

Spring架构介绍

与

源码分析



IOC篇

课程目标

- 为什么要学习本课程？
 - spring_ioc**是**spring**全家桶中的核心基础框架，掌握了**ioc**才能更好的学习**aop**，以及事务原理。在学习**springmvc**,**springboot**等框架的时候才能得以应手
- 通过学习本课程能够达到以下效果
 - 掌握**springioc**架构与核心源码，轻松应对项目中疑难杂症
 - 掌握**springioc**如何扩展，开发**spring**插件为应用增加翅膀
 - 介绍与分享一套项目实战中基于**springioc**扩展的插件及源码
- 您将收获以下内容
 - 一套UML图(**SPRING_IOC**架构原理及启动流程)
 - 一套**SPRING**扩展**DEMO**源代码
 - 一套**SPRING**使用**DEMO**源代码
 - 一套**SPRING**插件源代码(实际项目)

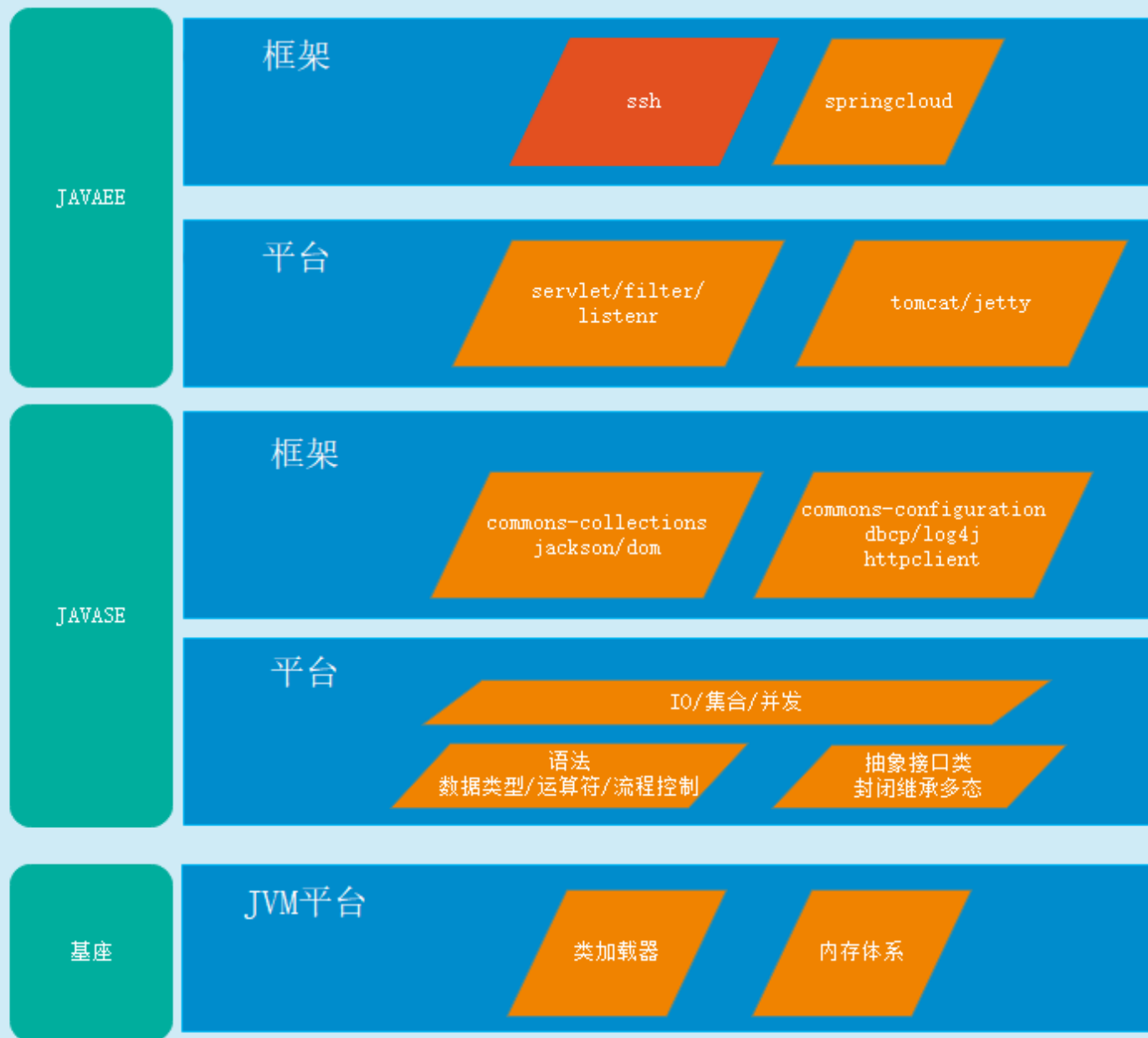


课程内容

- **Spring在JAVAEE体系中的位置**
- IOC架构介绍
- IOC源码分析
- IOC插件开发（附插件源码）
- 和AOP的结合点



JAVAEE体系



课程内容

- Spring在JAVAEET体系中的位置
- 架构
 - UML介绍
 - 源码分析
- 启动/使用流程
 - UML介绍
 - 源码分析
- IOC插件开发（附插件源码）
- 和AOP的结合点



