

# 编程提高群第三季

刘欣

微信: onlyliuxin97

微信公众号: 码农翻身(coderising)



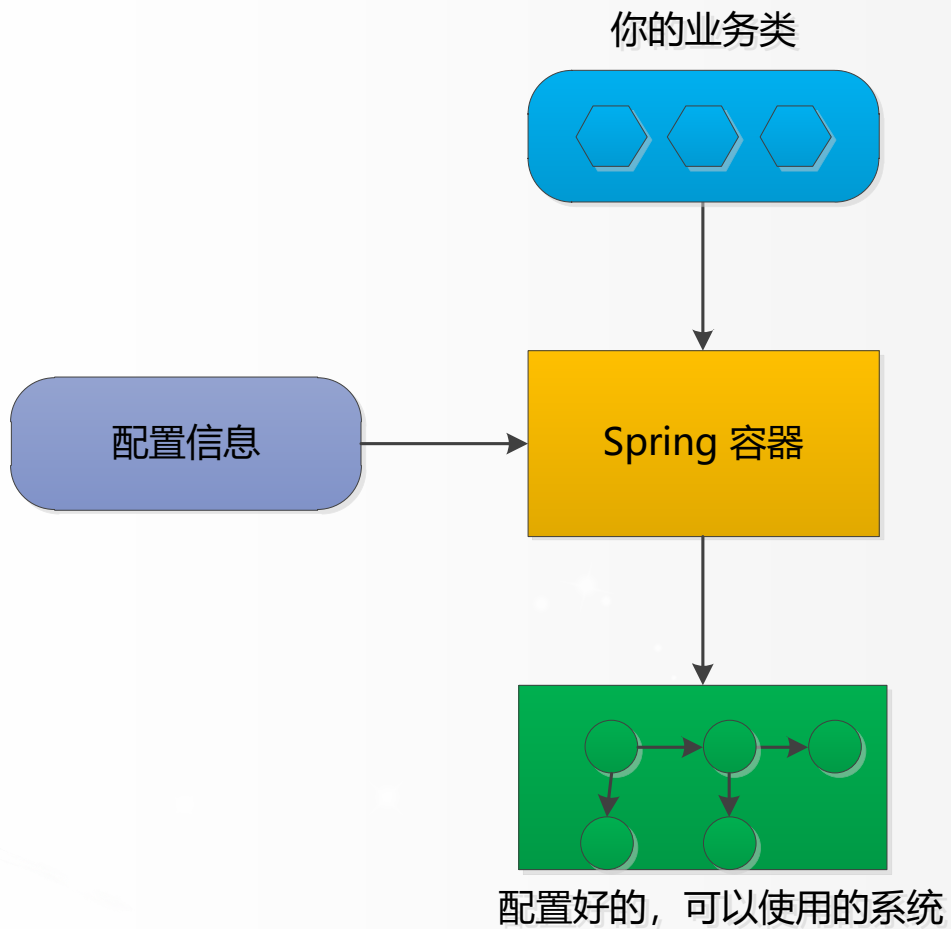
# 目标：从零开始造Spring

- 通读Spring 源码： **从入门到放弃!**
- 玩具IoC和AOP框架？ **NO!**
- **从零开始，一步步逼近Spring**
  - 要和Spring足够像，保持Spring的设计
  - 从最简单的结构开始，不断添砖加瓦
  - TDD的方法
  - 重点：学习Spring的设计思想，核心的抽象

