

## 03、常见面试题分析第三谈

---

### 01、枚举关于单例的问题？

#### 关于枚举

- 定义是enum来修改类，你明确一个道理，枚举是一个类.
- 它的构造函数是私有的private，说明外部不能直接去实例化，只能通过自己定义签名进行实例化。
- enum类不可以继承类，不继承抽象类，但是可以实现接口。因为enum是一个特殊类（final的类）

```
1 asses\com\kuangstudy\demo03> javap
  .\SingleEnum.class
2 Compiled from "SingleEnum.java"
3
4
5 public final class
  com.kuangstudy.demo03.SingleEnum extends
  java.lang.Enum<com.kuangstudy.demo03.SingleE
  num> {
6     public static final
  com.kuangstudy.demo03.SingleEnum MALE;
7     public static final
  com.kuangstudy.demo03.SingleEnum WOMAN;
8     public static
  com.kuangstudy.demo03.SingleEnum[] values();
9     public static
  com.kuangstudy.demo03.SingleEnum
  valueOf(java.lang.String);
10    static {};
```

## 枚举关于单例模式的问题？

互联网开发中，很多面对设计模式的问题，枚举可以单例设计模式，问：

## 枚举真的是单例的吗？

答案：枚举本身不是单例的。

```
1 package com.kuangstudy.demo03;
2
3 public enum SingleEnum {
4     MALE("男"), WOMAN("女");
5     private String name;
6     SingleEnum(String name){
7         this.name = name;
8     }
9 }
```

单例对象是指是什么？

- 在内存空间中只有一份内存地址。
- 恶汉，懒汉，双重`check`机制，枚举机制等等
- 首先第一件事情：
  - 构造函数私有化
  - 通过成员方法暴露当前对象初始化的工作。

#### 04、怎么证明枚举不是单例的呢？

不是，每个枚举引用都是独立的对象内存空间。

#### 05、问题枚举在什么情况下是单例的呢？

但是枚举在一种情况是下单例，就是枚举中定义单个对象的情况下就是单例的。

## 06、spring的bean是单例的吗？

spring的bean是通过所谓恶汉，懒汉，枚举，抽象类或者双从检查机制实现的单例设计模式吗？

记住飞哥一句话：

单例设计模式：它的思想就是告诉你一个对象在内存空间一个只有一个内存地址。

实现的方式1：所谓的恶汉，懒汉，枚举，抽象类、双从检查机制等能够实现单例对象的创建以外？

springbean的实现单例对象的机制：注册表（map）

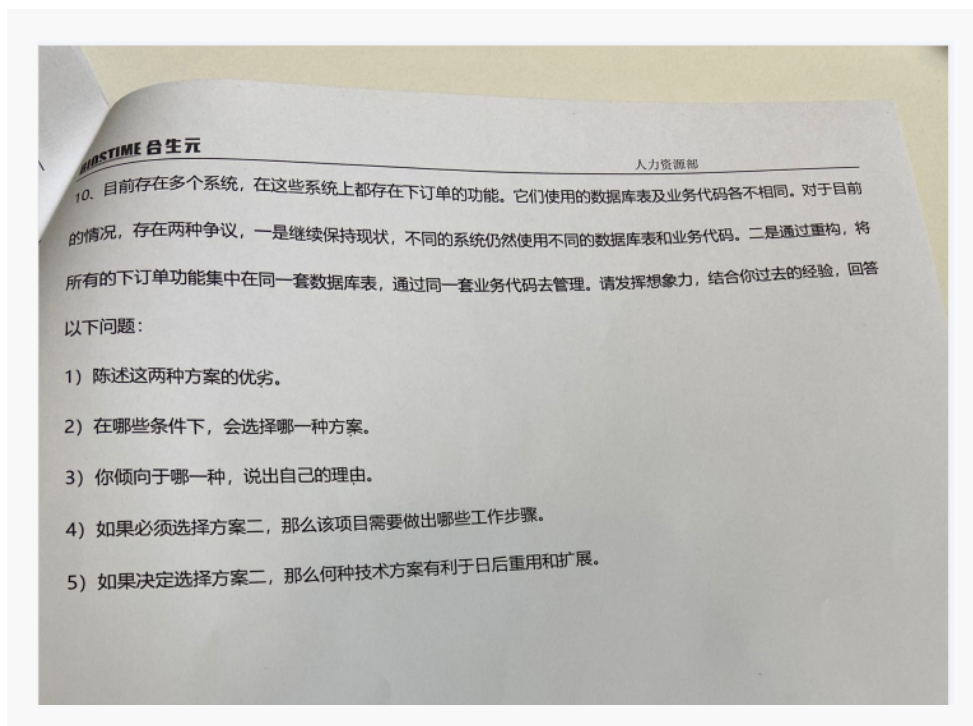
答案：不是，它只是一个单例的设计思想，它的实现就是注册列表的机制，其实就是一个Map

```
1 public class ApplicationClass{
2
3     // 成员变量
4     private static Map<BeanDefintion> map =
5     new ConcurrentHashMap<>();
6
7     map.put("com.kuangstudy.service.UserServiceI
8     mp1",new UserServiceImpl());
```

