

极客学院
jikexueyuan.com

Spring入门简介

Spring入门简介 — 课程概要

- Spring的概况
- Spring之IOC
- Spring之AOP

Spring的概况

Spring的概况

- Spring的简介
- Spring的起源
- Spring的优点
- Spring的特点
- Spring的基本框架
- Spring的总结

Spring的概况 – Spring的简介

Spring是一个轻量级控制反转(IoC)和面向切面(AOP)的容器框架，它主要是为了解决企业应用开发的复杂性而诞生的：

- 目的：解决企业应用开发的复杂性
- 功能：使用基本的JavaBean代替EJB
- 范围：任何Java应用

Spring的概况 – Spring的起源

Spring的存在是因为它自身有着得天独厚的优势：

- 它定位的领域是许多其他流行的framework没有的
- Spring是全面的和模块化的
- 它的设计从底部帮助你编写易于测试的代码
- Spring是潜在的一站式解决方案

Spring的概况 - Spring的优点

Spring天生就存在如下的优点：

- 低侵入式设计，代码污染极低
- Write Once, Run Anywhere
- DI有效的降低了耦合度
- AOP提供了通用任务的集中管理
- ORM和DAO简化了对数据库访问
- 高度开放性，并不强制

Spring的概况 - Spring的优点

Spring的优点给开发带来的好处：

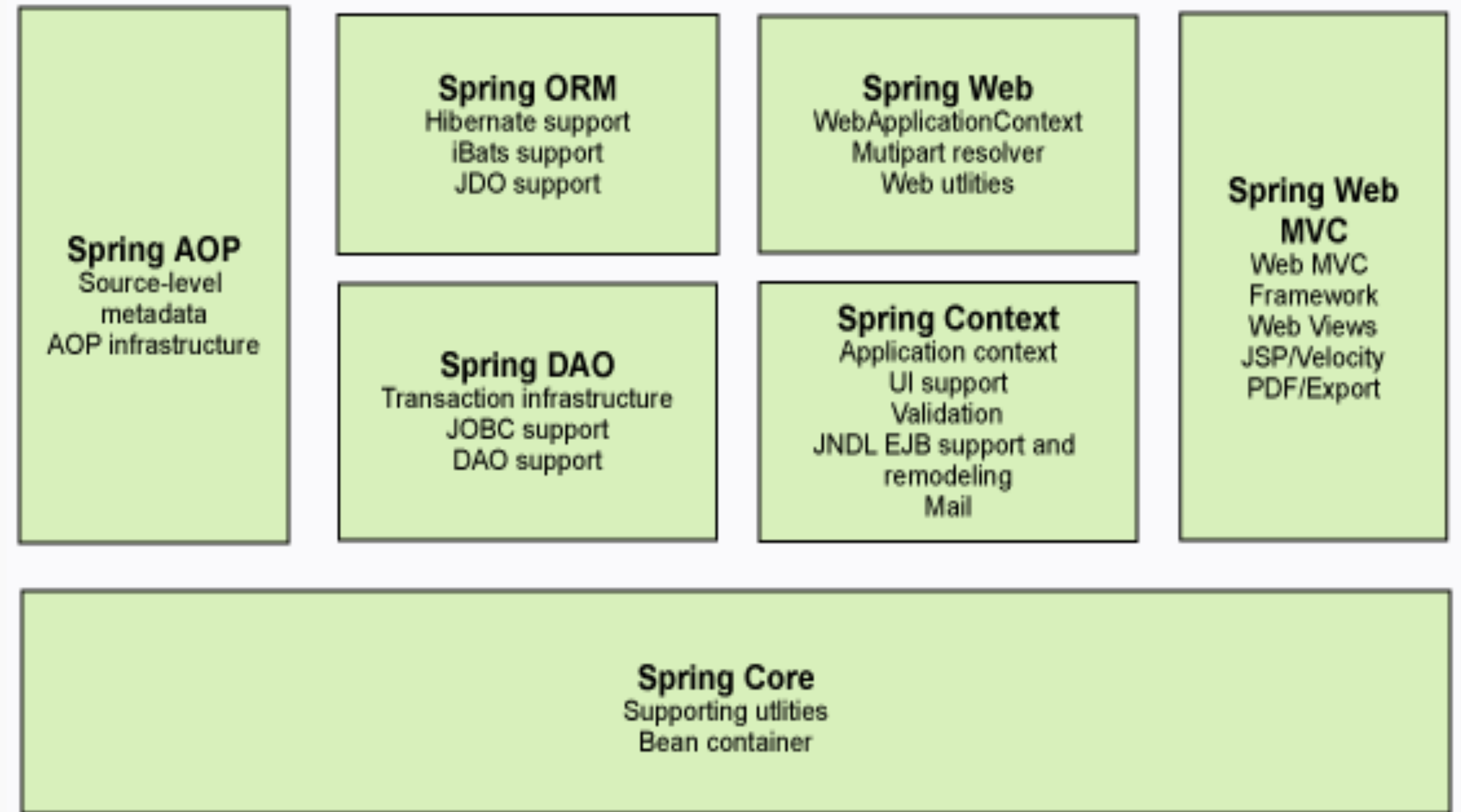
- 可以有效组织中间层对象
- 使用统一的配置文件
- 促进良好编程习惯，减少编程代价
- 易于单元测试
- 使EJB成为一种备选
- 为数据存取提供了一致的框架

