# 设计模式的定义

1. 设计模式是一套被反复使用的、多数人知晓的、经过分类编目的、代码设计经验的总结，使用设计模式是为了可重用代码、让代码更容易被他人理解并且提高代码的可靠性。
2. 设计模式是一种用于对软件系统不断重复的设计问题的解决方案进行文档化的技术，也是一种共享专家设计经验的技术。
3. 设计模式是在特定环境下为解决某一通用软件设计问题提供的一套定制的解决方案，该方案描述了对象和类之间的相互作用。

# 设计模式的分类

设计模式根据目的（模式是用来做什么）可分为创建型、结构型和行为型3类：

1. 创建型主要用于创建对象，共5种：工厂方法模式、抽象工厂模式、建造者模式、原型模式、单例模式；
2. 结构型模式主要用于处理类或对象的组合，共7种：适配器模式、桥接模式、组合模式、装饰模式、外观模式、享元模式、代理模式；
3. 行为型模式主要用于描述类或对象怎样交互和怎样分配职责，共 11种：职责链模式、命令模式、解释器模式、迭代器模式、中介者模式、备忘录模式、观察者模式、状态模式、策略模式、模板方法模式、访问模式；

需要注意的是，这23种设计模式并不是孤立存在的，很多模式之间存在联系，例如在访问者模式中操作对象结构中的元素通常需要使用迭代器模式，在解释器模式中定义中介符表达式和非终结符表达式时使用组合模式；此外，还可以通过组合两个或者多个模式来设计同一个系统，在充分发挥每一个模式的优势的同时使它们可以协同工作，完成一些更复杂的设计工作；

# 设计模式的优点

设计模式是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，可复用的设计方案，使用这些方案将避免做一些重复性的工作，而且可以设计出高质量的软件系统。

1. 设计模式融合了众多专家的经验，并以一种标准的形式供广大开发人员所用，它提供了一条通用的设计词汇和一种通用的语言以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。对于使用不同编程语言的开发和设计人员可以通过设计模式来交流系统设计方案，每一个模式都对应一个标准的解决方案，设计模式可以降低开发人员理解系统的复杂度；
2. 设计模式使人们更加简单、方便地复用成功的设计和体系结构，将已证实的技术表述成设计模式也会使新系统开发者更加容易和理解其设计思路。设计模式使得重用成功的设计更加容易，并避免那些导致不可重用的设计方案；
3. 设计模式使得设计方案更加灵活，且易于修改，使得系统具有较好的可维护性，实现了可维护性的复用；在JDK、Spring、Junit等类库和框架的设计中也大量使用了设计模式；
4. 设计模式的使用将提高人家系统的开发效率和软件质量，并且在一定程度上节约了设计成果。设计模式是一些通过多次实践得以证明的行之有效的解决方案，这些解决方案是针对某一个类问题最佳的设计方案，因此可以帮助设计人员构造优秀的软件系统，并可以直接重用这些设计模式，节省了系统设计成本；
5. 设计模式有助于初学者更深入理解面向对象思想，一方面可以帮助初学者更加方便的阅读和学习现有类库与其他系统或框架的源代码，另一方面还可以提高软件的设计水平和代码质量。