GOF 设计模式

Will Also

2019

目录

第一章　引言

1.1　什么是设计模式

Christopher Alexander：“每一个模式描述了一个在我们周围不断重复发生的问题，以及该问题的解决方案的核心。这样，你就能一次又一次地使用该方案而不必做重复劳动”。

设计模式的四个基本要素：

1. 模式名称（pattern name），用一两个词来描述模式的问题、解决方案和效果。
2. 问题（problem）描述了应该在何时使用模式。它解释了设计问题和问题存在的前因后果，它可能描述了特定的设计问题，如怎样用对象表示算法等。也可能描述了导致不灵活设计的类或对象结构。
3. 解决方案（solution）描述了设计的组成成分，它们之间的相互关系及各自的职责和协作方式。提供设计问题的抽象描述和怎样用一个具有一般意义的元素组合（类或对象组合）来解决这个问题。
4. 效果（consequence）描述了模式应用的效果及使用模式应权衡的问题。模式效果包括它对系统的灵活性、扩充性或可移植性的影响。

1.2　Smalltalk MVC 中的设计模式

MVC 包括三类对象。模型 Model 是应用对象，视图 View 是它在屏幕上的表示，控制器 Controller 定义用户界面对用户输入的响应方式。不使用 MVC，用户界面设计往往将这些对象混在一起，而 M V C 则将它们分离以提高灵活性和复用性。

MVC 的另一个特征是视图可以嵌套。

MVC 允许你在不改变视图外观的情况下改变视图对用户输入的响应方式。

View-Controller 关系是 Strategy 式的一个例子。一个策略是一个表述算法的对象。当你想静态或动态地替换一个算法，或你有很多不同的算法，或算法中包含你想封装的复杂数据结构，这时策略模式是非常有用的。

还是用了其他模式：用来指定视图缺省控制器的 Factory Method 和用来增加视图滚动的 Decorator，但主要关系还是由 Observer、Composite 和 Strategy 给出。

1.3 描述设计模式

1. 模式和分类：模式名简洁地描述了模式的本质。
2. 意图：设计模式是做什么的？它的基本原理和意图是什么？它解决的是什么样的特定设计问题？
3. 别名：模式的其他名称。
4. 动机：用以说明一个设计问题以及如何用模式中的类、对象来解决该问题的特定情景。
5. 适用性：什么情况下可以使用该设计模式？该模式可用来改进哪些不良设计？你怎样识别这些情况？
6. 结构：采用基于对象建模技术 OMT 的表示法对模式中的类进行图形描述
7. 参与者：指设计模式中的类和/或对象以及它们各自的职责。
8. 协作：模式的参与者怎样协作以实现它们的职责。
9. 效果：模式怎样支持它的目标？使用模式的效果和所需做的权衡取舍？系统结构的哪些方面可以独立改变？
10. 实现：实现模式时需要知道的一些提示、技术要点及应避免的缺陷，以及是否存在某些特定于实现语言的问题。
11. 代码示例：代码片段。
12. 已知应用：实际系统中发现的模式的例子。
13. 相关模式：与这个模式紧密相关的模式有哪些？其间重要的不同之处是什么？

1.4 组织编目

第一是目的准则，即模式是用来完成什么工作的。模式分为：创建型（Creational）、结构型（Structural）和行为型（Behavioral）。

- 创建型模式与对象的创建有关；
- 结构型模式处理类或对象的组合；
- 行为型模式对类或对象怎样交互和怎样分配职责进行描述。

第二是范围准则，指定模式主要是用于类还是用于对象。

		目的		
		创建型	结构型	行为型
范围	类	Factory Method	Adapter	Interpreter Template Method
	对象	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Mementor Observer State Strategy Visitor

表 1.1: 设计模式分类

- 类模式处理类和子类之间的关系，这些关系通过继承建立，是静态的，在编译时刻便确定下来了。
- 对象模式处理对象间的关系，这些关系在运行时刻是可以变化的，更具动态性。从某种意义上来说，几乎所有模式都使用继承机制，所以“类模式”只指那些集中于处理类间关系的模式，而大部分模式都属于对象模式的范畴。