**IOC / DI in Action Spring**

**Analysis**

**1, Inversion of Control**

a, 控制什么？

控制与之耦合的对象的生命周期

被基础信息委托给某平台让其帮助我们维护复杂的关系

b, 谁控制谁？

平台控制了基础信息对象

c, 为什么说是 反转？什么是正转？

主从关系的思维转变

d, 被反转的是什么？

**2, Dependency Injection**

a, 什么是依赖？

b, 注入的方式有哪些？

New

Setter

Interface Inject( InjetceClass )

c, 谁为谁注入？

d, 什么时候注入？

**生命周期：**

**Object New**

**JVM GC**

**finalize（）**

**Sample**

Huluwa want Buy skin from Yeye.

**IOC / DI in Spring**

1, Bean & Component & POJO

2, Spring 配置的 可选方案

a, 显示配置

JavaConfig.class 中进行配置

XML中进行配置

b, 隐式配置

Bean的自动发现和自动装配机制

3, 区别以及联系：

**Advanced of IOC / DI**

1, 导入和混合配置

2, Bean自动装配的歧义性

3, Bean的作用域

**Summary**

1，Spring IOC: 由Spring IOC的容器来维护注册到容器中Bean 对象的生命周期和对象之间的依赖关系。

2，相比于传统的JavaEE，在开发思维上，体现了主从思维的改变。

3，技术实现上，是在系统运行时（runtime），动态主动的使用DI（Dependency Injection，依赖注入）向某个Bean对象提供它所需要的其他Bean对象。

1. **LLR Guidle Line**
2. **XB 例子更友好**
3. **RH Presentation 提问的应对更友好**
4. **JJ** 
   1. **例子 强焦点**
   2. **讲究技巧**
5. **BH**
6. **同理心**
7. **专注内容的部分，时间太过**

1. **WD**
   1. **问题，Task为导向，驱动感**
   2. **Workshop的形式 带引导**