## 简介

《Design Patterns: Elements of Reusable Object-Oriented Software》是一本由Erich Gamma、Richard Helm、Ralph Johnson和John Vlissides合著的经典书籍，通常简称为GoF（Gang of Four，四人帮）书籍。这本书描述了23种常见的设计模式，它们被分为三种类型：创建型模式、结构型模式和行为型模式。

这些设计模式提供了一种通用的解决方案，用于解决在软件设计中常见的问题。每个设计模式都描述了一个问题，然后提供了一个可重用的解决方案。这些解决方案已经在实践中被证明是有效的，并且能够提高代码的可维护性、可扩展性和可重用性。

GOF设计模式为我们提供了丰富的工具箱，可以帮助我们更好地设计和组织软件系统。在员工管理系统的开发过程中，结合不同的设计模式可以使系统更加灵活、可扩展和易于维护。然而，需要根据实际情况和设计原则谨慎选择合适的模式，并注意避免过度设计。

## 下面我将根据我们的员工管理系统进行运用解释

创建型模式中，我们使用了单例模式 ，通过私有化构造函数和静态方法来实现，确保只能通过特定的方式获取唯一的实例。这个模式确保系统中只有一个员工管理系统实例。

我们还是用了工厂模式，也就是说我们创建了不同类型的对象，比如员工和管理员。工厂模式可以帮助将对象的创建逻辑封装起来，使系统更加灵活和可扩展。

结构型模式中，我们使用了代理模式，创建一个代理类，控制对真实对象的访问，并在必要时进行权限检查。这样控制对敏感信息的访问权限，例如我们在员工管理系统中分为了管理员和普通员工两种类型，对信息拥有不同的访问权限。

行为型模式中，我们使用了命令模式，实现对员工信息的操作。将每个操作封装成一个命令对象，客户端通过调用命令对象来执行相应的操作。如添加员工、删除员工。

## 除了上面用到的模式以外，我们可以在下面进行改进升级

在创建型模式中，我们可以使用抽象工厂模式。如果员工管理系统需要提供不同国家或地区的特定功能，可以使用抽象工厂模式来创建适应不同文化习惯的员工管理对象。

在结构型模式中，我们可以使用装饰器模式。员工管理系统可能需要动态地给员工添加额外的信息或功能，比如给员工添加奖金信息或者培训记录，这时可以使用装饰器模式。

在行为型模式中，我们可以添加策略模式。员工管理系统可能包含不同的计算工资方式，比如按小时计算工资和固定工资两种策略。通过策略模式，可以使得工资计算算法独立于具体的员工类，方便扩展和修改。