**实验十二 设计模块（一 ）**

**实验目的：**

1. 培养设计原则实践的能力

2. 学习依赖注入（dependency injection）

**实验内容：**

**1. 阅读下面DI资料****（或查阅其它相关资料），学习依赖注入技术。**

[Dependency injection - Wikipedia](https://en.wikipedia.org/wiki/Dependency_injection#:~:text=In%20software%20engineering%2C%20dependency%20injection,object%20is%20called%20a%20service.)

Dependency Injection-A Practical Introduction.pdf

1. 依赖注入的关键概念和好处：

* 解耦：通过将依赖关系的创建与使用分离，使得被依赖对象的变更不会直接影响到依赖它的类，从而降低了模块间的耦合。
* 测试性：在单元测试时，可以轻松地为被测试类提供模拟或存根的依赖，使测试更加集中于被测试类本身的逻辑，而不受外部依赖的影响。
* 扩展性：新的实现可以很容易地插入到系统中，只需修改配置而无需改动大量现有代码。
* 后期绑定(Late Binding)：依赖关系直到运行时才确定，这意味着可以在应用程序运行时动态改变依赖对象，增加了灵活性。
* 遵循SOLID原则：尤其是单一职责原则(S)、开放封闭原则(O)和依赖倒置原则(D)，帮助构建更健壮、可维护的代码。

1. 常见的依赖注入模式包括：

* 构造器注入(Constructor Injection)：通过构造函数传递依赖。
* 属性注入(Property Injection)：通过设置公共属性来注入依赖。
* 方法注入(Method Injection)：通过方法参数来传递依赖。