

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《软件系统功能设计实战训练》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

## 本周设计作业

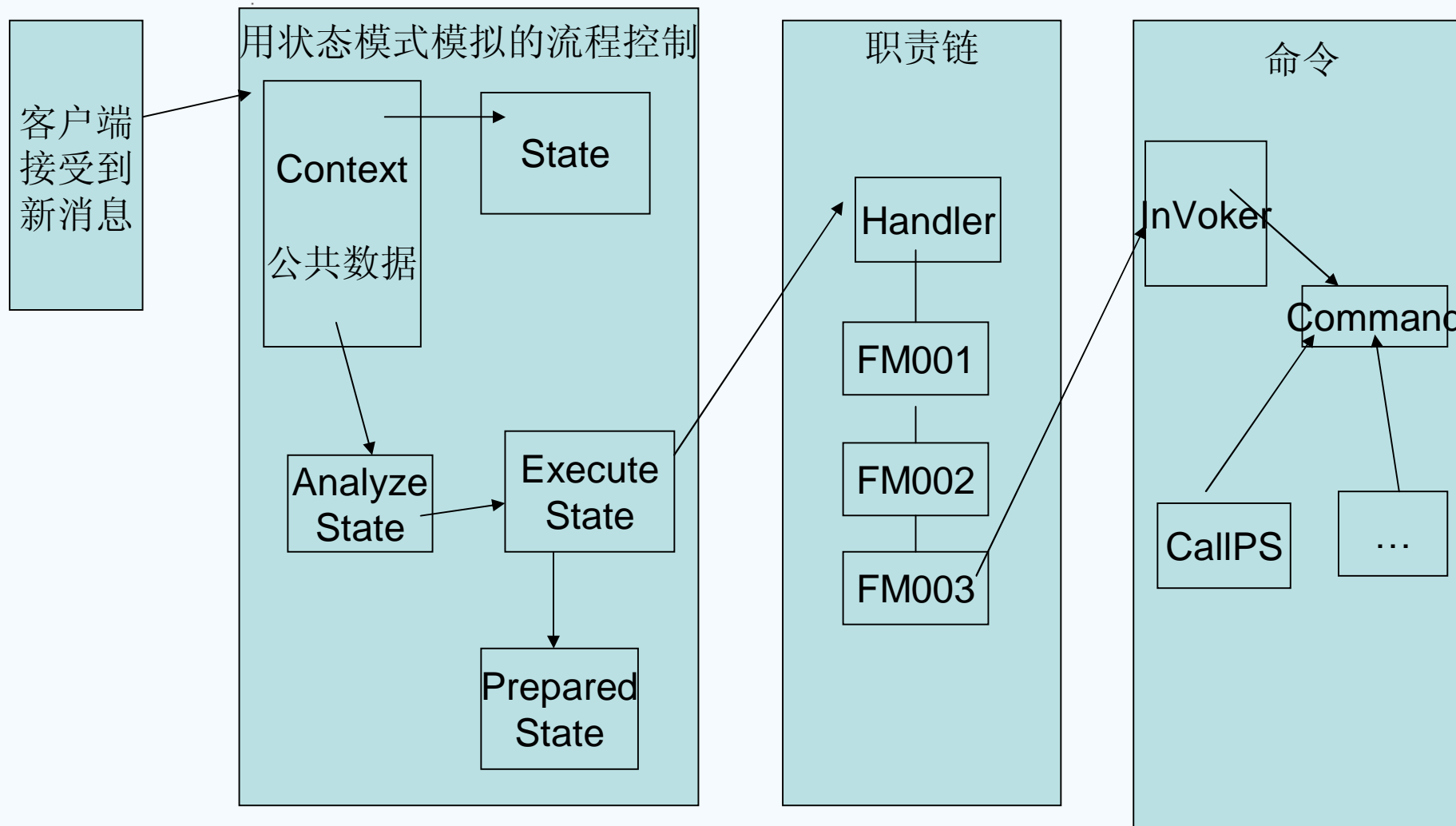
**n** 本周设计作业的项目背景：

热轧厂与加热炉的交互控制系统——来自某大型钢铁企业生产型ERP

**n** 学习目标：

- (1) 在实战中练习设计的理念和方法
- (2) 学习如何设计消息交互系统
- (3) 学习如何处理交互报文
- (4) 综合应用生成器模式、原型模式、备忘录模式、状态模式、命令模式、职责链模式、代理模式等，包含但不限于这些模式

