实验十三 设计模块（二）

实验目的：

面向对象设计原则

实验内容：

论述利斯科夫替换原则（里氏代换原则）、单一职责原则、开闭原则、德（迪）米特法则、依赖倒转原则、合成复用原则，结合自己的实践项目举例说明如何应用 （保存到每个小组选定的协作开发平台上，以组为单位）。

1.利斯科夫替换原则（里氏代换原则）  
里氏代换原则指出，如果S是T的子类型，那么程序中的对象T都可用S来替换，而不会改变程序的任何设计功能。这要求子类必须能够替换其父类，并且保持原有程序的正确性。这要求子类必须实现与基类相同的方法签名，并且对于相同的输入，子类和基类必须产生相同的输出。

在人员管理系统中的应用：  
在人员管理系统中，我们有一个基类Employee，表示员工。

然后，我们也会涉及一些子类，如Manager。它们都继承自Employee。

按照里氏代换原则，我们可以在任何需要Employee对象的地方使用Manager或Engineer对象，因为这两个子类应该能够替换其父类Employee，而不会破坏程序的正确性。

2.单一职责原则

单一职责原则表明一个类或模块应该只有一个引起它变化的原因。

也就是说，一个类或模块应该只负责一项功能或关注点。

在人员管理系统中的应用：  
在人员管理系统中，我们可以应用单一职责原则来设计各个模块。

如果我们设计一个EmployeeService类，它只负责处理与员工相关的业务逻辑，如添加、删除、更新员工信息等。

在我们设计的另一个EmployeeReportService类则只负责生成员工报告，不负责其他业务逻辑。

3.开闭原则  
开闭原则意味着软件实体（如类、模块、函数等）应该对扩展开放，对修改关闭。

也就是说，当软件需要变化时，应该通过扩展来实现，而不是修改已有的代码。

在人员管理系统中的应用：  
在人员管理系统中，我们可以使用开闭原则来设计可扩展的系统。例如，当我们需要添加新的员工类型（如实习生）时，我们可以创建一个新的子类Intern，它继承自Employee类，并添加特定的属性和方法。这样，我们就不需要修改已有的Employee类或其他相关类，只需要扩展系统即可。

4.迪米特法则  
迪米特法则（Law of Demeter, LoD）也称为最少知识原则，它要求一个对象应该对其他对象保持最少的了解。也就是说，一个对象应该只与它的直接朋友通信，而不应该与陌生人通信。

在人员管理系统中的应用：  
在人员管理系统中，我们可以应用迪米特法则来降低类之间的耦合度。

例如，Employee类只需要知道与其直接相关的类的信息，而不需要知道其他不相关的类的信息。这样可以降低系统的复杂性，提高可维护性。

5.依赖倒转原则  
依赖倒转原则（DIP）要求高层模块不应该依赖于低层模块，它们都应该依赖于抽象。抽象不应该依赖于细节，细节应该依赖于抽象。

在人员管理系统中的应用：  
在人员管理系统中，我们可以使用接口或抽象类来定义员工的行为，然后让具体的员工类实现这些接口或抽象类。

这样，高层模块（如业务逻辑层）就可以依赖于这些抽象接口或类，而不是具体的员工类。当需要添加新的员工类型时，我们只需要实现新的员工类并使其符合抽象接口或类的要求即可，而不需要修改高层模块的代码。

6.合成复用原则  
合成复用原则（CARP）要求尽量使用对象组合/聚合来实现复用，而不是使用继承。

在人员管理系统中的应用：  
在人员管理系统中，我们可以使用对象组合/聚合来实现复用。

比如在系统中，一个Team类可以包含多个Employee对象，并通过组合这些对象来实现团队的功能。当需要添加新的团队成员时，我们只需要将新的Employee对象添加到Team对象中即可，而不需要通过继承来扩展Team类的功能。这样可以提高系统的灵活性和可扩展性。