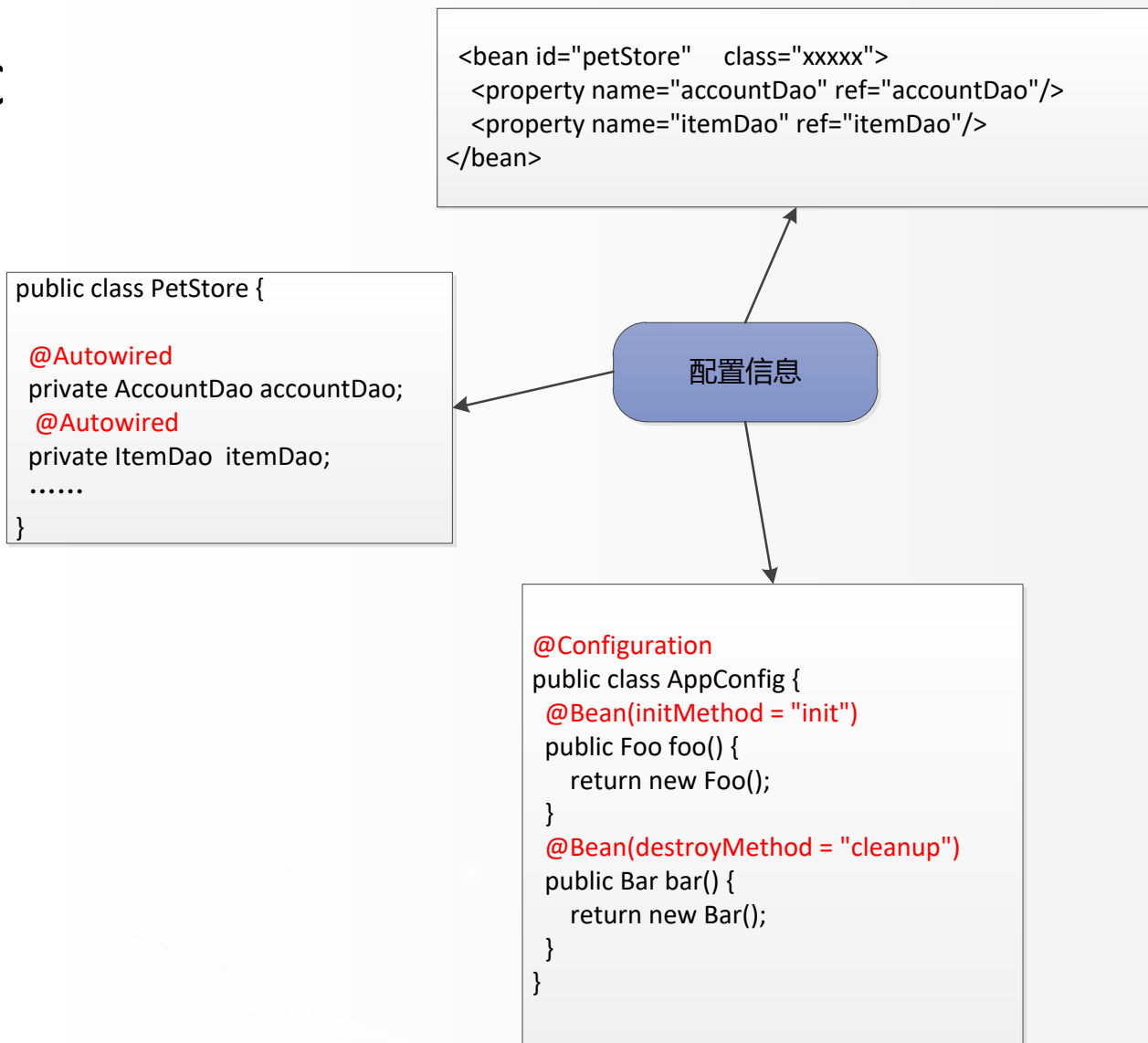
## Spring 3.2.18

- IoC和AOP已经很完整
- 支持注解
- 下载地址：
  - <http://repo.spring.io/libs-release/org/springframework/spring/>

