实验十二 设计模块（一 ）

郑家彤 202100032018

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. 参考教材6.2，结合项目的进程和开发历程，从设计原则的几个方面，组员对负责设计的模块进行评估，思考存在的问题和解决方案。

2. 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" \l ":~:text=In%20software%20engineering%2C%20dependency%20injection,object%20is%20called%20a%20service.)

Dependency Injection-A Practical Introduction.pdf

依赖注入定义

依赖注入允许程序在运行时决定组件需要的依赖，而不是编译时。这意味着我们可以在不修改代码的情况下动态地替换或选择具体实现，使得系统更加灵活。

为什么使用依赖注入

扩展性：新功能添加容易，因为依赖关系外部管理，可插入新实现而无需修改现有代码。

测试性：单元测试更简单，可替换依赖为模拟，测试独立部分代码而不涉及真实依赖。

耦合：减少依赖，代码间关系，易于维护和理解。

晚绑定：运行时选择依赖，无需编译时决定，提高灵活性。

常见模式

构造器注入：最推荐方式，通过构造函数传入依赖。

属性注入：设置属性，通过setter方法后设置依赖。

方法注入：传参，方法参数传入依赖。

服务定位器：容器查找：从容器获取。

项目跟踪，建立能反映项目及小组每个人工作的进度、里程碑、工作量的跟踪图或表，将其保存到每个小组选定的协作开发平台上，每周更新。