

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）……

## 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

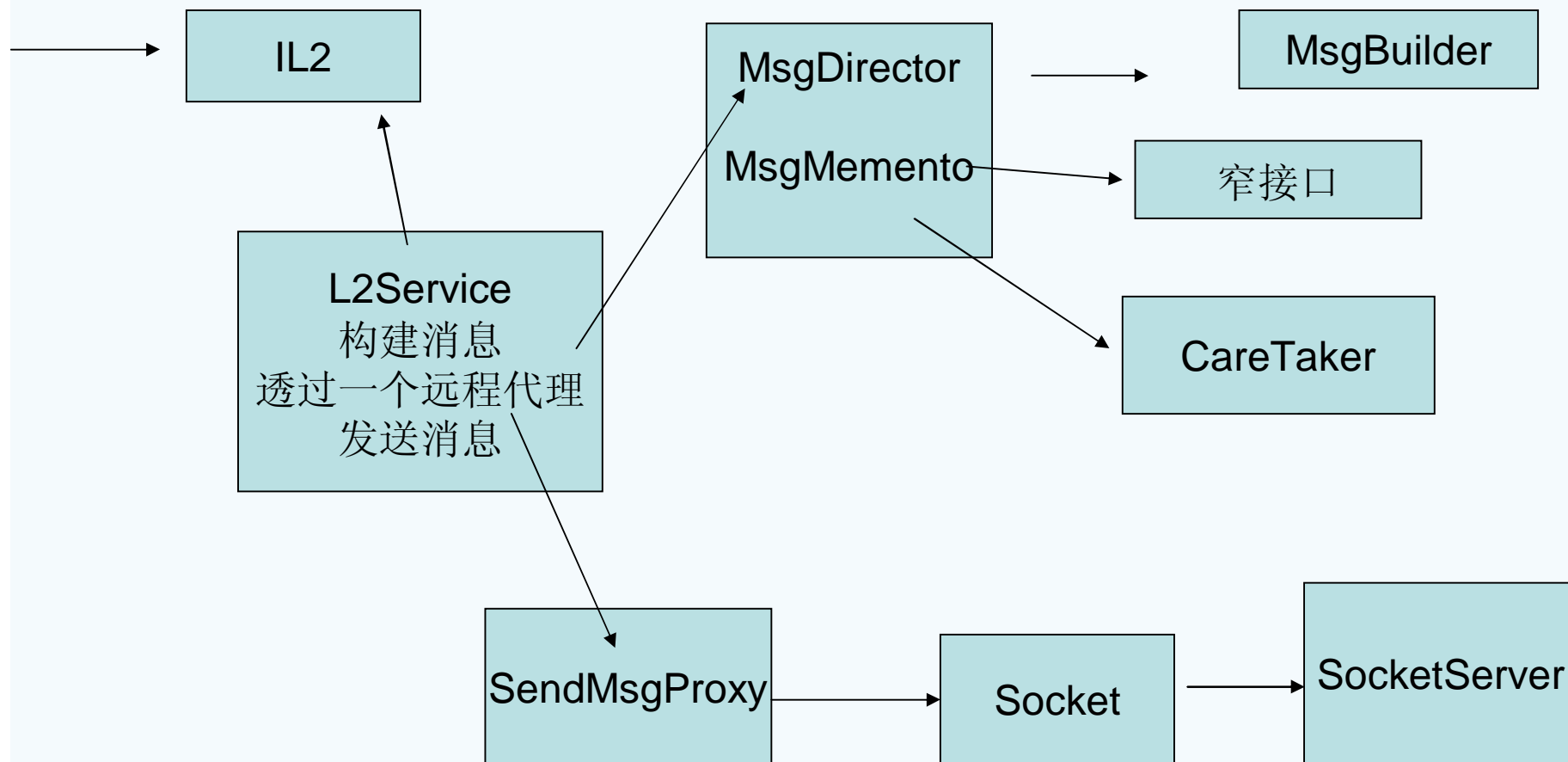
## 本周设计作业

### n 基本功能

- 1: 搭建好消息交互传递部分，引入代理模式
- 2: 实现消息的灵活构建，引入生成器、备忘录和原型模式

### 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

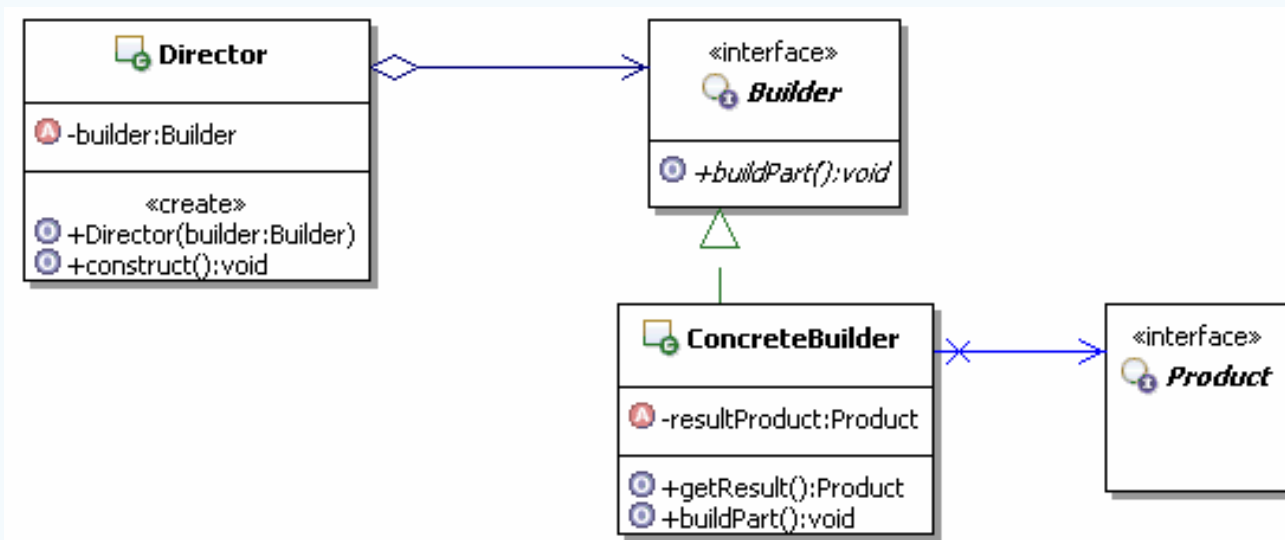


## 初识生成器模式

### n 定义

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。

### n 结构和说明





## 生成器模式的知识要点

### n 生成器模式的知识要点

- 1: 生成器模式重在解决一步一步构造复杂对象的问题，更为重要的是，这个构建的过程是统一的，固定不变的，变化的部分放到生成器部分了。
- 2: 生成器模式的重心在于分离构建算法和具体的构造实现，从而使得构建算法可以重用，具体的构造实现可以很方便的扩展和切换，从而可以灵活的组合来构造出不同的产品对象。
- 3: 生成器模式一般存在两个部分，一个部分是部件构造和产品装配，另一个部分是整体构建的算法。
- 4: 生成器模式有一种退化的情况，就是让客户端和Director融合起来，让客户端直接去操作Builder，就好像是指导者自己想要给自己构建产品一样。
- 5: 生成器模式里面，指导者承担的是整体构建算法部分，是相对不变的部分。
- 6: 指导者分离出去的变化部分，就到了生成器那边，每个生成器对象都可以有两部分功能，一个是创建部件对象，一个是组装部件。
- 7: 在标准的生成器模式里面，在Builder实现里面会提供一个返回装配好的产品的方法，在Builder接口上是没的。
- 8: 在标准的生成器模式里面，一般是不需要对产品定义抽象接口。