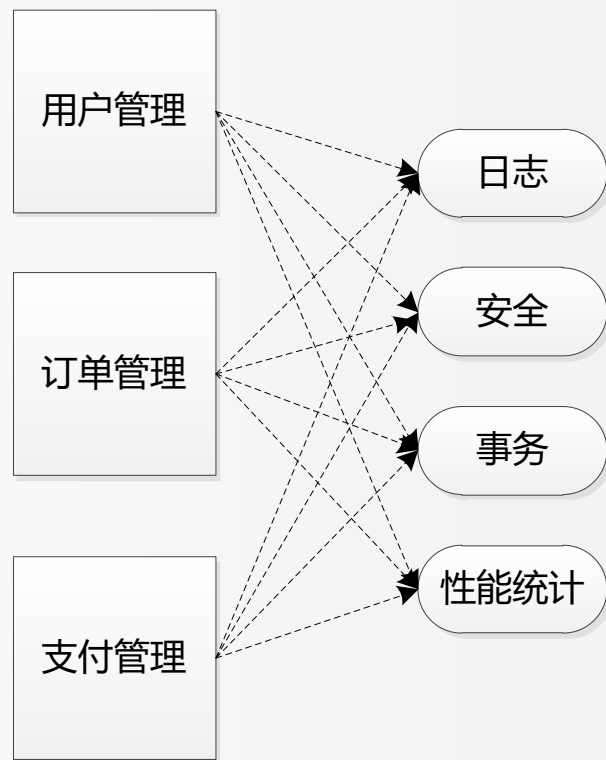
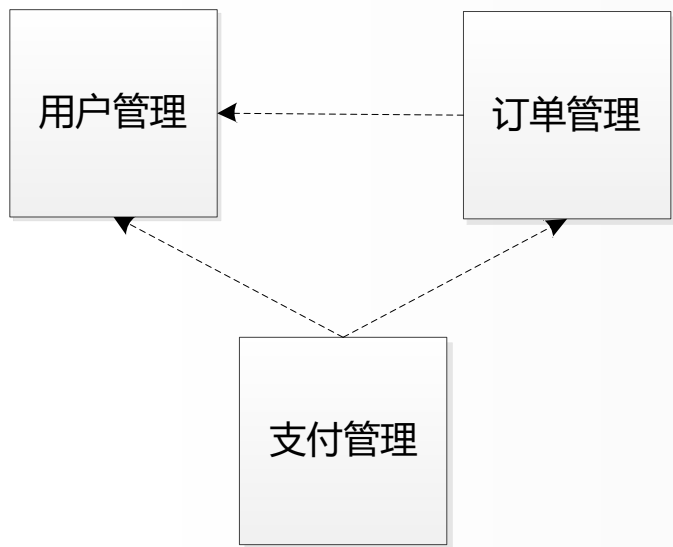
# IoC