Spring的概况 – Spring的特点

- 方便解耦，简化开发
- AOP编程的支持
- 声明式事务的支持
- 方便程序的测试
- 方便集成各种优秀框架
- 降低JavaEE API的使用难度
- Spring的源码是经典学习范例

Spring的概况 – Spring的核心模块

- 核心容器(Spring Core)
- 应用上下文(Spring Context)
- AOP模块(Spring AOP)
- JDBC和DAO模块(Spring DAO)
- 对象实体映射 (Spring ORM)
- Web模块(Spring Web)
- MVC模块(Spring Web MVC)



Spring的概况 – 总结

- Spring带来了复杂的J2EE开发的春天
- 它的核心是轻量级的IoC容器，它的目标是为J2EE应用提供了全方位的整合框架，在Spring框架下实现多个子框架的组合，这些子框架之间可以彼此独立，也可以使用其它的框架方案加以代替
- Spring希望为企业应用提供一站式的解决方案

Spring之IOC

Spring之IOC

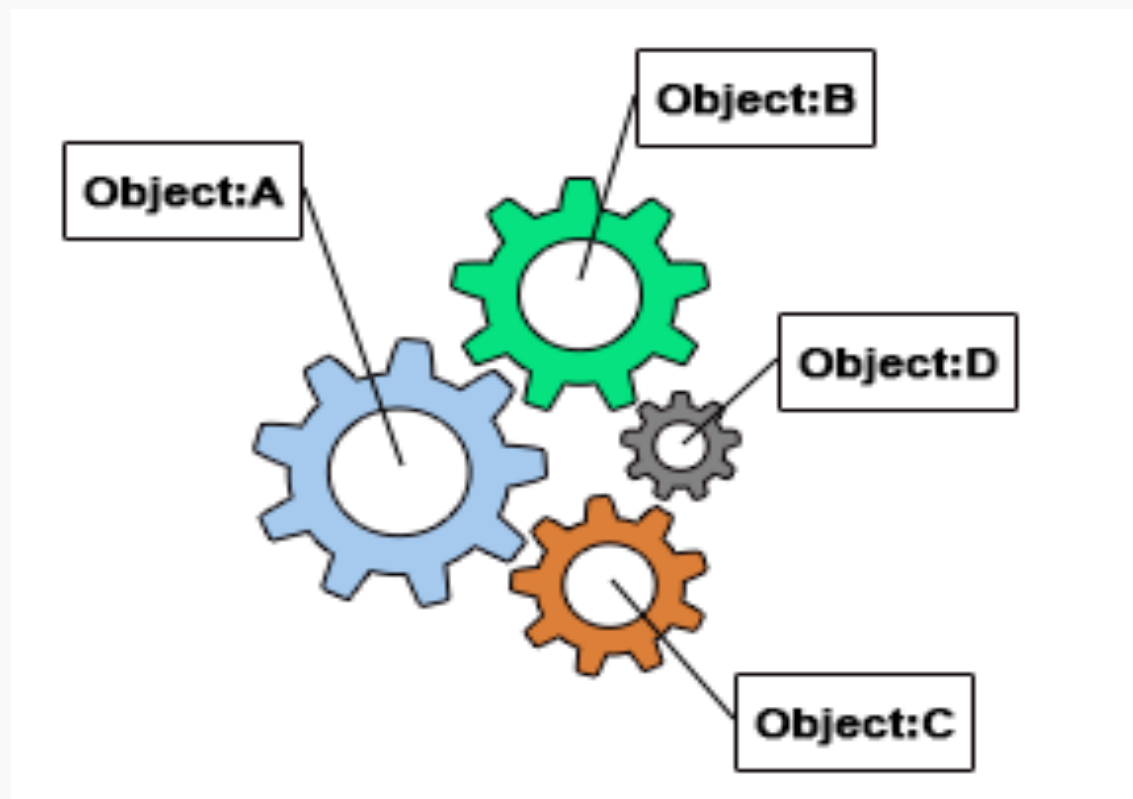
- 浅谈IOC
- IOC理论的背景
- 依赖注入 (DI)
- IOC的好处
- IOC的通俗理解

