1. Spring简介

Spring 也表示是一个开源框架，是为了解决企业[应用程序开发](http://baike.baidu.com/view/1753085.htm" \t "_blank)复杂性由Rod Johnson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。

　Rod Johnson在2002年编著的《Expert one to one J2EE design and development》一

书中，对Java EE正统框架臃肿、低效、脱离现实的种种现状提出了质疑，并积极寻求探索革新之道。以此书为指导思想，他编写了interface21框架，这是一个力图冲破Java EE传统开发的困境，从实际需求出发，着眼于轻便、灵巧，易于开发、测试和部署的轻量级开发框架。Spring框架即以interface21框架为基础，经过重新设计，并不断丰富其内涵，于2004年3月24日，发布了1.0正式版。同年他又推出了一部堪称经典的力作《Expert one-to-one J2EE Development without EJB》，该书在Java世界掀起了轩然大波，不断改变着Java开发者程序设计和开发的思考方式。在该书中，作者根据自己多年丰富的实践经验，对[EJB](http://baike.baidu.com/view/3542.htm" \t "_blank)的各种笨重臃肿的结构进行了逐一的分析和否定，并分别以简洁实用的方式替换之。至此一战功成，Rod Johnson成为一个改变Java世界的大师级人物。

　　传统J2EE应用的开发效率低，应用服务器厂商对各种技术的支持并没有真正统一，导致J2EE的应用没有真正实现Write Once及Run Anywhere的承诺。Spring作为开源的中间件，独立于各种应用服务器，甚至无须应用服务器的支持，也能提供应用服务器的功能，如声明式事务等。

Spring致力于J2EE应用的各层的解决方案，而不是仅仅专注于某一层的方案。可以说Spring是企业应用开发的“一站式”选择，并贯穿表现层、业务层及持久层。然而，Spring并不想取代那些已有的框架，而是与它们无缝地整合。

Spring是一个开源框架，它由Rod Johnson创建。它是为了解决企业应用开发的复杂性而创建的。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。

◆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，并提供了更多的企业应用功能

◆范围：任何Java应用

简单来说，Spring是一个轻量级的控制反转([IoC](http://baike.baidu.com/view/146665.htm" \t "_blank))和面向切面([AOP](http://baike.baidu.com/view/73626.htm))的容器框架。

◆**轻量**——从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

◆**控制反转**——Spring通过一种称作控制反转（IoC）的技术促进了松耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

◆**面向切面**——Spring提供了[面向切面编程](http://baike.baidu.com/view/1865230.htm" \t "_blank)的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

◆**容器**——Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置[原型](http://baike.baidu.com/view/228368.htm" \t "_blank)（prototype），你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。

◆**框架**——Spring可以将简单的组件配置、组合成为复杂的应用。在Spring中，应用对象被声明式地组合，典型地是在一个XML文件里。Spring也提供了很多基础功能（事务管理、持久化框架集成等等），将应用逻辑的开发留给了你。

所有Spring的这些特征使你能够编写更干净、更可管理、并且更易于测试的代码。它们也为Spring中的各种模块提供了基础支持。

　Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式，如图所

[](http://baike.baidu.com/image/d1571724cc23f467d40742ac)

Spring

示：

组成Spring框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。每个模块的功能如下：1、核心[容器](http://baike.baidu.com/view/864334.htm" \t "_blank)：核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是[工厂模式](http://baike.baidu.com/view/1306799.htm" \t "_blank)的实现。BeanFactory 使用[控制反转](http://baike.baidu.com/view/1486379.htm" \t "_blank) （IOC） 模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。

2、Spring 上下文：Spring 上下文是一个[配置文件](http://baike.baidu.com/view/2117618.htm" \t "_blank)，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 [JNDI](http://baike.baidu.com/view/209575.htm)、EJB、[电子邮件](http://baike.baidu.com/view/1524.htm" \t "_blank)、国际化、校验和调度功能。

3、Spring AOP：通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。

4、Spring DAO：JDBC [DAO](http://baike.baidu.com/view/160596.htm) 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。

5、Spring ORM：Spring 框架插入了若干个 [ORM](http://baike.baidu.com/view/197951.htm) 框架，从而提供了 ORM 的对象关系工具，其中包括 [JDO](http://baike.baidu.com/view/73546.htm)、[Hibernate](http://baike.baidu.com/view/7291.htm" \t "_blank) 和 [iBatis](http://baike.baidu.com/view/628102.htm) SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。