该设计方式可应用于多种有类似功能的系统，比如：任何多模块交互协作完成业务流程的系统（稍微复杂点的系统，几乎都有这样的功能）……



做最好的在线学习社区

网 址: <http://sishuok.com>  
咨询QQ: 2371651507

## 本周设计作业

### n 基本业务

#### 1: 基本的电文接口

加热炉L2电文	描述	MES电文	描述
FM001	加热炉对MES通讯回复电文	MF001	MES对加热炉通讯回复电文
FM002	心跳电文	MF002	心跳电文
FM003	加热炉对MES申请板坯号	MF003	MES返回申请的连铸板坯实绩（PDI）
FM004	加热炉对MES请求轧制计划	MF004	计划板坯生产顺序电文
FM005	加热炉对MES板坯入炉生产实绩	MF005	生产计划调整电文
FM006	加热炉对MES返回拒绝的板坯		
FM007	加热炉对MES板坯号出炉生产实绩		

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费视频独家大放送

## 本周设计作业

### n 基本功能

- 1: 实现消息处理的流程，要求能灵活的装配和扩展，流程并不固定，引入状态模式
- 2: 实现消息的处理部分，不同的消息，需要的处理功能不一样，引入职责链模式
- 3: 实现消息处理部分和真正业务实现部分的解耦，引入命令模式

### n 作业要求

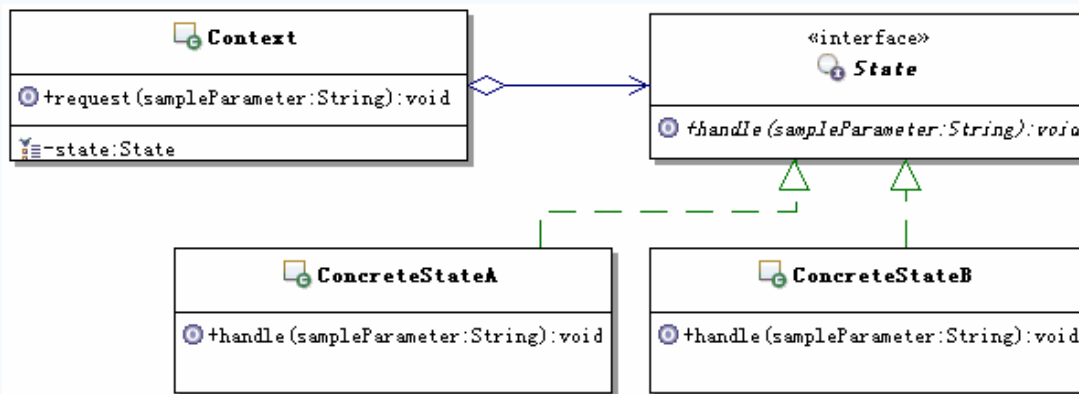
- 1: 在老师给定的概要代码基础上，实现上述基本要求的功能设计
- 2: 重点在接口和功能实现的设计上，无须关注具体业务实现
- 3: 对于每个api，可以适当写点样例代码，能够调用运行更佳
- 4: 考虑合理的结构，职责的划分，以及设计模式的合理使用

