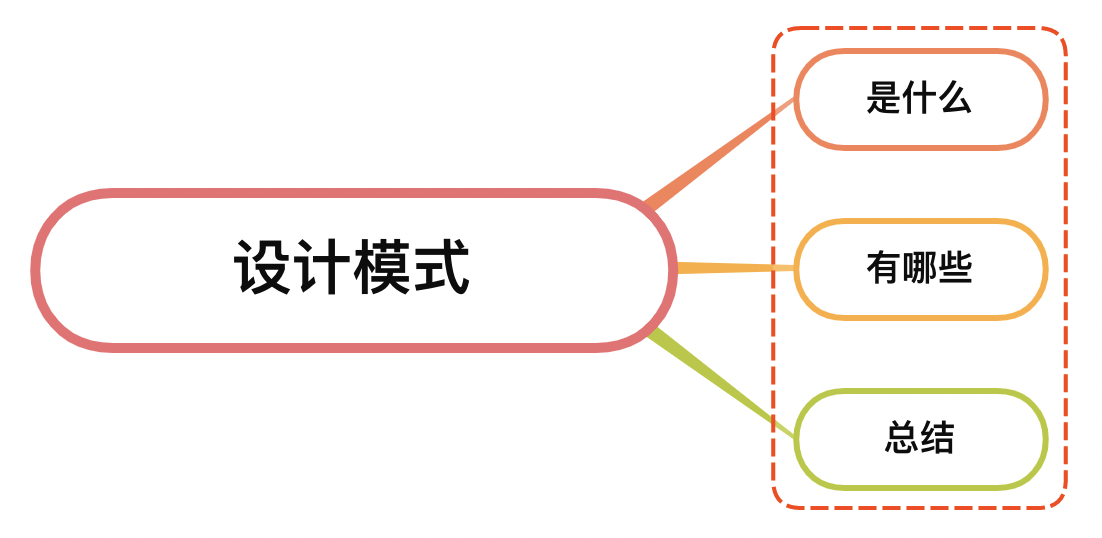
# 面试官：说说对设计模式的理解？常见的设计模式有哪些？



## 一、是什么

在软件工程中，设计模式是对软件设计中普遍存在的各种问题所提出的解决方案

设计模式并不直接用来完成代码的编写，而是描述在各种不同情况下，要怎么解决问题的一种方案

设计模式能使不稳定依赖于相对稳定、具体依赖于相对抽象，避免会引起麻烦的紧耦合，以增强软件设计面对并适应变化的能力

因此，当我们遇到合适的场景时，我们可能会条件反射一样自然而然想到符合这种场景的设计模式

比如，当系统中某个接口的结构已经无法满足我们现在的业务需求，但又不能改动这个接口，因为可能原来的系统很多功能都依赖于这个接口，改动接口会牵扯到太多文件

因此应对这种场景，我们可以很快地想到可以用适配器模式来解决这个问题

## 二、有哪些

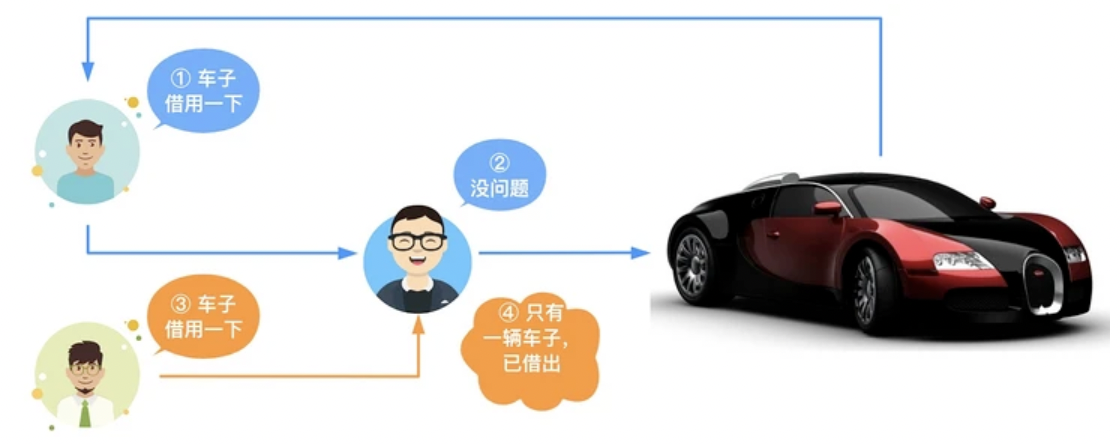
常见的设计模式有：

* 单例模式
* 工厂模式
* 策略模式
* 代理模式
* 中介者模式
* 装饰者模式
* ......

