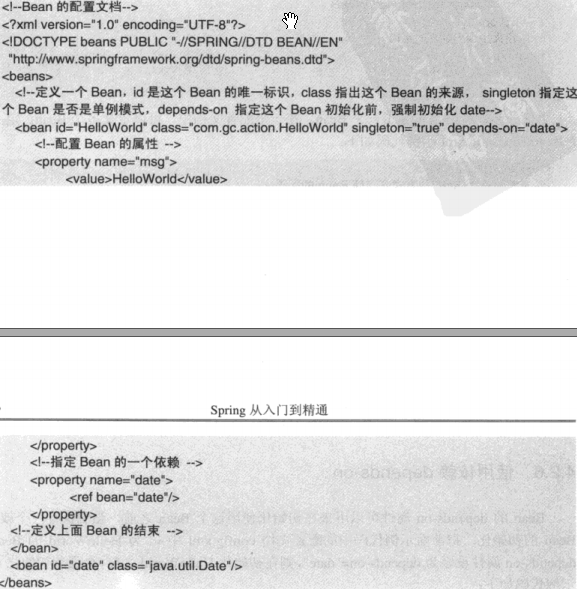
**Spring核心容器**

1. bean的基本属性

Spring依赖注入的三种方式：接口注入，set注入，构造注入



depends-on：初始化HelloWorld时会强制初始化date

singleton：表示项目中只有一个bean实例

reg 有三个属性：local bean parent

1. bean的生命周期

定义->初始化->使用->销毁

1. bean自动装配

byNme模式

byType模式

constructor模式

autodetect模式

no模式

1. bean依赖检查
2. 集合注入
3. **管理bean**

BeanWrapper（很少用）

BeanFactory

ApplicationContext

**Spring的AOP**

1. 概念

面向切面编程，把散落在程序中的公共部分提取出来，做成切面类，这样的好处在于，代码的可重用，一旦涉及到该功能的需求发生变化，只要修改该代码就行，否则需要到处修改

传统OOP（面向对象编程）通过抽象，封装，继承，多态等方法把事物进行抽象和封装，建立一种以对象为基础的层次结构。但是具体到事物的内部，OOP可能会导致大量重复的代码，比如日志，使用OOP会导致代码散布在所有对象内部。

AOP则解决了这种问题，它利用一种叫横切的技术刨解开对象内部，将那些与核心业务代码无关，并且到处都用到的代码封装成单独的模块。

1. point cut（切入点）

<aop:config>

<aop:pointcut expression=”execution(……)”id=””/>

<aop:aspect id =”” ref=””>

<aop:beforemethod=””/>

<aop:aftermethod=””/>

</aop:aspect>

</aop:config>