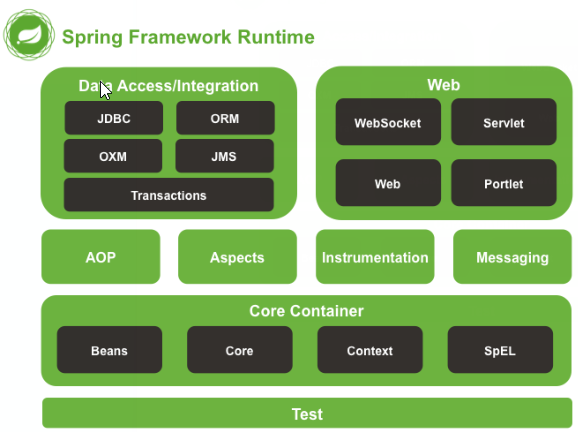
·Spring简介

1. 理念: 使现有技术更加实用。本身是一个大杂烩，整合现有技术
   * 1. 我们现阶段主要学习spring framework以后进入公司可能会用到的spring security
     2. 与spring security相同的安全方面还有shiro
2. Spring优点
   * 1. 轻量级框架，轻量级框架有一个好处就是当我们想要替换他时无需更改代码
     2. 提供ioc容器---控制反转（学习重点）
     3. Aop面向切面编程
     4. 对事物的支持
3. **spring是什么？**

Spring是分层的 Java SE/EE应用 full-stack 轻量级开源框架，以 IoC（Inverse Of Control：

反转控制）和 AOP（Aspect Oriented Programming：面向切面编程）为内核，提供了展现层 SpringMVC 和持久层 Spring JDBC 以及业务层事务管理等众多的企业级应用技术，还能整合开源世界众多著名的第三方框架和类库，逐渐成为使用最多的Java EE 企业应用开源框架。

1. Spring 框架



在这里我们将重点学习Core Container 和aop & aspects，前者指的就是ioc(控制反转)，后者指的时aop(面向切面)

1. 什么是 IOC (Inverse of Control)

通过案例:(01-SpringIoc)可以得出(案例可以看成是ioc 的源码或实现原理)

* + 1. 通俗的说IOC 就是 解耦 ，层与层之间进行了分离的功能。让我们更多的面对业务逻辑，而不再关注对象的创建。
    2. 对象由原来的程序创建对象，变为程序接收对象
    3. 实现了service 和 dao 的解耦工作，service 和 dao 层实现分离，没有直接的依赖。如果daof发生改变，service 代码依然可以使用而不改变自身。
    4. Ioc 是一种编程思想。由主动编程变为被动接收。
    5. Ioc d的实现是 通过容器实现的，Ioc容器 --- BeanFactory

控制反转:

控制:指谁来控制对象的创建；传统的应用程序由程序本身来控制对象的创建，而使用spring 后由spring来创建对象。

反转:是权限的转移，以前是由程序来主动创建对象，现在是被动的接收，这是一种权限的转移。

控制反转还有一个别名 --------依赖注入 （dependency injection） : 通过set方法注入程序依赖的 属性 或者 对象；

总结: Spring Ioc 就是： 对象由Spring 创建 管理 装配(属性装配)

以上为重点

------------------------------------------------------------------------------------

1. 使用 Ioc 来创建对象有三种不同的方法，而每种方法中又有不同的方法(共有六种)
   * 1. 使用无参构造来创建
        1. Beam.xml

<bean id=” user ” class”cn.gsdx.domain.User” >

<property name=”name” value=”张三”> </property>

</bean>

此时一旦我们使用 ApplicationContext 实例化的对象 ac 后，调用ac.getBean() ; 会自动调用无参构造方法为我们实例化对象。

然后通过<property>进行属性赋值。

* + 1. 使用有参构造来创建
       1. Beam.xml

<bean id=” user ” class”cn.gsdx.domain.User” >

<constructor-arg index = “0” value=”李四”></constructor-arg>

</bean>

<!-- index 代表的是有参构造方法的参数下标 -->

* + - 1. Beam.xml

<bean id=” user ” class”cn.gsdx.domain.User” >

<constructor-arg name = “name” value=”李四”></constructor-arg>

</bean>

<!-- name 代表的是有参构造方法的参数名称 -->

* + - 1. Beam.xml

<bean id=” user ” class”cn.gsdx.domain.User” >

<constructor-arg type= “String” value=”李四”></constructor-arg>

</bean>

<!-- String 代表的是有参构造方法的参数类型 -->

* + 1. 通过工厂来创建
       1. 静态工厂创建
       2. 动态工厂来创建