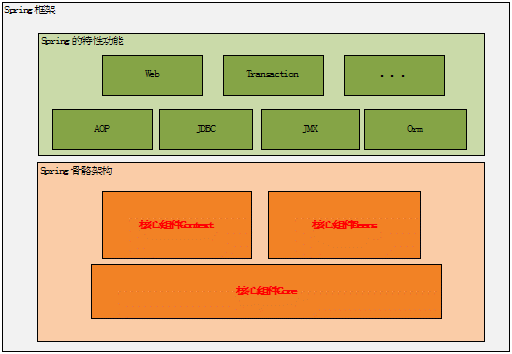
# Spring 的骨骼架构

Spring 总共有十几个组件，但是真正核心的组件只有几个，下面是 Spring 框架的总体架构图：

Spring框架的总体架构图:



从上图中可以看出 Spring 框架中的核心组件只有三个：Core、Context 和 Beans。它们构建起了整个 Spring 的骨骼架构。没有它们就不可能有 AOP、Web 等上层的特性功能。下面也将主要从这三个组件入手分析 Spring。

# Spring 的设计理念

前面介绍了 Spring 的三个核心组件，如果再在它们三个中选出核心的话，那就非 Beans 组件莫属了，为何这样说，其实 Spring 就是面向 Bean 的编程（BOP,Bean Oriented Programming），Bean 在 Spring 中才是真正的主角。

Bean 在 Spring 中作用就像 Object 对 OOP 的意义一样，没有对象的概念就像没有面向对象编程，Spring 中没有 Bean 也就没有 Spring 存在的意义。就像一次演出舞台都准备好了但是却没有演员一样。为什么要 Bean 这种角色 Bean 或者为何在 Spring 如此重要，这由 Spring 框架的设计目标决定，Spring 为何如此流行，我们用 Spring 的原因是什么，想想你会发现原来 Spring 解决了一个非常关键的问题他可以让你把对象之间的依赖关系转而用配置文件来管理，也就是他的依赖注入机制。而这个注入关系在一个叫 Ioc 容器中管理，那 Ioc 容器中有又是什么就是被 Bean 包裹的对象。Spring 正是通过把对象包装在 Bean 中而达到对这些对象管理以及一些列额外操作的目的。

它这种设计策略完全类似于 Java 实现 OOP 的设计理念，当然了 Java 本身的设计要比 Spring 复杂太多太多，但是都是构建一个数据结构，然后根据这个数据结构设计他的生存环境，并让它在这个环境中按照一定的规律在不停的运动，在它们的不停运动中设 计一系列与环境或者与其他个体完成信息交换。这样想来回过头想想我们用到的其他框架都是大慨类似的设计理念。

# 核心组件如何协同工作

前面说 Bean 是 Spring 中关键因素，那 Context 和 Core 又有何作用呢？前面吧 Bean 比作一场演出中的演员的话，那 Context 就是这场演出的舞台背景，而 Core 应该就是演出的道具了。只有他们在一起才能具备能演出一场好戏的最基本的条件。当然有最基本的条件还不能使这场演出脱颖而出，还要他表演的节目足够的精 彩，这些节目就是 Spring 能提供的特色功能了。

我们知道 Bean 包装的是 Object，而 Object 必然有数据，如何给这些数据提供生存环境就是 Context 要解决的问题，对 Context 来说他就是要发现每个 Bean 之间的关系，为它们建立这种关系并且要维护好这种关系。所以 Context 就是一个 Bean 关系的集合，这个关系集合又叫 Ioc 容器，一旦建立起这个 Ioc 容器后 Spring 就可以为你工作了。那 Core 组件又有什么用武之地呢？其实 Core 就是发现、建立和维护每个 Bean 之间的关系所需要的一些列的工具，从这个角度看来，Core 这个组件叫 Util 更能让你理解。

## 三个组件关系

## 

# 核心组件详解

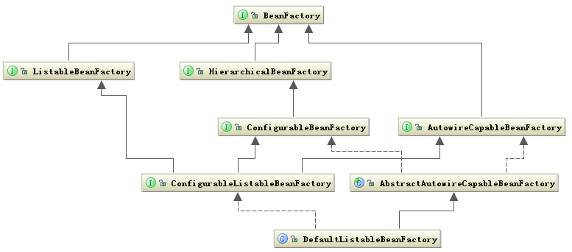
这里将详细介绍每个组件内部类的层次关系，以及它们在运行时的时序顺序。我们在使用 Spring 是应该注意的地方。

## **Bean 组件**

前面已经说明了 Bean 组件对 Spring 的重要性，下面看看 Bean 这个组件式怎么设计的。Bean 组件在 Spring 的 org.springframework.beans 包下。这个包下的所有类主要解决了三件事：Bean 的定义、Bean 的创建以及对 Bean 的解析。对 Spring 的使用者来说唯一需要关心的就是 Bean 的创建，其他两个由 Spring 在内部帮你完成了，对你来说是透明的。

Spring Bean 的创建时典型的工厂模式，他的顶级接口是 BeanFactory，下图是这个工厂的继承层次关系：

### Bean工厂的继承关系



BeanFactory 有三个子类：ListableBeanFactory、HierarchicalBeanFactory 和 AutowireCapableBeanFactory。但是从上图中我们可以发现最终的默认实现类是 DefaultListableBeanFactory，他实现了所有的接口。那为何要定义这么多层次的接口呢？查阅这些接口的源码和说明发现，每个接口 都有他使用的场合，它主要是为了区分在 Spring 内部在操作过程中对象的传递和转化过程中，对对象的数据访问所做的限制。例如 ListableBeanFactory 接口表示这些 Bean 是可列表的，而 HierarchicalBeanFactory 表示的是这些 Bean 是有继承关系的，也就是每个 Bean 有可能有父 Bean。AutowireCapableBeanFactory 接口定义 Bean 的自动装配规则。这四个接口共同定义了 Bean 的集合、Bean 之间的关系、以及 Bean 行为。