## 初识状态模式

### n 定义

允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它的类。

### n 结构和说明



Context：环境，也称上下文，通常用来定义客户感兴趣的接口，同时维护一个来具体处理当前状态的实例对象。

State：状态接口，用来封装与上下文的一个特定状态所对应的行为。

ConcreteState：具体实现状态处理的类，每个类实现一个状态的具体处理。



## 状态模式的知识要点

### n 状态模式的知识要点

- 1: 状态模式的功能就是分离状态的行为，通过维护状态的变化，来调用不同的状态对应的不同的功能
- 2: 状态和行为是相关联的，它们的关系可以描述为：**状态决定行为**。
- 3: 状态模式的状态对应的行为具有平行性，平行性指的是各个状态的行为相互是独立的、没有关联的，相互之间是不可替换的。而策略模式的各个策略之间是平等性，平等性强调的是可替换性，大家是同一行为的不同描述或实现，因此在同一个行为发生的时候，可以根据条件来挑选任意一个实现来进行相应的处理
- 4: 状态模式中，客户端通常不负责运行期间状态的维护，也不负责决定到底后续使用哪一个具体的状态处理对象
- 5: 在状态模式中，通常有两个地方可以进行状态的维护和转换控制。一个是在上下文中，另外一个地方就是在状态的处理类里面
- 6: 由于状态模式对状态的维护是一个内部行为，因此增加新的状态的时候，就需要修改内部的状态维护的代码，算是一种不完美的OCP
- 7: 由于状态模式要维护状态，也就是数据，因此要考虑何时创建和销毁状态对象
- 8: 可以使用数据库来记录和维护状态

## 思考状态模式

### n 状态模式的本质

状态模式的本质是：根据状态来分离和选择行为

### n 何时选用状态模式

- 1: 如果一个对象的行为取决于它的状态，而且它必须在运行时刻根据状态来改变它的行为。可以使用状态模式，来把状态和行为分离开，虽然分离开了，但状态和行为是有对应关系的，可以在运行期间，通过改变状态，就能够调用到该状态对应的状态处理对象上去，从而改变对象的行为
- 2: 如果一个操作中含有庞大的多分支语句，而且这些分支依赖于该对象的状态。可以使用状态模式，把各个分支的处理分散包装到单独的对象处理类里面，这样，这些分支对应的对象就可以不依赖于其它对象而独立变化了

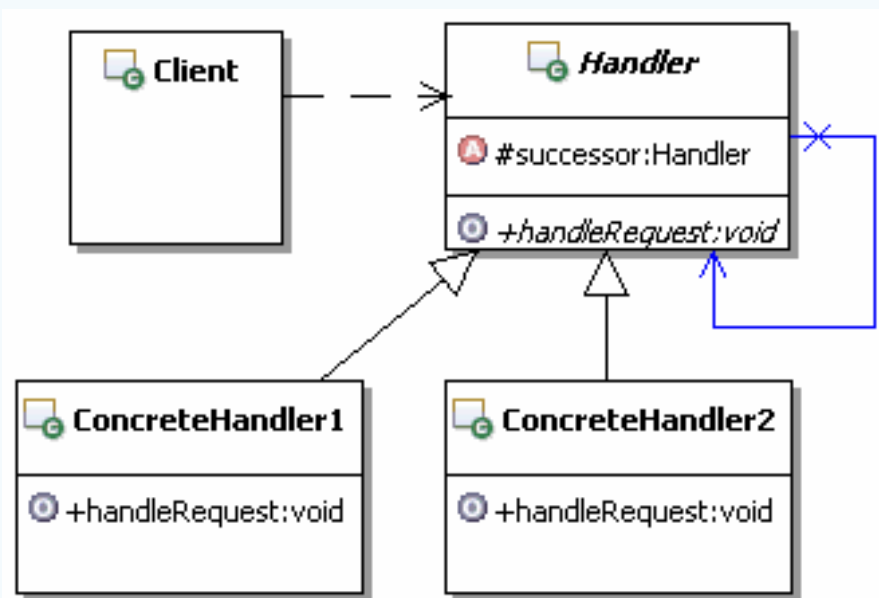


## 初识职责链模式

### n 定义

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。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。