课程内容

- Spring在JAVAEET体系中的位置
- 架构
 - UML介绍
 - 源码分析
- 启动/使用流程
 - UML介绍
 - 核心流程源码分析(BEAN生命周期)
- IOC插件开发（附插件源码）
- 和AOP的结合点



【容器级-BeanFactoryPostProcessor】 BeanFactoryPostProcessor调用postProcessBeanFactory方法

第一个BEAN

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessBeforeInstantiation】

【构造器】调用Person的构造器实例化

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessPropertyValues】

【Bean自身-注入属性】注入属性phone

【Bean自身-BeanNameAware接口】调用BeanNameAware.setBeanName()

【Bean自身-BeanFactoryAware接口】调用BeanFactoryAware.setBeanFactory()

【容器级-BeanPostProcessor.postProcessBeforeInitialization】对属性进行更改

【Bean自身-InitializingBean接口】调用InitializingBean.afterPropertiesSet()

【容器级-BeanPostProcessor.postProcessAfterInitialization】对属性进行更改

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessAfterInitialization】

第二个BEAN

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessBeforeInstantiation】

【构造器】调用Person的构造器实例化

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessPropertyValues】

【Bean自身-注入属性】注入属性phone

【Bean自身-BeanNameAware接口】调用BeanNameAware.setBeanName()

【Bean自身-BeanFactoryAware接口】调用BeanFactoryAware.setBeanFactory()

【容器级-BeanPostProcessor.postProcessBeforeInitialization】对属性进行更改

【Bean自身-InitializingBean接口】调用InitializingBean.afterPropertiesSet()

【容器级-BeanPostProcessor.postProcessAfterInitialization】对属性进行更改

【容器级-InstantiationAwareBeanPostProcessorAdapter.postProcessAfterInitialization】

.....