Spring之IOC – 浅谈IOC

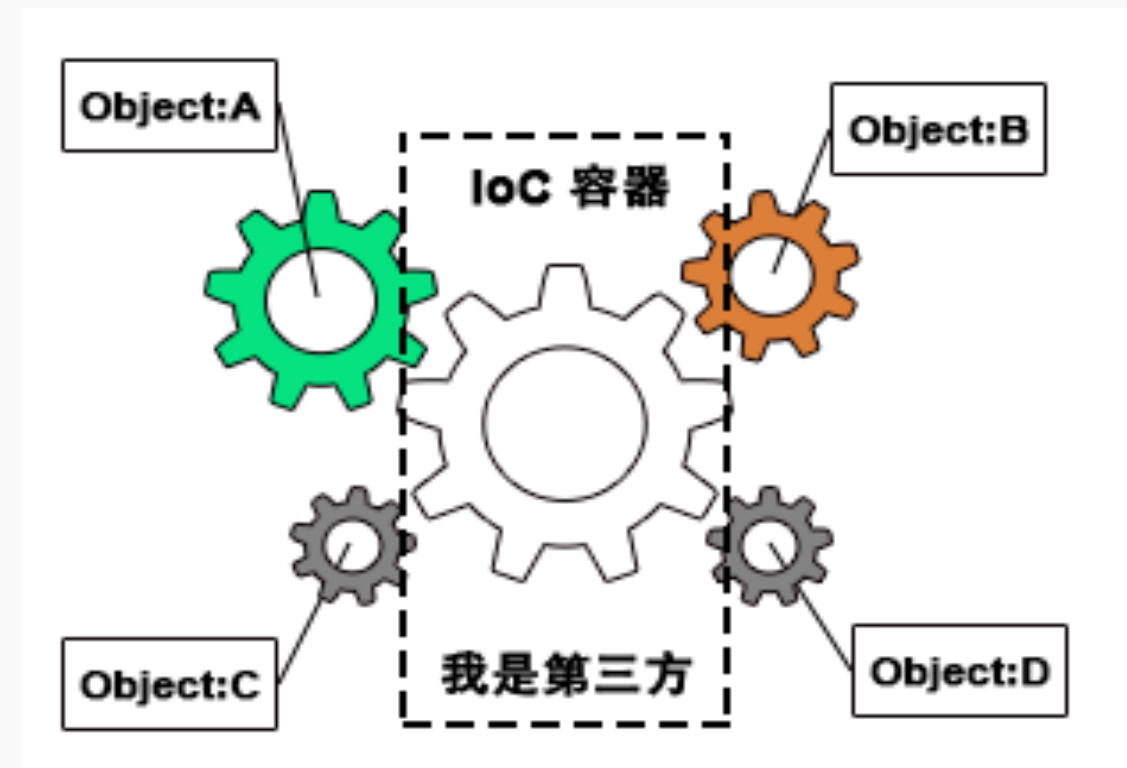
IOC（Inversion of Control，控制反转）是spring的核心，贯穿始终。所谓IOC，对于spring框架来说，就是由spring来负责控制对象的生命周期和对象间的关系：

