# 后台使用Spring框架的质量评估

## 1：什么是spring

## Spring是一个开源框架，Spring是一个轻量级的Java 开发框架，由Rod Johnson 在其著作Expert One-On-One J2EE Development and Design中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许使用者选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。Spring的核心是控制反转（IoC）和面向切面（AOP）。简单来说，Spring是一个分层的JavaSE/EEfull-stack(一站式) 轻量级开源框架。

## 2：常见的spring的特点：

1.方便解耦，简化开发

　　通过Spring提供的IoC容器，我们可以将对象之间的依赖关系交由Spring进行控制，避免硬编码所造成的过度程序耦合。有了Spring，用户不必再为单实例模式类、属性文件解析等这些很底层的需求编写代码，可以更专注于上层的应用。

2.AOP编程的支持

　　通过Spring提供的AOP功能，方便进行面向切面的编程，许多不容易用传统OOP实现的功能可以通过AOP轻松应付。

3.声明事物的支持

　　在Spring中，我们可以从单调烦闷的事务管理代码中解脱出来，通过声明该方式灵活地进行事务的管理，提高开发效率和质量。

4.方便程序的测试

　　可以用非容器依赖的编程方式进行几乎所有的测试工作，在Spring里，测试不再是昂贵的操作，而是随手可做的事情。例如：Spring对Junit4支持，可以通过注解方便的测试Spring程序。

5.方便集成各种优秀框架

　　Spring不排斥各种优秀的开源框架，相反，Spring可以降低各种框架的使用难度，Spring提供了对各种优秀框架（如Struts,Hibernate、Hessian、Quartz）等的直接支持。

6.降低Java EE API的使用难度

　　Spring对很多难用的Java EE API（如JDBC，Java，远程调用等）提供了一个薄薄的封装层，通过Spring的简易封装，这些Java EE API的使用难度大为降低。

7.Java 源码是经典学习范例

　　　　Spring的源码设计精妙、结构清晰、匠心独用，处处体现着大师对Java设计模式灵活运用以及对Java技术的高深造诣。Spring框架源码无疑是Java技术的最佳实践范例。如果想在短时间内迅速提高自己的Java技术水平和应用开发水平，学习和研究Spring源码将会使你收到意想不到的效果。

## 3：使用spring的优点：

　Spring能有效地组织你的中间层对象，无论你是否选择使用了EJB。如果你仅仅使用了Struts或其他的包含了J2EE特有APIs的framework，你会发现Spring关注了遗留下的问题。Spring能消除在许多工程上对Singleton的过多使用。根据我的经验，这是一个主要的问题，它减少了系统的可测试性和面向对象特性。

　　Spring能消除使用各种各样格式的属性定制文件的需要，在整个应用和工程中，可通过一种一致的方法来进行配置。曾经感到迷惑，一个特定类要查找迷幻般的属性关键字或系统属性，为此不得不读Javadoc乃至源编码吗？有了Spring，你可很简单地看到类的JavaBean属性。倒置控制的使用（在下面讨论）帮助完成这种简化。

　　Spring能通过接口而不是类促进好的编程习惯，减少编程代价到几乎为零。

　　Spring被设计为让使用它创建的应用尽可能少的依赖于他的APIs。在Spring应用中的大多数业务对象没有依赖于Spring。

使用Spring构建的应用程序易于单元测试。

　　Spring能使EJB的使用成为一个实现选择，而不是应用架构的必然选择。你能选择用POJOs或local EJBs来实现业务接口，却不会影响调用代码。

　　Spring帮助你解决许多问题而无需使用EJB。Spring能提供一种EJB的替换物，它们适于许多web应用。例如，Spring能使用AOP提供声明性事务而不通过使用EJB容器，如果你仅仅需要与单个的数据库打交道，甚至不需要JTA实现。