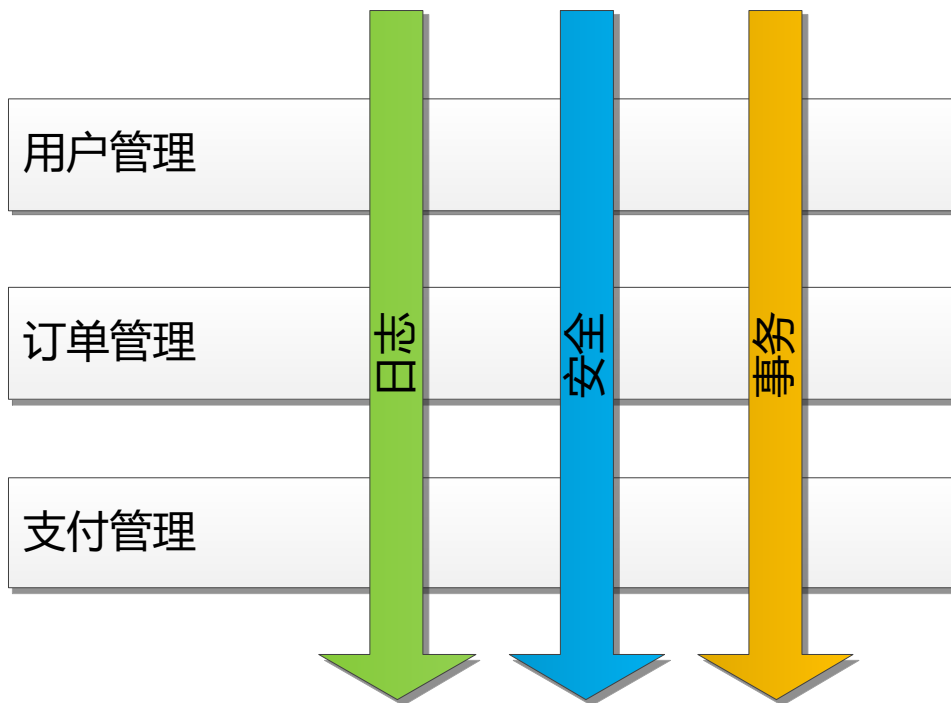


# IoC



# AOP



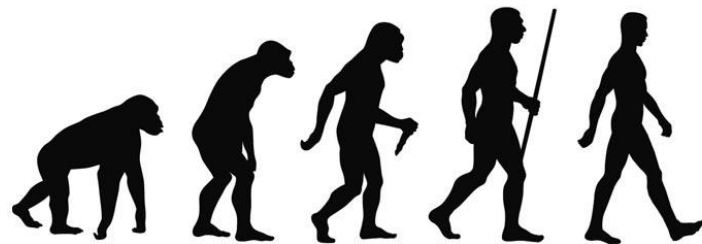


```
<bean id="tx" class="com.coderising.Transaction" />
<aspect ref="tx">
    <pointcut id="place-order" expression="execution( * com.coderising.*.execucte(..))" />
    <before pointcut-ref="place-order" method="begin" />
    <after pointcut-ref="place-order" method="commit" />
</aspect>
```

# 为什么要用TDD?

- 程序员的两个最基本的技能
- 1. 写测试用例
- 2. 重构

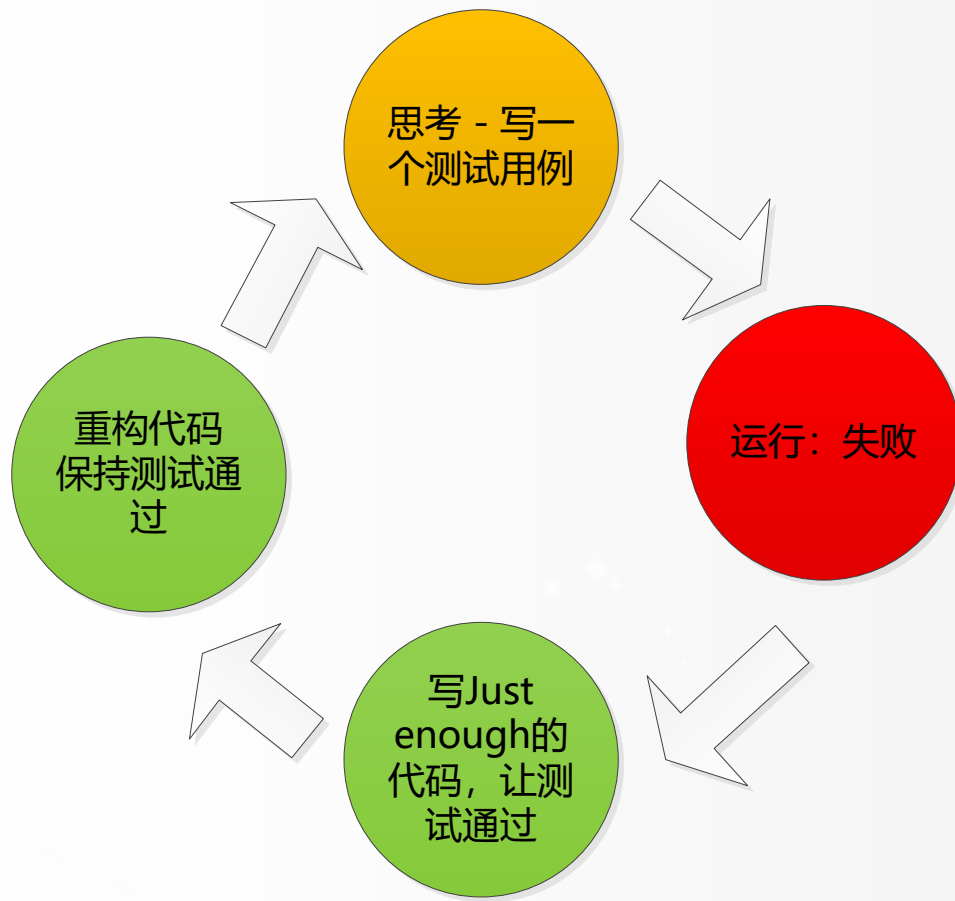
**JUnit**



Refactoring



# 如何去实施TDD?



# TDD Demo

编写一个函数，返回**小于给定值max**的所有素数组成的数组

```
public static int[] getPrimes(int max);
```

# 先做一个简单的任务分解

- 边界条件

- `getPrimes(0)`, `getPrimes(-1)` , `getPrimes(2)` 应该返回什么?