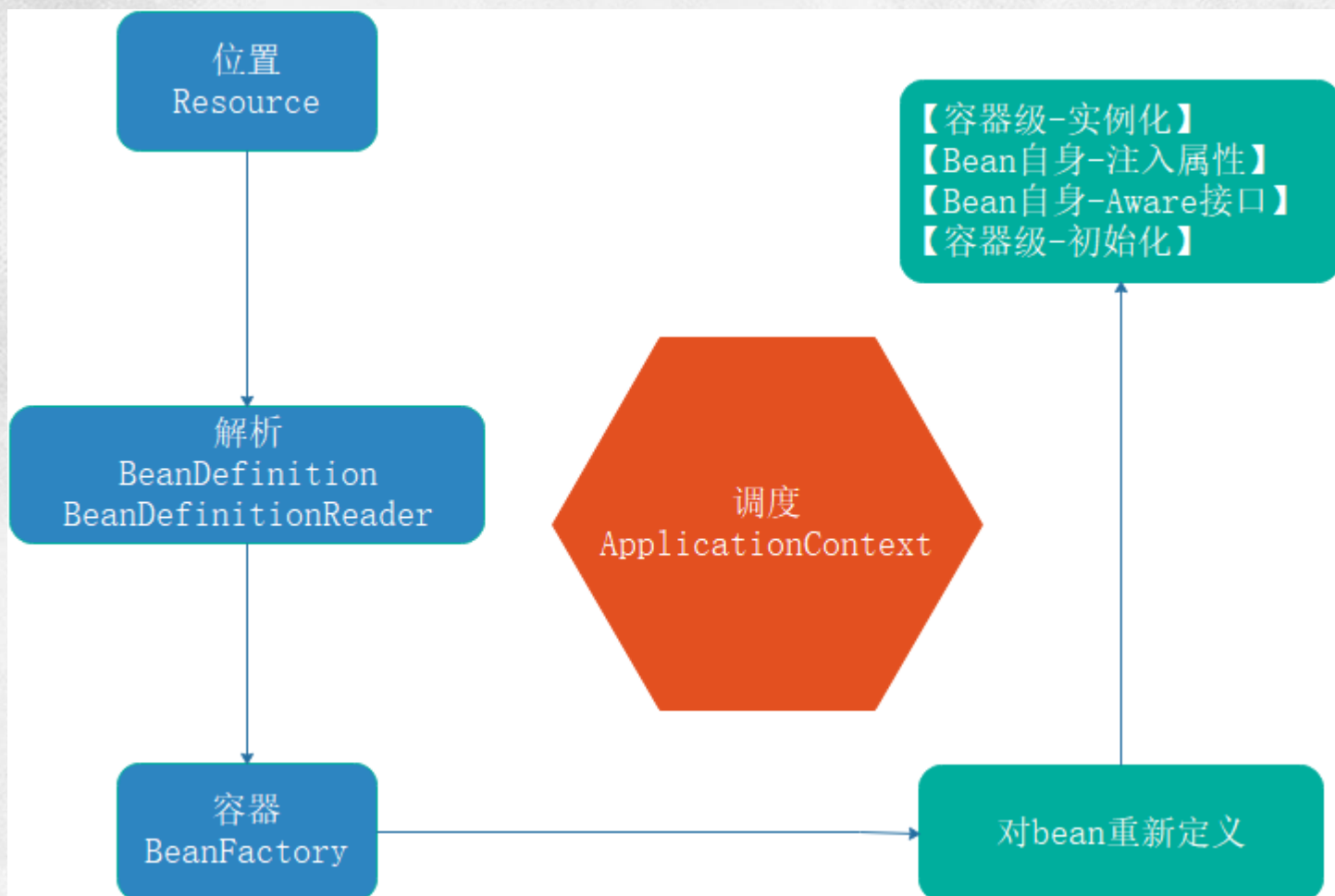


课程内容

- Spring在JAVAEE体系中的位置
- 架构
 - UML介绍
 - 源码分析
- 启动/使用流程
 - UML介绍
 - 核心流程源码分析(BEAN生命周期)
- IOC插件
 - 需求介绍
 - 开发（附插件源码）
- 课堂总结

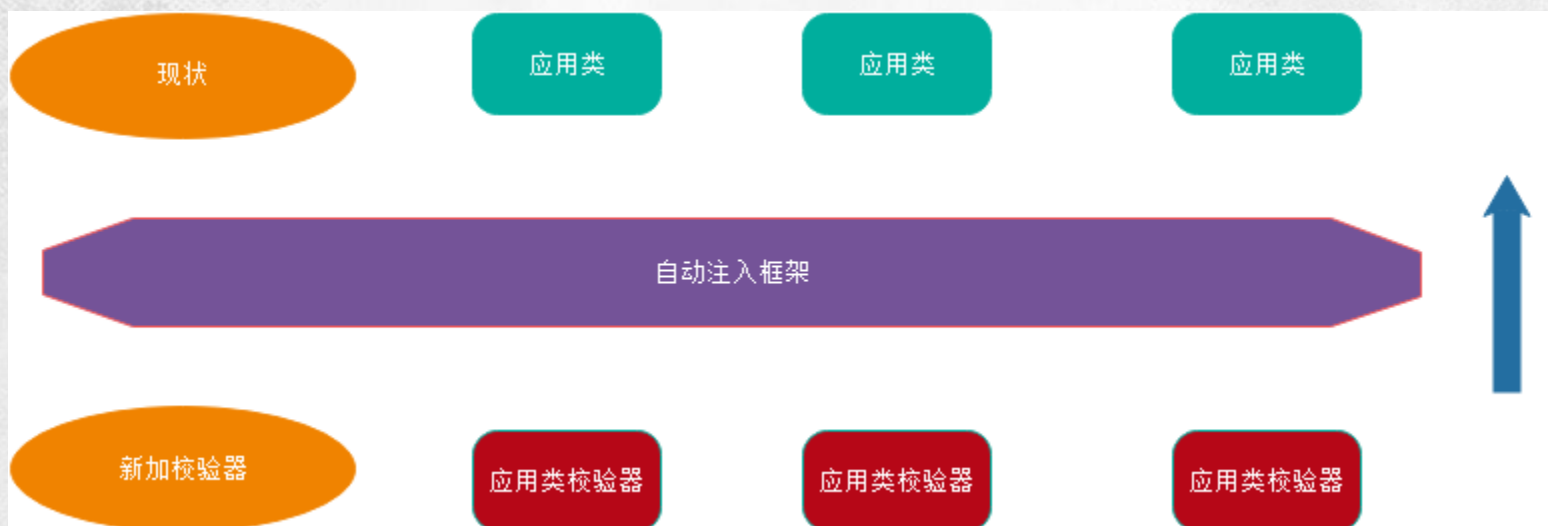


插件开发



插件需求两种场景

场景一、对旧工程改造，需要注入一些变量



插件需求两种场景

场景二、处理器的开发