## 02、关于final的问题



## 03、关于合生元的面试题的架构分析的问题



### 1、陈述两种方案的优劣

- 方案一缺点：订单，支付其实属于通用逻辑，业务冗余，需要维护多份，不利用后续扩展。
- 方案一缺点：系统都独立的，需要不同人员维护，人员成本增加，服务器成本也会增加
- 方案一优点：系统都独立的，并不会因为别的系统的宕机而受到影响和波及。互相不影响。
- 方案二优点：订单，支付其实属于通用逻辑，解决冗余的问题，集中统一管理，方便后续的修改和扩展。
- 方案二优点：节约公司的人员成本，和相对来说节约的服务器的运行成本和资源。
- 方案二缺点：系统全部的压力全部集中古来，并发压力很大，要求系统的容错性，集群部署等问题都随之增大

- 方案二缺点：对技术人员的要求也越来越高。学习也很高

## 2、在那些条件下，会选择哪一个方案

- 项目如果在初期的或者系统相对独立的情况下，可以考虑使用第一种方案，为什么呢？
  - 因为项目在初期一定去追求速度和周期，以及节约公司的预算和成本
  - 项目的架构一定是慢慢随着公司的业务的发展和不断演进的过程，而不是一蹴而就的过程
  - 因为有方案一项目的架构，可以得到很多的技术的沉淀以及业务的得以实现，从技术架构的层面来分析，为后续架构封装和抽离成为的核心基础。
  - 也可以快速的推进公司的各个业务。
- 方案二：一般建议公司的业务已经很稳定，或者要追求更高的并发或服务后续公司业务的扩张和进展，这个可以考虑使用方案二
  - 达到业务的集中集中化处理
  - 形成公共业务模块，减少重复造轮子的过程。
  - 也可以节约很多的人力成本

## 3、你倾向那一种？

根据公司业务在那个阶段，具体去选择那个方案，但是成熟的架构师他一定会有完整的架构解决方案，而不是等着问题出现以后在去思考。而是提前预判和以及公司团队人员的建设以及技能提升都会有一个整体的解决方案。

## 4、如果要实施方案二的，准备步骤和流程

- 进行整体架构的产品白皮书的说明文档。有一个整体的架构逻辑
- 要组织和协调部门召开产品的架构调整会议，然后进行架构的说明和规范以及每个系统的需求分析
- 同时要召开是否要招聘相关开发人员进行人才的补充
- 确定的整体架构中会出现的问题和难点，以及如何公开，如何设计，牵涉技术栈是什么等都要进行细化。
- 然后开始进行整体架构的规范文档，需求文档，表设计文档，框架设计，测试文档等，项目进度管理、版本控制、安全性、服务器配置文档等等
- 协调和组织开发人员进行业务的确认和开发工作
- 产品经理接入管理项目进度，产品的沟通和确认对接，UI的设计和确认，前端的设计和确认，
- 等等

## 03、关于集合**List**的面试题

### 03-01、Java

**ConcurrentModificationException**异常原因和解决方法

## 05、关于线程安全的共享问题

## 06、关于学习方法的问题

后面加入如果找工作的：每个人一定要自己的学习计划和安排。---自己的节奏。

周一~周四：上班时间，一定以自己的工作为主，保证休息时间， -

- 比如今天加班，（你应该不要听直播课，但是我早上的时候，我一定确定自己做一件事情：比如看一个面试题，准备好文章，准备视频，或者晚上要看飞哥录播视频）
- 今天比较闲，那就快速的看录播，写总结和笔记。

周五：

- 睡懒觉，吃饭，开始规划，建议白天如果有约或者要出去玩，就尽情的玩。
- 项目阶段，项目搭建，和部署，和业务的开发一定要定要练习，技术掌握能力



