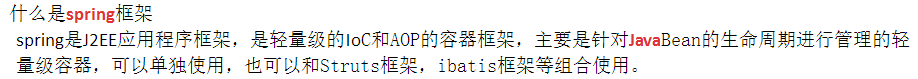
Spring框架介绍

# 什么是Spring框架？



# Spring框架的作用

## Spring框架是用来处理对象的创建以及对象的依赖关系的。

## 使用Spring框架之后最大的改变：程序不再使用new调用构造方法创建Java对象了，所有的Java对象都由Spring容器负责创建。

## 当调用者需要调用被依赖对象的方法时，调用者无须主动获取被依赖对象，只要等待Spring容器注入即可。

Spring容器以松耦合的方式来管理对象之间的依赖关系。

### 原始社会：需要什么，自己造什么；

### 工业社会：需要什么，去相应工厂购买；

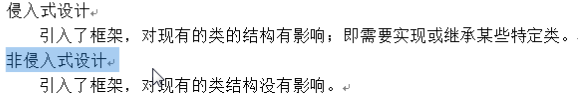
### 共产主义社会：需要什么，无须自己制造，也不需要去工厂买，直接“坐等”社会提供。

# Spring框架的专有名词（专有术语）

## 组件/框架设计

### 侵入式设计：Struts框架

### 非侵入式框架：Hibernate框架、Spring框架。

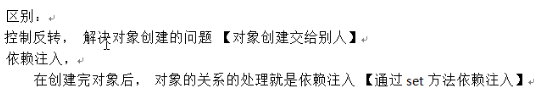




## 控制反转Inversion of Control

对象的创建交给外部容器完成。

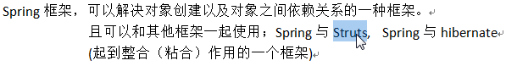
## 依赖注入：处理对象的依赖关系



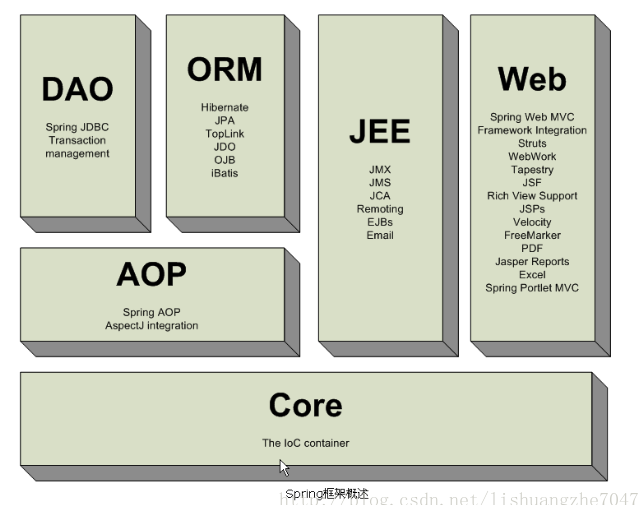
## AOP ：面向切面编程。



# Spring框架：



# Spring框架概述：



## IoC控制反转（Inversion of Control）

**IoC(Inversion of Control)控制反转**，对象创建责任的反转，在spring中BeanFacotory是IoC容器的核心接口，负责实例化，定位，配置应用程序中的对象及建立这些对象间的依赖。**XmlBeanFacotory**实现BeanFactory接口，通过获取xml配置文件数据，组成应用对象及对象间的依赖关系。

**Spring中有三种注入方式，一种是set注入，一种是接口注入，另一种是构造方法注入。**

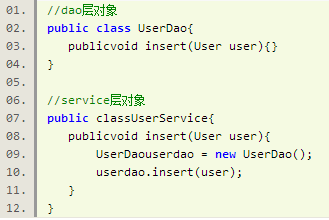
## AOP面向切面编程

AOP就是纵向的编程，如下图所示，业务1和业务2都需要一个共同的操作，与其往每个业务中都添加同样的代码，不如写一遍代码，让两个业务共同使用这段代码。

Spring中面向切面变成的实现有两种方式，**一种是动态代理，一种是CGLIB，动态代理必须要提供接口，而CGLIB实现是有继承。**

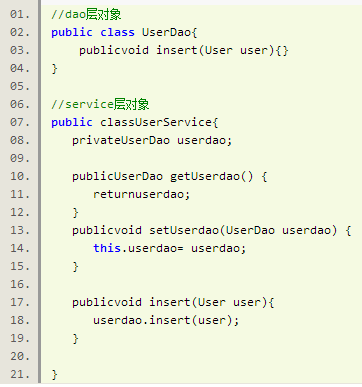
# 为什么要使用Spring框架？

在不使用spring框架之前，我们的service层中要使用dao层的对象，不得不在service层中new一个对象。如下：



**存在的问题：层与层之间的依赖。**

使用框架后：



service层要用dao层对象需要配置到xml配置文件中，至于对象是怎么创建的，关系是怎么组合的都交给了spring框架去实现。

# Spring框架的优点与缺点

## 优点：

### 轻量级的容器框架没有侵入性

### 使用IoC容器更加容易组合对象直接间关系，面向接口编程，降低耦合

### Aop可以更加容易的进行功能扩展，遵循ocp开发原则

### 创建对象默认是单例的，不需要再使用单例模式进行处理

## 缺点：

业务功能依赖spring特有的功能，依赖与spring环境。

# Spring框架是一个非常普通且实用的框架，是从实际开发中抽取出来的框架。

# Spring框架的组成结构图