6、Spring Web 模块：Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。

7、Spring MVC 框架：[MVC](http://baike.baidu.com/view/31.htm" \t "_blank) 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成为高度可配置的，MVC 容纳了大量[视图](http://baike.baidu.com/view/71981.htm" \t "_blank)技术，其中包括 JSP、[Velocity](http://baike.baidu.com/view/77351.htm" \t "_blank)、[Tiles](http://baike.baidu.com/view/25374.htm" \t "_blank)、iText 和 POI。Spring 框架的功能可以用在任何 [J2EE](http://baike.baidu.com/view/1507.htm) 服务器中，大多数功能也适用于不受管理的环境。Spring 的核心要点是：支持不绑定到特定 J2EE 服务的可重用业务和[数据访问对象](http://baike.baidu.com/view/185335.htm" \t "_blank)。毫无疑问，这样的对象可以在不同J2EE 环境（Web 或EJB）、独立应用程序、测试环境之间重用。

　控制反转模式（也称作依赖性注入）的基本概念是：不创建对象，但是描述创建它们的方式。在代码中不直接与[对象](http://baike.baidu.com/view/2387.htm" \t "_blank)和[服务](http://baike.baidu.com/view/133203.htm)连接，但在配置文件中描述哪一个组件需要哪一项服务。容器 （在 Spring 框架中是 IOC 容器） 负责将这些联系在一起。在典型的 IOC 场景中，容器创建了所有对象，并设置必要的属性将它们连接在一起，决定什么时间调用方法。下表列出了 IOC 的一个实现模式。

|  |  |
| --- | --- |
| 类型 1 | 服务需要实现专门的接口，通过接口，由对象提供这些服务，可以从对象查询[依赖性](http://baike.baidu.com/view/3012900.htm" \t "_blank)（例如，需要的附加服务）。 |
| 类型 2 | 通过 [JavaBean](http://baike.baidu.com/view/28155.htm) 的属性（例如 setter 方法）分配依赖性。 |
| 类型 3 | 依赖性以[构造函数](http://baike.baidu.com/view/411124.htm" \t "_blank)的形式提供，不以 JavaBean 属性的形式公开。 |

Spring 框架的 IOC 容器采用类型 2 和类型3 实现。

面向方面的编程，即 AOP，是一种编程技术，它允许程序员对横切关注点或横切典型的职责[分界线](http://baike.baidu.com/view/948516.htm" \t "_blank)的行为（例如日志和事务管理）进行模块化。AOP 的核心构造是方面，它将那些影响多个类的行为封装到可重用的模块中。

AOP 和 IOC 是补充性的技术，它们都运用模块化方式解决企业[应用程序开发](http://baike.baidu.com/view/1753085.htm" \t "_blank)中的复杂问题。在典型的面向对象开发方式中，可能要将日志记录语句放在所有方法和 Java 类中才能实现日志功能。在 AOP 方式中，可以反过来将[日志](http://baike.baidu.com/view/269351.htm" \t "_blank)服务模块化，并以声明的方式将它们应用到需要日志的[组件](http://baike.baidu.com/view/379950.htm)上。当然，优势就是 Java 类不需要知道日志服务的存在，也不需要考虑相关的代码。所以，用 Spring AOP 编写的应用程序代码是松散耦合的。

　　AOP 的功能完全集成到了 Spring [事务管理](http://baike.baidu.com/view/3871410.htm)、日志和其他各种特性的上下文中。

**IOC容器**

Spring 设计的核心是 org.springframework.beans 包，它的设计目标是与 JavaBean 组件一起使用。这个包通常不是由用户直接使用，而是由服务器将其用作其他多数功能的底层中介。下一个最高级抽象是 BeanFactory 接口，它是工厂设计模式的实现，允许通过名称创建和检索对象。BeanFactory 也可以管理对象之间的关系。

BeanFactory 支持两个对象模型。

1、[单态](http://baike.baidu.com/view/1293882.htm) 模型提供了具有特定名称的对象的共享实例，可以在查询时对其进行检索。Singleton 是默认的也是最常用的[对象模型](http://baike.baidu.com/view/2215158.htm)。对于无状态服务对象很理想。

2、原型 模型确保每次检索都会创建单独的对象。在每个用户都需要自己的对象时，原型模型最适合。

bean 工厂的概念是 Spring 作为 IOC 容器的基础。IOC 将处理事情的责任从应用程序代码转移到框架。