- 正常输入

- `getPrimes(9)`, `getPrimes(17)`, `getPrimes(30)`

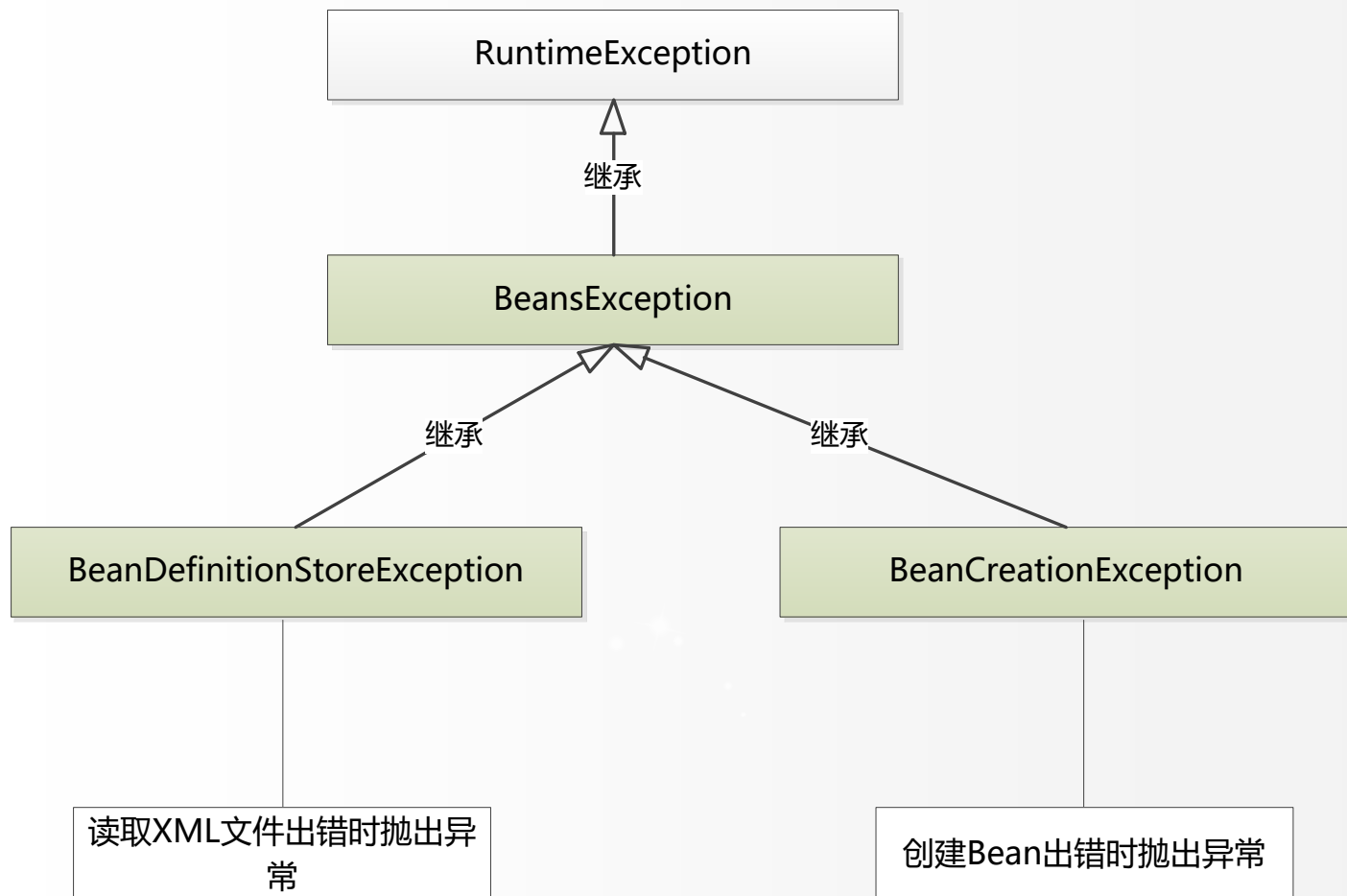
# 期望

- 主动思考
- 坚持
- 动手!!!