- 传统开发模式：对象之间互相依赖
- IOC开发模式：IOC容器安排对象之间的依赖

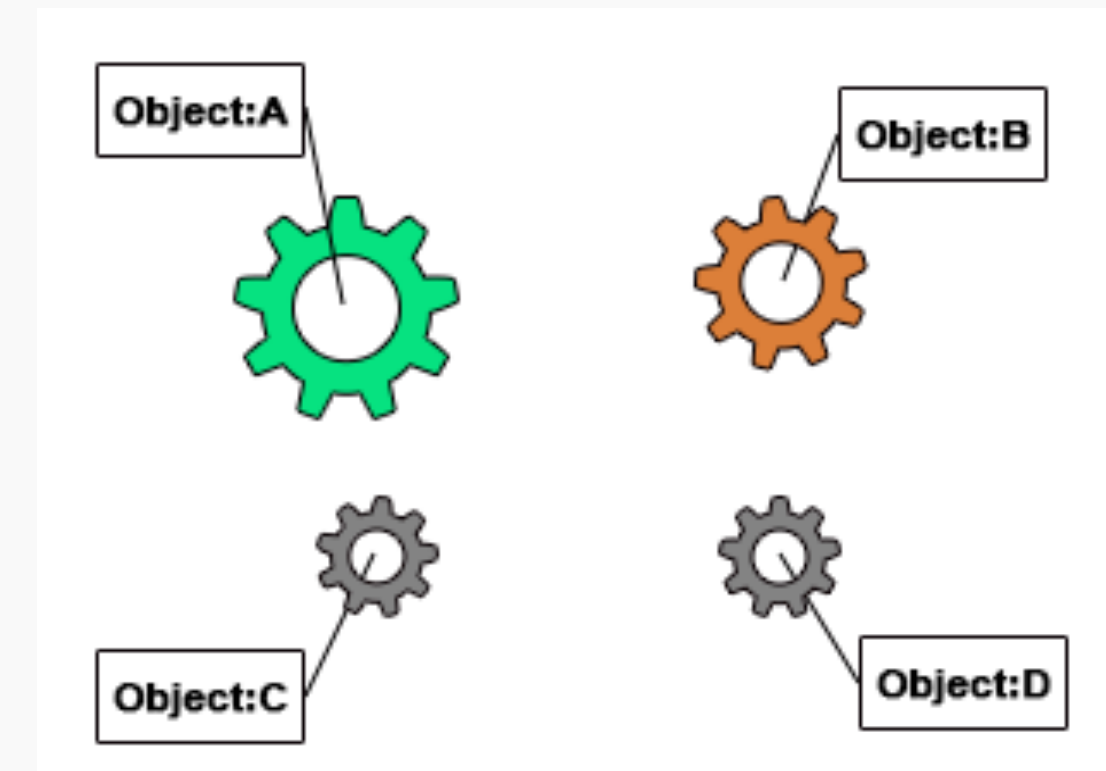
Spring之IOC – IOC理论的背景



- 图1：耦合的对象



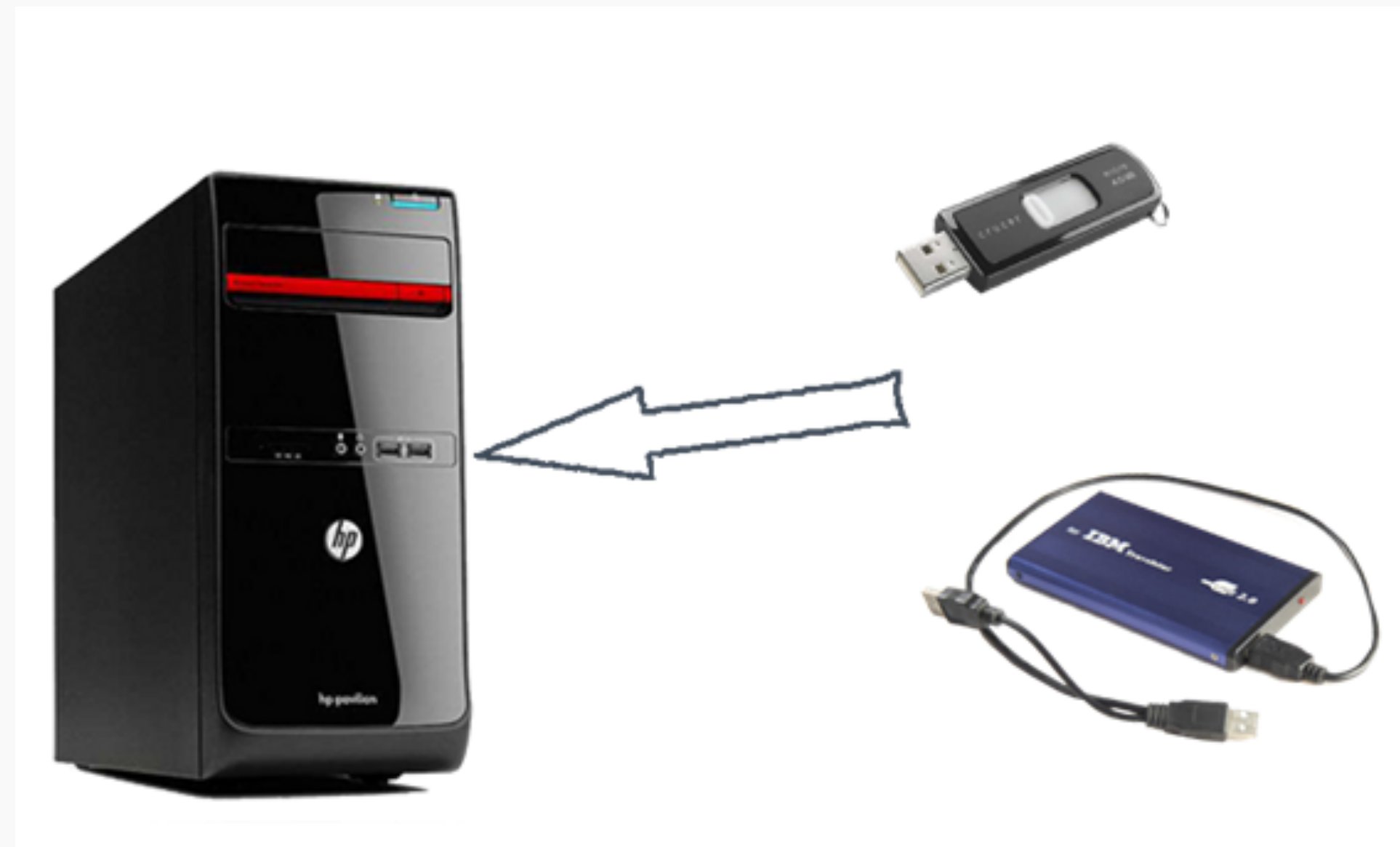
- 图2：解耦的过程



- 图3：理想的系统

Spring之IOC – 依赖注入 (DI)

IOC的另外的名字叫做依赖注入 (Dependency Injection)，所谓的依赖注入，就是由IOC容器在运行期间，动态地将某种依赖关系注入到对象之中。所以，依赖注入(DI)和控制反转(IOC)是从不同的角度的描述的同件事情，就是指通过引入IOC容器，利用依赖关系注入的方式，实现对象之间的解耦



Spring之IOC – IOC的好处

IOC在编程过程中不会对业务对象构成很强的侵入性，使用IOC之后，对象具有更好的可实行性，可重用性和可扩展性：

- 降低组件之间的耦合度
- 提高开发效率和产品质量
- 统一标准，提高模块的复用性
- 模块具有热插拔特性

Spring之IOC – IOC通俗的理解

IOC通俗的理解如下：

- IOC控制反转：说的是创建对象实例的控制权从代码控制剥离到IOC容器控制，实际就是你在xml文件控制，侧重于原理
- DI依赖注入：说的是创建对象实例时，为这个对象注入属性值或其它对象实例，侧重于实现

