

1、IOC概念

IOC (Inversion Of Control)：控制反转。是一种思想，或者说一种概念，一种理论

主要含义：将对象的创建、赋值、管理等工作交给代码以外的容器实现，通俗的讲就是项目中的Java对象的创建由外部的资源完成，不需要在代码中实现

控制的含义：代表对象的创建、对象属性的赋值以及对象之间关系的管理

反转的含义：讲原来开发人员创建对象、管理对象的权限移交给代码之外的容器实现。该容器代替开发人员创建对象、管理对象，给属性赋值。

正转的含义：就是开发人员在项目代码中进行编写new对象的语句来创建对象，然后有开发人员管理对象

容器：容器可以是一个软件（Tomcat），也可以是一个框架（Spring）

IOC的作用：在改动少量的代码情况下，实现功能的多样化。简单来讲就是解耦合

Java中创建对象的方式：

- 1、构造方法，new Student()
- 2、反射
- 3、序列化
- 4、克隆
- 5、ioc：容器创建对象
- 6、动态代理

IOC的体现：

--在之前servlet学习中，我们只需要创建一个Servlet实现类集成HttpServlet，然后配置这个类的完整的类名到web.xml文件中-注册这个实现类，web容器就能自动帮助我们创建实现类的对象，这是Tomcat创建对象，不需要我们在项目代码中编写创建对象的代码。Tomcat这个web容器存放了Servlet对象，Listener对象（监听器），Filter对象（过滤器）

2、实现IOC的技术（方式）

--控制反转，是一种框架实现的概念之一，理论。实现控制反转的方式有很多，在Spring中IOC实现的技术叫做：

Dependency Injection：翻译过来叫 "依赖注入"，简称DI

使用依赖注入，我们只需要在程序中提供使用的对象名称就可以了。而对象如何在容器中被创建，被赋值，被查找使用都由容器实现，不需要我们管理。

--学习JavaSE阶段的反射机制时，我们知道框架都是基于反射机制的，所以Spring的DI方式底层创建对象，就是使用的反射机制

Spring是一个容器，管理对象，给对象属性赋值，底层通过反射创建对象。