# 第一个测试用例

给定一个xml配置的文件（内含bean的定义），能够  
从中获取：

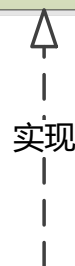
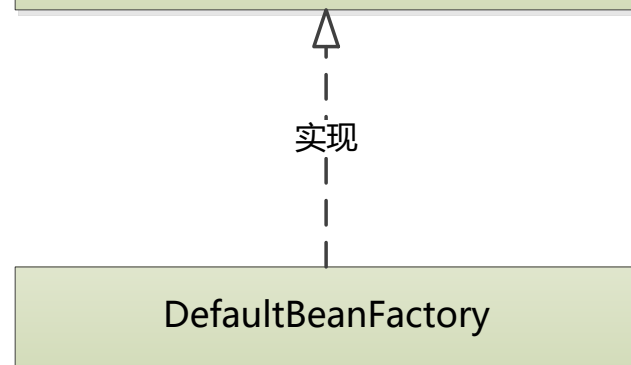
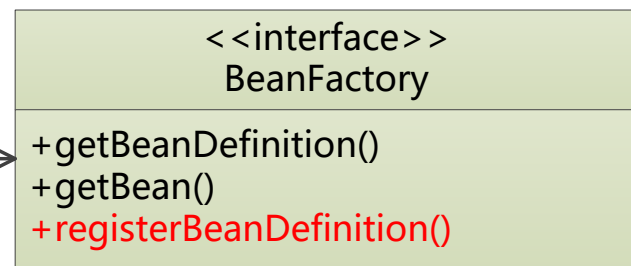
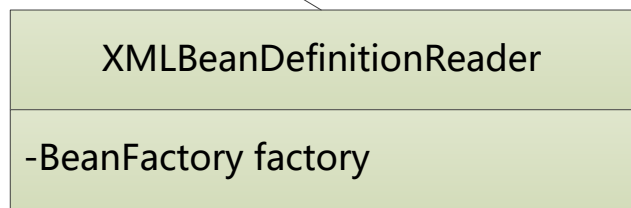
1. Bean的定义
2. Bean 的实例



# SRP： 单一职责原则

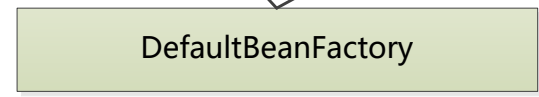
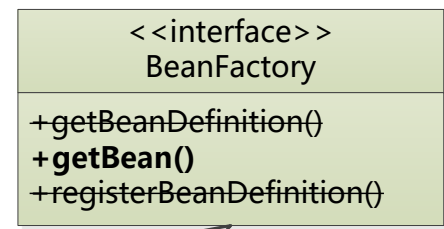
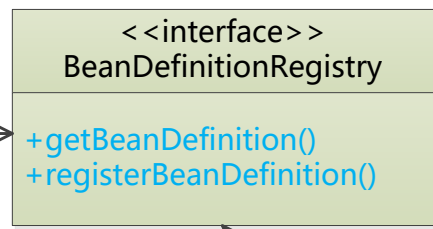
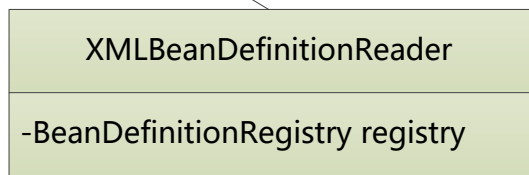
- 职责： 是引起变化的原因
  - 如果有多于一个的动机去改变一个类，这个类就具有多于一个职责
  - 把多个职责耦合在一起，一个的变化可能会削弱或者抑制这个类完成其他职责的能力
- SRP： 对一个类而言， 应该仅有一个引起它变化的原因。

解析XML文件，调用  
registerBeanDefinition  
方法





解析XML文件，调用  
registerBeanDefinitio  
n方法



实现

实现