Spring之AOP

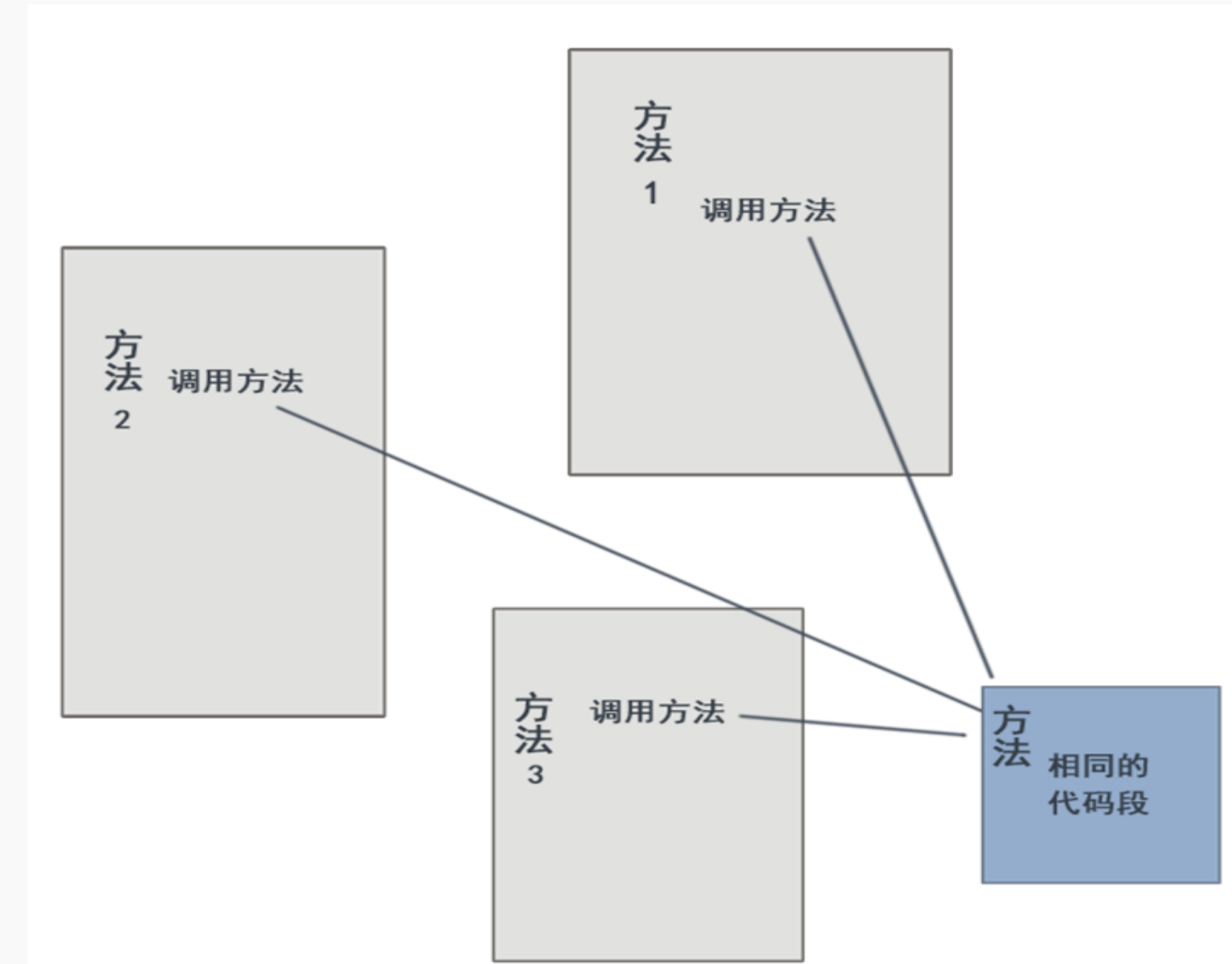
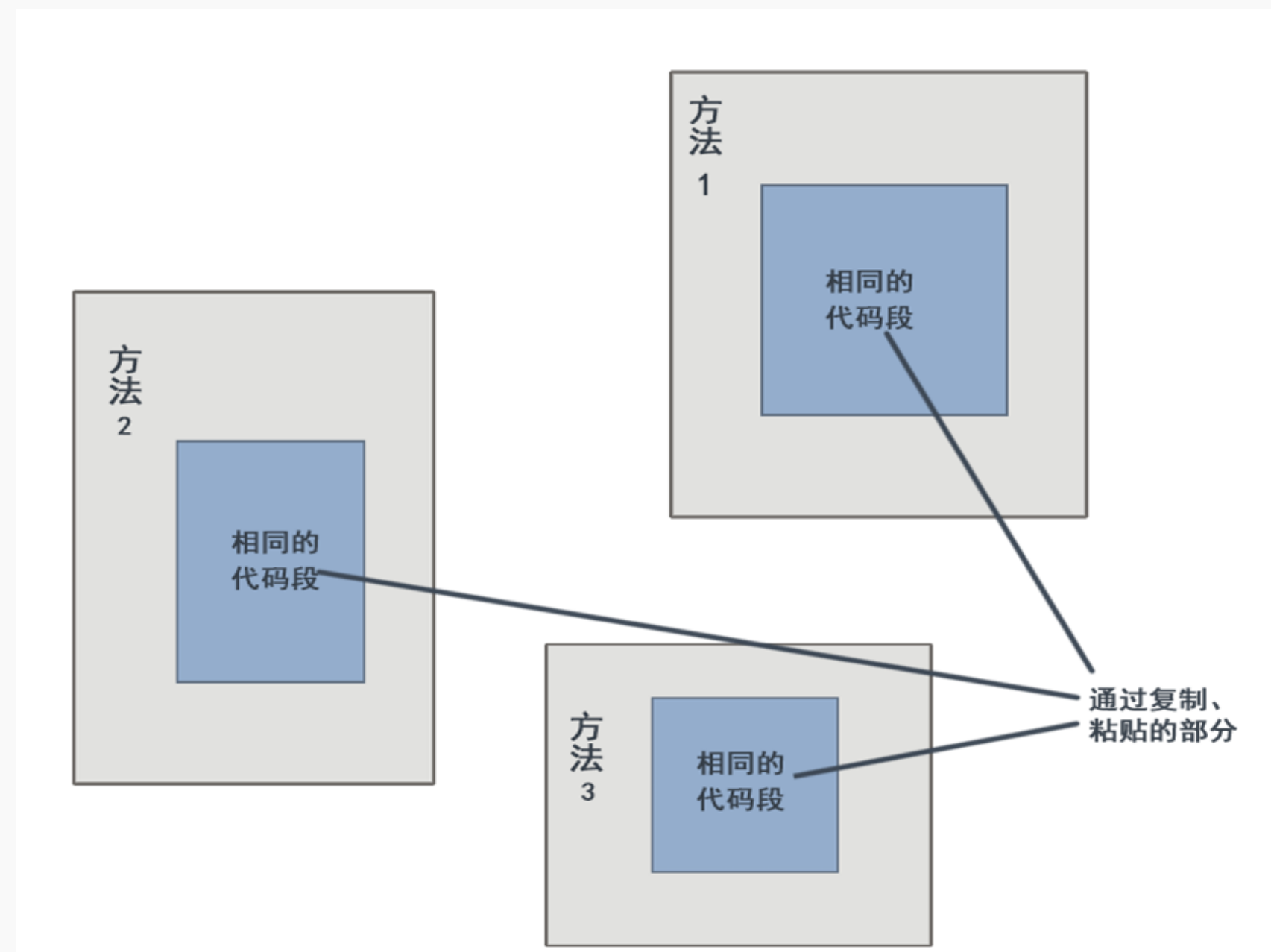
- 什么是AOP
- AOP的存在价值
- AOP的原理剖析
- AOP的关键概念
- AOP的通俗理解