## 思考生成器模式

### n 生成器模式的本质

生成器模式的本质是：分离整体构建算法和部件构造

### n 何时选用生成器模式

- 1: 如果创建对象的算法，应该独立于该对象的组成部分以及它们的装配方式时
- 2: 如果同一个构建过程有着不同的表示时

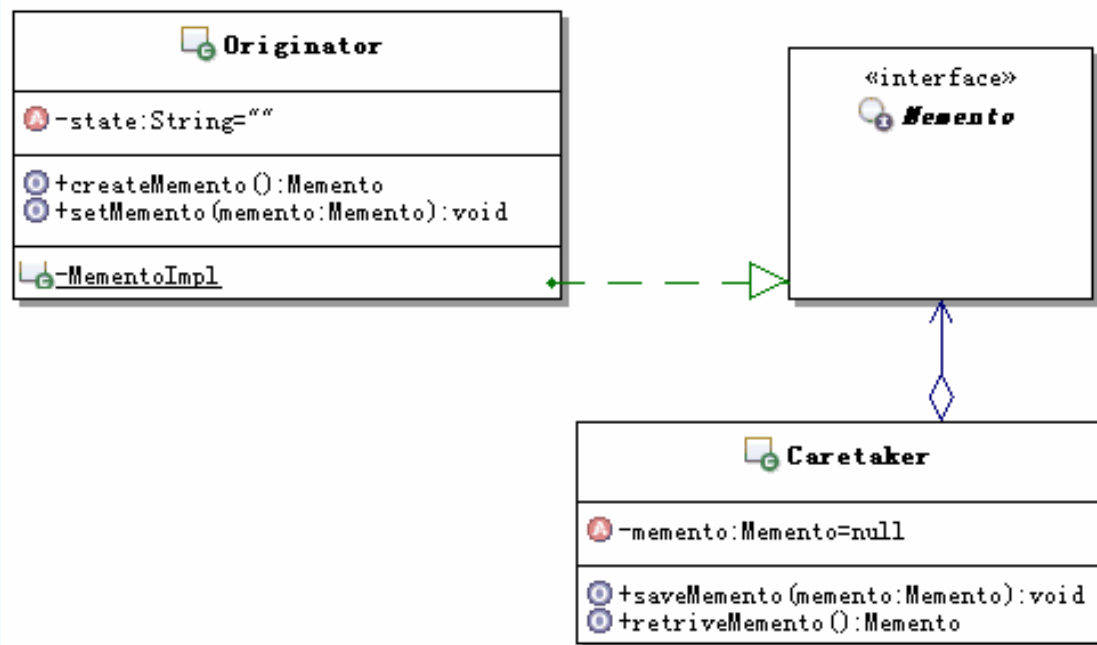


## 初识备忘录模式

### n 定义

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。

### n 结构和说明



## 备忘录模式的知识要点

### n 备忘录模式的知识要点

- 1: 备忘录模式的功能，首先是在不破坏封装性的前提下，捕获一个对象的内部状态。但是要记住，备忘的目的是为了在后面需要的时候进行恢复，把对象的状态恢复到备忘录所保存的状态，这才是备忘录真正的目的。
- 2: 备忘录对象通常都会实现窄接口，一般不考虑宽接口，而且通常会实现成为原发器对象的一个私有的静态内部类
- 3: 管理者对象可要可不要，管理者对象只是存放备忘录对象，并不能操作备忘录对象
- 4: 使用备忘录模式，潜在的代价就是比较耗费内存
- 5: 备忘录模式通常会结合原型模式来使用
- 6: 备忘录的存储方式可以多样，除了内存，还可以离线存储