### n 结构和说明



## 职责链模式的知识要点

### n 职责链模式的知识要点

- 1: 职责链模式实现了请求者和具体的接收者解耦，这样就可以动态的切换和组合接收者了
- 2: 客户端发出请求的时候，客户端并不知道谁会真正处理他的请求，客户端只知道他提交请求的第一个对象，然后这个请求会在整条链上传递，直到有人处理了这个请求为止
- 3: 使用职责链模式，一个要注意的点就是谁来构建链，常见分为内部链和外部链，内部链又分在Handler里面构建链，还有一种是在各个职责里面指定后续的职责
- 4: 在职责链模式中，请求不一定会被处理，因为可能没有合适的处理者
- 5: 在实际开发中，经常会把标准的职责链变化成一个请求在职责链中传递，每个职责对象负责处理请求的某一方面的功能，处理完成后，不是停止，而是继续向下传递请求，当请求通过很多职责对象处理过后，功能也就处理完了，把这样的职责链称为功能链
- 6: 动态组合才是职责链模式的精华所在，因为要实现请求对象和处理对象的解耦，请求对象不知道谁才是真正的处理对象，因此要动态的把可能的处理对象组合起来，由于组合的方式是动态的，这就意味着可以很方便的修改和添加新的处理对象，从而让系统更加灵活和具有更好的扩展性

## 思考职责链模式

### n 职责链模式的本质

职责链模式的本质是：分离职责，动态组合

### n 何时选用职责链模式

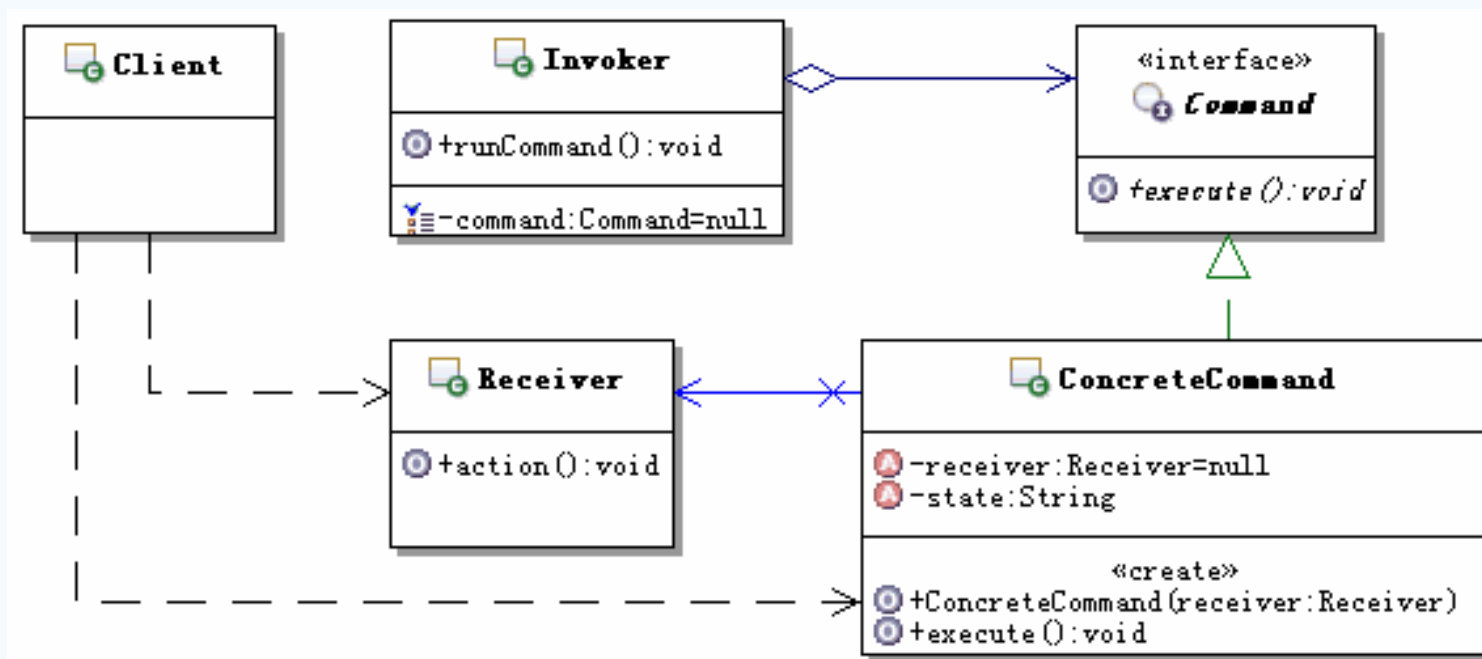
- 1: 如果有多个对象可以处理同一个请求，但是具体由哪个对象来处理该请求，是运行时刻动态确定的。这种情况可以使用职责链模式，把处理请求的对象实现成为职责对象，然后把它们构成一个职责链，当请求在这个链中传递的时候，具体由哪个职责对象来处理，会在运行时动态判断
- 2: 如果你想在不明确指定接收者的情况下，向多个对象中的一个提交一个请求的话，可以使用职责链模式，职责链模式实现了请求者和接收者之间的解耦，请求者不需要知道究竟是哪一个接收者对象来处理了请求。
- 3: 如果想要动态指定处理一个请求的对象集合，可以使用职责链模式，职责链模式能动态的构建职责链，也就是动态的来决定到底哪些职责对象来参与到处理请求中来，相当于是动态指定了处理一个请求的职责对象集合