Spring之AOP – 什么是AOP

- AOP的基本概念
- AOP与OOP的关系
- AOP的主要功能
- AOP的主要意图

Spring之AOP – AOP的存在价值

AOP 专门用于处理系统中分布于各个模块中的交叉关注点的问题，在 Java EE 应用中，常常通过 AOP 来处理一些具有横切性质的系统级服务，如事务管理、安全检查、缓存、对象池管理等，AOP 已经成为一种非常常用的解决方案：



Spring之AOP – AOP的原理剖析

AOP 代理其实是由 AOP 框架动态生成的一个对象，该对象可作为目标对象使用，AOP 代理所包含的方法与目标对象的方法如下图所示：



- 定义普通业务组件
- 定义切入点
- 定义增强处理

代理对象的方法 = 增强处理 + 被代理对象的方法

Spring之AOP – AOP的关键概念

以下是官方文档所给出的AOP的关键概念的解释：

- 切面 - Aspect
- 连接点 - Join Point
- 通知 - Advice
- 切入点 - Point Cut
- 引入 - Introduction
- 目标对象 - Target Object
- AOP代理 - AOP Proxy
- 织入 - Weaving

Spring之AOP – AOP的通俗理解

AOP通俗的理解：

- 一个组件A，不关心其他常用的服务组件B，但是这个组件A使用组件B的时候，不是组件A自身去调用，而是通过配置等其他方式，比如Spring中可以通过xml配置文件。这样就使得A压根就不需要知道服务组件B是怎样的，爱存在不存在，爱怎么存在都与A无关。A只关心自己的业务逻辑，具体A使用B的时候，配置文件去做，与具体的A组件无关。

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台