## 思考备忘录模式

### n 备忘录模式的本质

备忘录模式的本质是：保存和恢复内部状态

### n 何时选用备忘录模式

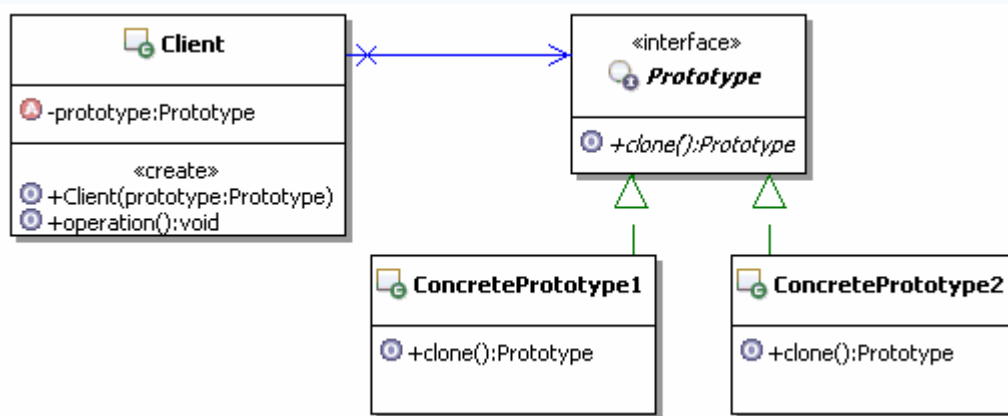
- 1: 如果必须保存一个对象在某一个时刻的全部或者部分状态，这样在以后需要的时候，可以把该对象恢复到先前的状态。可以使用备忘录模式，使用备忘录对象来封装和保存需要保存的内部状态，然后把备忘录对象保存到管理者对象里面，在需要的时候，再从管理者对象里面获取备忘录对象，来恢复对象的状态
- 2: 如果需要保存一个对象的内部状态，但是如果用接口来让其它对象直接得到这些需要保存的状态，将会暴露对象的实现细节并破坏对象的封装性。可以使用备忘录模式，把备忘录对象实现成为原发器对象的内部类，而且还是私有的，从而保证只有原发器对象才能访问该备忘录对象。这样既保存了需要保存的状态，又不会暴露原发器对象的内部实现细节。

## 初识原型模式

### n 定义

用原型实例指定创建对象的种类，并通过拷贝这些原型创建新的对象。

### n 结构和说明



**Prototype:** 声明一个克隆自身的接口，用来约束想要克隆自己的类，要求它们都要实现这里定义的克隆方法。

**ConcretePrototype:** 实现Prototype接口的类，这些类真正实现克隆自身的功能

**Client:** 使用原型的客户端，首先要获取到原型实例对象，然后通过原型实例克隆自身来创建新的对象实例。

## 原型模式的知识要点

### n 原型模式的知识要点

- 1: 原型模式的主要功能就是：通过克隆来创建新的对象实例。一般来讲，新创建出来的实例的数据是和原型实例一样的。
- 2: 原型模式的克隆方法和new操作最明显的不同就在于：new一个对象实例，一般属性是没有值的，或者是只有默认值；如果是克隆得到的一个实例，通常属性是有值的，属性的值就是原型对象实例在克隆的时候，原型对象实例的属性的值。
- 3: Java中已经有了clone方法的基本实现，可以直接使用，但是只是实现了浅度克隆，如果是深度克隆，还是需要额外代码实现
- 4: 要考虑浅度克隆和深度克隆的问题，要想深度克隆成功，必须要整个克隆所涉及的对象都要正确实现克隆方法，如果其中有一个没有正确实现克隆，那么就会导致克隆失败。
- 5: 原型模式的重心在创建新的对象实例，克隆是实现的手段。



## 思考原型模式

### n 原型模式的本质

原型模式的本质是：克隆生成对象

### n 何时选用原型模式

- 1: 如果一个系统想要独立于它想要使用的对象时，可以使用原型模式，让系统只面向接口编程，在系统需要新的对象的时候，可以通过克隆原型来得到
- 2: 如果需要实例化的类是在运行时刻动态指定时，可以使用原型模式，通过克隆原型来得到需要的实例