### 单例模式

保证一个类仅有一个实例，并提供一个访问它的全局访问点。实现的方法为先判断实例存在与否，如果存在则直接返回，如果不存在就创建了再返回，这就确保了一个类只有一个实例对象

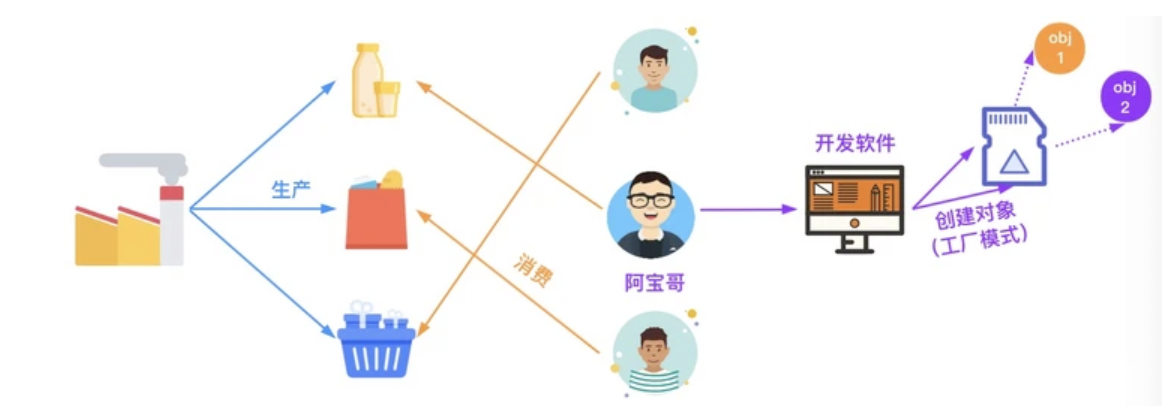
如下图的车，只有一辆，一旦借出去则不能再借给别人：



### 工厂模式

工厂模式通常会分成3个角色：

* 工厂角色-负责实现创建所有实例的内部逻辑.
* 抽象产品角色-是所创建的所有对象的父类，负责描述所有实例所共有的公共接口
* 具体产品角色-是创建目标，所有创建的对象都充当这个角色的某个具体类的实例



### 策略模式

策略模式，就是定义一系列的算法，把他们一个个封装起来，并且使他们可以相互替换

至少分成两部分：

* 策略类（可变），策略类封装了具体的算法，并负责具体的计算过程
* 环境类（不变），接受客户的请求，随后将请求委托给某一个策略类

### 代理模式

代理模式：为对象提供一个代用品或占位符，以便控制对它的访问

例如实现图片懒加载的功能，先通过一张loading图占位，然后通过异步的方式加载图片，等图片加载好了再把完成的图片加载到img标签里面

### 中介者模式

中介者模式的定义：通过一个中介者对象，其他所有的相关对象都通过该中介者对象来通信，而不是相互引用，当其中的一个对象发生改变时，只需要通知中介者对象即可

通过中介者模式可以解除对象与对象之间的紧耦合关系

### 装饰者模式

装饰者模式的定义：在不改变对象自身的基础上，在程序运行期间给对象动态地添加方法

通常运用在原有方法维持不变，在原有方法上再挂载其他方法来满足现有需求

## 三、总结

不断去学习设计模式，会对我们有着极大的帮助，主要如下：

* 从许多优秀的软件系统中总结出的成功的、能够实现可维护性、复用的设计方案，使用这些方案将可以让我们避免做一些重复性的工作
* 设计模式提供了一套通用的设计词汇和一种通用的形式来方便开发人员之间沟通和交流，使得设计方案更加通俗易懂
* 大部分设计模式都兼顾了系统的可重用性和可扩展性，这使得我们可以更好地重用一些已有的设计方案、功能模块甚至一个完整的软件系统，避免我们经常做一些重复的设计、编写一些重复的代码
* 合理使用设计模式并对设计模式的使用情况进行文档化，将有助于别人更快地理解系统
* 学习设计模式将有助于初学者更加深入地理解面向对象思想

## 参考文献

* https://zh.wikipedia.org/wiki/%E8%AE%BE%E8%AE%A1%E6%A8%A1%E5%BC%8F\_(%E8%AE%A1%E7%AE%97%E6%9C%BA)
* https://juejin.cn/post/6844903795017646094
* https://segmentfault.com/a/1190000030850326