x 设计模式

应用场景举例

摘要

正文

设计模式（Design Pattern）是前辈们对代码开发经验的总结，是解决特定问题的一系列套路。它不是语法规定，而是一套用来提高代码可复用性、可维护性、可读性、稳健性以及安全性的解决方案。

设计模式根据目的可以分为创建型模式、结构型模式和行为型模式 3 种，其中创建型模式用于描述“怎样创建对象”，它的主要特点是“将对象的创建与使用分离”。GoF 中提供了单例、原型、工厂方法、抽象工厂、建造者等 5 种。结构型模式用于描述如何将类或对象按某种布局组成更大的结构，GoF 中提供了代理、适配器、桥接、装饰、外观、享元、组合等 7 种结构型模式。行为型模式用于描述类或对象之间怎样相互协作共同完成单个对象都无法单独完成的任务，以及怎样分配职责。GoF 中提供了模板方法、策略、命令、职责链、状态、观察者、中介者、迭代器、访问者、备忘录、解释器等 11 种行为型模式。

设计模式还可以根据作用范围分为类模式和对象模式，其中类模式用于处理类与子类之间的关系，这些关系通过继承来建立，是静态的，在编译时刻便确定下来了。GoF 中的工厂方法、（类）适配器、模板方法、解释器属于该模式。对象模式用于处理对象之间的关系，这些关系可以通过组合或聚合来实现，在运行时刻是可以变化的，更具动态性。GoF 中除了以上 4 种，其他的都是对象模式。

单例模式

定义：一个类只有一个实例，且该类能自行创建这个实例的一种模式.

优点：由于单例模式只允许创建一个对象，共享该对象可以节省内存，并加快对象访问速度。

例子：网站计数器/ 数据库连接池. 我们需要统计网站的访问人数，用单例模式实现了一个计数器类，该类只创建一个可共享的对象，实现了网站访问人数统计功能.

抽象工厂

定义：抽象工厂模式提供一个接口，可以创建一系列相关或相互依赖的对象.

优点：用户只需要知道具体工厂的名称就可得到所要的产品，无须知道产品的具体创建过程；灵活性增强，对于新产品的创建，只需多写一个相应的工厂类。

例子：可以针对 Oracle、MySQL、DB2 等分别建立抽象工厂，如指定当前工厂为 Oracle 工厂，则创建出来的数据库连接，数据集等一系列的对象都是符合 Oracle 操作要求的，这样便于数据库之间的切换.

责任链模式

定义：为了避免请求发送者与多个请求处理者耦合在一起，将所有请求的处理者通过前一对象记住其下一个对象的引用而连成一条链；当有请求发生时，可将请求沿着这条链传递，

直到有对象处理它为止.

优点：客户只需要将请求发送到责任链上即可，无须关心请求的处理细节和请求的传递过程，所以责任链将请求的发送者和请求的处理者解耦了；可扩展性；灵活性；责任分担；

例子：点线面三种空间矢量数据的处理，定义的坐标系不同，附加操作不同，点的话增加 POI 统计数，线的话增加路段数量，面的话增加用地数量，处理后保存到各自对应的坐标系数据集.

适配器模式

定义: 将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类能一起工作。

例子: 监控摄像头有多种品牌, 每种类型摄像头有既有的驱动组件, 用于控制摄像头开/关机, 镜头旋转, 变焦等功能, 但是这些组件接口不一, 想要统一操作接口, 用适配器模式实现. 定义一个目标接口, 在给每种品牌摄像机分别定义一个适配器类, 适配器类实现目标接口, 并继承既有驱动, 实现客户端对目标接口的透明调用.

策略模式

定义: 该模式定义了一系列算法，并将每个算法封装起来，使它们可以相互替换，且算法的变化不会影响使用算法的客户。

优点: 多重条件语句不易维护，而使用策略模式可以避免使用多重条件语句。

例子: 坐标系的转换, 高德/ 百度/ 标准 wgs84 , 如原始空间数据的坐标系为高德坐标, 需要转换为其它坐标系, 定义一个抽象策略类和两个具体策略类, 具体策略类 A—标准转高德, 具体策略类 B—标准转百度, 还有一个环境类, 持有一个策略类的引用, 最终给客户端调用.