương đối khó hiểu và dễ nhầm lẫn khi chuyển đổi từ tiếng Anh sang tiếng Vệt cũng sẽ được nói sơ qua nhằm phục vụ cho nội dung báo cáo.

* Chém gió về hướng đối tượng:
  + Tài liệu sử dụng: Thinking in Java.

1. **Giới thiệu về mẫu thiết kế** 
   1. **Khái niệm mẫu thiết kế:**

Trong quá trình thiết kế phần mềm, chúng ta có thể gặp phải một vài vấn đề khó giải quyết, hoặc cách giải quyết của chúng không phù hợp, hoặc giải quyết vấn đề không trọn vẹn, chúng ta có thể nghĩ đến các mẫu thiết kế phần mềm. Nói chung, mẫu thiết kế là một giải pháp chung được các lập trình viên và các nhà phát triển đi trước tìm ra và tổng hợp lại, chúng thường được áp dụng trong việc giải quyết một số vấn đề thường xảy ra trong quá trình chúng ta thiết kế phần mềm.

Nói một cách khác, ta có thể coi một mẫu thiết kế như một khuôn mẫu mô tả cách thức giải quyết một vấn đề mà ta có thể sử dụng trong nhiều tình huống khác nhau.

* 1. **Lịch sử ra đời**

Khái niệm mẫu thiết kế mà chúng ta đang dùng có nguồn gốc trong lĩnh vực kiến trúc xây dựng. Một kỹ sư người Áo là Christopher Alexander (sinh năm 1936) đã nhận thấy có rằng có nhiều thiết kế được sử dụng nhiều lần trong quá trình thiết kế các công trình kiến trúc. Ông đã xuất bản hai quyển sách có tầm ảnh hưởng lớn là “A Pattern Language : Towns, Buildings, Construction” (xuất bản năm 1977) và “The Timeless Way of Building” (xuất bản năm 1979). Trong các quyển sách này, ông đã mô tả các mẫu thiết kế kiến trúc “Mỗi mẫu mô tả một vấn đề mà chúng xảy ra lặp đi lặp lại trong môi trường của chúng tôi. Đồng thời mô tả giải pháp cốt lõi để giải quyết vấn đề đó, theo một cách mà bạn có thể sử dụng giải pháp này hàng triệu lần mà không cần phải giải quyết một vấn đề giống nhau hai lần.

Gần hai thập kỉ sau, các chuyên gia trong lĩnh vực phần mềm bắt đầu kết hợp các nguyên lý của Alexander và việc tạo ra các tài liệu hướng dẫn đầu tiên về mẫu thiết kế cho các nhà phát triển mới. Vào năm 1994, hội nghị đầu tiên về các mẫu thiết kế được tổ chức. Tên của hội nghị là Pattern Languages of Program Design (PLoP), tạm dịch là “Các ngôn ngữ mô mẫu của thiết kế chương trình”. Không lâu sau đó (1995), cuốn sách có tầm ảnh hưởng lớn nhất đến các mẫu thiết kế phần mềm là *Design Patterns: Elements of Reusable Object-Oriented Software* đã được xuất bản, nó giới thiệu 23 mẫu thiết kế đầu tiên, do bốn người Erich Gamma, Richard Helm, Ralph Johnson và John Vlissides đồng tác giả. Các tác giả sau này thường được nhắc đến với biệt danh *Gang of Four*  hay GoF. Kể từ đó, càng ngày càng có nhiều mẫu thiết kế được ghi chép và phân loại. Tuy nhiên, 23 mẫu GoF đầu tiên được biết đến và được ứng dụng nhiều nhất.

* 1. **Phân loại**

Các mẫu thiết kế được chia thành 3 loại chính: Mẫu kiến tạo, mẫu kiến trúc và mẫu hành vi

1. *Mẫu kiến tạo:*

Các mẫu kiến tạo có mục đích chung là.

Bao gồm các mẫu:

* Abstract Factory: mô tả một dòng họ …
* Factory Method: mô tả …
* Singleton:
* Prototype:
* Builder:

1. *Mẫu kiến trúc:*

Các mẫu kiến trúc có mục đích chung là mô tả ….

Bao gồm các mẫu:

* Strategy

1. *Mẫu hành vi*
   1. **Lợi ích khi sử dụng mẫu**

Việc sử dụng các mẫu thiết kế trong quá trình sản xuất phần mềm có rất nhiều lợi ích, tiêu biểu là:

* 1. **Cấu trúc trình bày mẫu**

Các mẫu thiết kế trong báo cáo sẽ được trình bày theo cấu trúc dưới đây:

|  |  |
| --- | --- |
| Mục trình bày | Nội dung trình bày |
| Tên, phân loại, bí danh | Mô tả ngắn gọn về mẫu |
| Mục đích, ý định | Mô tả mẫu này làm được những gì |
| Động lực sử dụng | Nêu một ví dụ về một vấn đề cần giải quyết có thể sử dụng mẫu này. |
| Khả năng ứng dụng | Liệt kê một số tình huống cụ thể trong thiết kế phần mềm có thể áp dụng mẫu này |
| Cấu trúc mẫu | Mô tả mẫu bằng một sơ đồ UML bao gồm các lớp và đối tượng |
| Các thành viên | Trình bày ý nghĩa của các lớp/đối tượng tha gia vào mẫu thiết kế và trcahs nhiệm của chúng |
| Sự cộng tác | Mô tả các thức các thành viên của mẫu tương tác với nhau như thế nào để thực hiện trách nhiệm của chúng |
| Các hệ quả mang lại | Trình bày về ưu điểm khi sử dụng mẫu, các nhược điểm sẽ dẫn tới khi sử dụng mẫu. |
| Các chú ý khi cài đặt | Các chú ý đặc biệt khi cài đặt mẫu |
| Mã nguồn minh họa | Trình bày một ví dụ demo cụ thể về mẫu |
| Ví dụ thực tế | Nêu ra những ví dụ thực tế về các hệ thống (đã pt triển và đang chạy) có sử dụng mẫu này |
| Các mẫu liên quan | Những mẫu nào có liên hệ đến mẫu này, những điểm quan trọng cần phân biệt, mẫu này có thể phối hợp với những mẫu nào. |

1. **Các mẫu Thiết kế GoF** 
   1. **Mẫu Singleton**
   2. **Mẫu Proxy**
   3. **Mẫu Template**
   4. **Mẫu Façade**
   5. **Mẫu Observer**
2. **Tổng kết**
3. **Tài liệu tham khảo**

* Sách : thinking in java 4th Edition – tác giả - năm xuất bản – nhà xuất bản – chương sử dụng
* Tài liệu internet:
  + <http://sce2.umkc.edu/BIT/burrise/pl/design-patterns/>