## 初识命令模式

### n 定义

将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。

### n 结构和说明



## 命令模式的知识要点

### n 命令模式的知识要点

- 1: 命令模式的关键之处就是把请求封装成为对象，也就是命令对象，并定义了统一的执行操作的接口，这个命令对象可以被存储、转发、记录、处理、撤销等，整个命令模式都是围绕这个对象在进行
- 2: 在命令模式中经常会有一个命令的组装者，用它来维护命令的“虚”实现和真实实现之间的关系
- 3: 命令模式的接收者可以是任意的类，只要完成命令要求的功能即可
- 4: 可以有智能的命令，也就是不需要接受者，命令的实现类就真正实现了命令要求的功能
- 5: 命令模式实现了命令的发起方和真正实现对象之间的解耦
- 6: 命令的参数化配置的意思是：可以用不同的命令对象，去参数化配置客户的请求
- 7: 命令模式实现可撤销操作的常见方式：一种是补偿式；另一种是存储恢复式
- 8: 多个命令还可以组合在一起，形成宏命令，宏命令从本质上来说仍然是命令，运行宏命令，其实就是循环运行一个宏命令内的每一个命令
- 9: 命令模式可以退化成为类似于Java回调的实现形式



## 思考命令模式

### n 命令模式的本质

命令模式的本质是：**封装请求**

### n 何时选用命令模式

- 1: 如果需要抽象出需要执行的动作，并参数化这些对象，可以选用命令模式，把这些需要执行的动作抽象成为命令，然后实现命令的参数化配置
- 2: 如果需要在不同的时刻指定、排列和执行请求，可以选用命令模式，把这些请求封装成为命令对象，然后实现把请求队列化
- 3: 如果需要在支持取消操作，可以选用命令模式，通过管理命令对象，能很容易的实现命令的恢复和重做的功能
- 4: 如果需要在支持当系统崩溃时，能把对系统的操作功能重新执行一遍，可以选用命令模式，把这些操作功能的请求封装成命令对象，然后实现日志命令，就可以在系统恢复回来后，通过日志获取命令列表，从而重新执行一遍功能
- 5: 在需要事务的系统中，可以选用命令模式，命令模式提供了对事务进行建模的方法，命令模式有一个别名就是Transaction。