

IOC理解

IOC，即控制反转，不是技术，而是一种设计思想，要理解IOC，就要与传统创建对象的方法作对比
理解OC的关键是要明确“谁控制谁，控制什么，为何是反转”

- 谁控制谁，控制什么：IOC容器控制对象，控制了外部资源获取（不只是对象还包括了文件）
- 为何是反转，哪些方面反转了。

有反转就有正转，传统应用程序是由我们自己在对象中主动去直接获取对象，也就是正转
而反转则是由容器来帮忙创建及注入依赖对象，
为何是反转：因为由容器帮我们查找及注入依赖对象，对象只是被动的接受依赖对象，所以是反转
哪些方面反转了：依赖对象的获取被反转了

- IOC能做什么

- 传统创建对象的方法的弊端：在类内部主动创建依赖对象，从而导致类与类之间高耦合，难于测试
- 有了IOC容器后，把创建和查找依赖对象的控制权交给了容器，由容器进行注入组合对象，所以对象与对象之间是松散耦合，如此一来方便测试，也利于功能复用，更重要的是使得程序的整个体系结构变得非常灵活
-