　　Spring为数据存取提供了一致的框架，不论是使用JDBC或O/R mapping产品（如Hibernate）。

　　Spring确实使你能通过最简单可行的解决办法解决你的问题。这些特性是有很大价值的。

总结起来，Spring有如下优点：

1.低侵入式设计，代码污染极低

2.独立于各种应用服务器，基于Spring框架的应用，可以真正实现Write Once,Run Anywhere的承诺

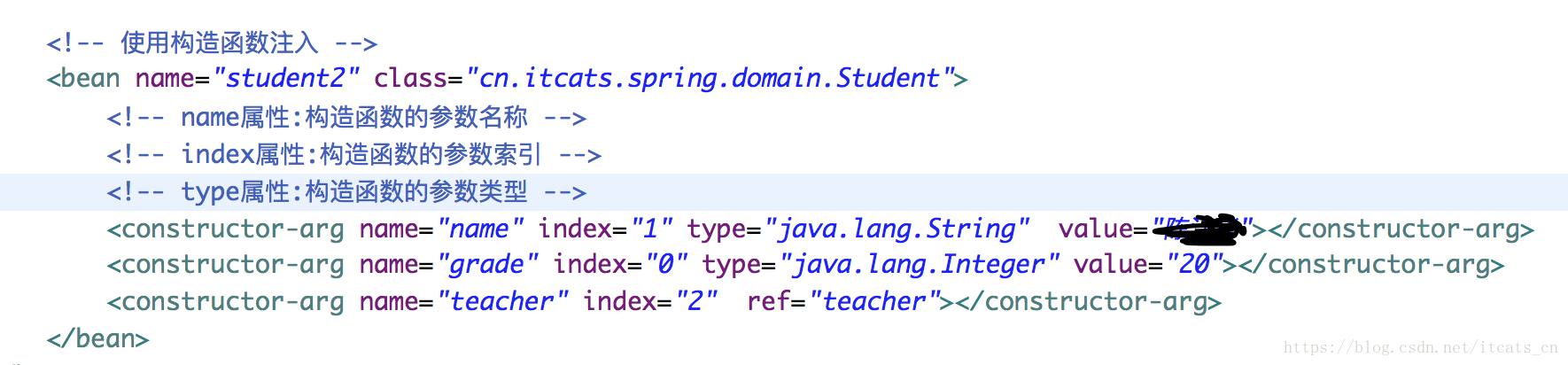
3.Spring的DI机制降低了业务对象替换的复杂性，提高了组件之间的解耦

4.Spring的AOP支持允许将一些通用任务如安全、事务、日志等进行集中式管理，从而提供了更好的复用

5.Spring的ORM和DAO提供了与第三方持久层框架的良好整合，并简化了底层的数据库访问

6.Spring并不强制应用完全依赖于Spring，开发者可自由选用Spring框架的部分或全部

## 4：spring的注入方式代码：



## 5：对于绿色农产品项目使用Spring框架的质量分析

使用Spring能有效地组织你的中间层对象，无论你是否选择使用了EJB。如果你仅仅使用了Struts或其他的包含了J2EE特有APIs的framework，你会发现Spring关注了遗留下的问题。Spring能消除在许多工程上对Singleton的过多使用。

并且在绿色农产品开发中Spring能消除使用各种各样格式的属性定制文件的需要，在使用应用和工程中，可通过一种一致的方法来进行配置。曾经感到迷惑，一个特定类要查找迷幻般的属性关键字或系统属性，有了Spring，你可很简单地看到类的JavaBean属性。

Spring能通过接口而不是类促进好的编程习惯，减少编程代价到几乎为零。

Spring被设计为让使用它创建的应用尽可能少的依赖于他的APIs。在Spring应用中的大多数业务对象没有依赖于Spring。

使用Spring构建的应用程序易于单元测试。

Spring能使EJB的使用成为一个实现选择，而不是应用架构的必然选择。你能选择用POJOs来实现业务接口，却不会影响调用代码。

Spring帮助你解决许多问题而无需使用EJB。Spring能提供一种EJB的替换物，它们适于许多web应用。例如，Spring能使用AOP提供声明性事务而不通过使用EJB容器，如果你仅仅需要与单个的数据库打交道，甚至不需要JTA实现。

Spring确实使你能通过最简单可行的解决办法解决你的问题。这些特性是有很大价值的

总之，使用spring框架对项目开发有这至关重要的作用，该项目正是结合了spring这些优点，是绿色农产品开发更加容易，方便。