实验十四 设计模块（三 ）

实验目的：

学习设计模式，能在项目设计中运用设计模式进行面向对象设计

实验内容：

1. 阅读下面设计模式资料（或查阅其它相关资料），结合项目的进程和开发历程，分析项目采用了那些设计模式

Design Patterns-Elements of Reusable Object-Oriented Software.pdf

The GoF Design Patterns Reference.pdf

[Design Patterns - Wikipedia](https://en.wikipedia.org/wiki/Design_Patterns)

阅读相关资料，设计模式大致可以分为三大类：创建型模式、结构型模式和行为型模式。

* 创建型模式：关注对象的创建机制，使得系统可以独立于如何创建、组合和表示对象。

工厂模式（Factory Pattern）：提供了一个创建对象的接口，允许子类决定实例化哪一个类。工厂方法让类的实例化延迟到子类。

抽象工厂模式（Abstract Factory Pattern）：提供一个接口，用于创建一系列相关或相互依赖的对象，而无需指定它们的具体类。

建造者模式（Builder Pattern）：将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

原型模式（Prototype Pattern）：通过复制一个现有的对象实例来创建新的对象实例，而不是通过实例化对象的类。

单例模式（Singleton Pattern）：确保一个类仅有一个实例，并提供一个全局访问点。

* 结构型模式： 关注如何组合类和对象以形成更大的结构。

适配器模式（Adapter Pattern）：将一个类的接口转换成客户希望的另一个接口。

桥接模式（Bridge Pattern）：将抽象部分与它的实现部分分离，使它们都可以独立变化。

组合模式（Composite Pattern）：将对象组合成树形结构以表示“部分-整体”的层次结构。

装饰器模式（Decorator Pattern）：动态地给一个对象添加一些额外的职责。

外观模式（Facade Pattern）：为子系统中的一组接口提供一个一致的界面。

享元模式（Flyweight Pattern）：运用共享技术有效地支持大量细粒度的对象。

代理模式（Proxy Pattern）：为其他对象提供一个代理以控制对这个对象的访问。

* 行为型模式：关注对象之间的职责分配。

策略模式（Strategy Pattern）：定义一系列算法，把它们一个个封装起来，并且使它们可以相互替换。

模板方法模式（Template Method Pattern）：定义一个操作中的算法骨架，而将一些步骤延迟到子类中。

观察者模式（Observer Pattern）：定义了对象之间的一种一对多依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都会得到通知并自动更新。

命令模式（Command Pattern）：将请求封装为对象，从而让你使用不同的请求、队列或者日志来参数化其他对象。

迭代器模式（Iterator Pattern）：提供一种方法顺序访问一个聚合对象中各个元素，而又不暴露该对象的内部表示。

访问者模式（Visitor Pattern）：表示一个作用于某对象结构中的各元素的操作。

责任链模式（Chain of Responsibility Pattern）：使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

备忘录模式（Memento Pattern）：在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，这样以后就可将该对象恢复到原先保存的状态。

解释器模式（Interpreter Pattern）：给定一个语言，定义它的文法的一种表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。

状态模式（State Pattern）：允许一个对象在其内部状态改变时改变它的行为。

策略模式（Strategy Pattern）：定义了一系列算法，并将每一个算法封装起来，使它们可以相互替换。

中介者模式（Mediator Pattern）：用一个中介对象来封装一系列的对象交互，使原有对象之间的耦合松散。

2. 给出4种设计模式的例子（语言不限，以组为单位），并总结其特点 （保存到每个小组选定的协作开发平台上）

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。

* 创建型设计模式

创建型模式是处理对象创建的设计模式，试图根据实际情况使用合适的方式创建对象。基本的对象创建方式可能会导致设计上的问题，或增加设计的复杂度。创建型模式通过以某种方式控制对象的创建来解决问题。

创建型模式由两个主导思想构成。一是将系统使用的具体类封装起来，二是隐藏这些具体类的实例创建和结合的方式。

创建型模式又分为对象创建型模式和类创建型模式。对象创建型模式处理对象的创建，类创建型模式处理类的创建。详细地说，对象创建型模式把对象创建的一部分推迟到另一个对象中，而类创建型模式将它对象的创建推迟到子类中。

* 结构型设计模式

结构型模式讨论的是类和对象的结构，它采用继承机制来组合接口或实现（类结构型模式），或者通过组合一些对象，从而实现新的功能（对象结构型模式）

* 行为型设计模式

行为型模式用于描述程序在运行时复杂的流程控制，即描述对象之间怎样相互协作共同完成单个对象都无法单独完成的任务，是一种微观关系的体现。行为型模式包括：第一组(观察者模式、模板方法、命令模式、状态模式、职责链模式)第二组(解释器模式、中介者模式、访问者模式、策略模式、备忘录模式、迭代器模式)

* 单例模式

单例模式确保一个类只有一个实例，并提供一个全局访问点。具有以下特点：全局唯一性：系统中只存在一个实例。

懒加载：实例在第一次被请求时创建。避免资源浪费：适用于资源消耗较大的场景，如数据库连接。