

Spring控制反转(IoC)的理解 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Spring_E6_8E_A7_E5_c104_144538.htm Spring框架的核心就是控制反

转(Inversion of Control)和依赖注入(Dependency Injection)，通过这两方面来实现松耦合。使用IoC，对象是被动的接受依赖类，而不是自己主动的去找。容器在实例化的时候主动将它的依赖类注入给它。可以这样理解：控制反转将类的主动权转移到接口上，依赖注入通过xml配置文件在类实例化时将其依赖类注入。通过下面的实例来逐步的理解：首先假设有一个需求，类Business需要调用类Dependency的方法f()，按照日常的做法，得到下面的代码：

```
/**类Dependency** public class Dependency { public void f() {}. } /**类Business** public class Business { Dependency d. public Business() { d = new Dependency(). } public void doSth() { d.f(). } }
```

对上述实现做出如下修改：首先，将Business里的Dependency实例的获得该为setter方式，其次，将Dependency类改为某个接口的实现。故可以得到下面新的代码：

```
/**接口IDependency** public interface IDependency { void f(). } /**类Dependency** public class Dependency { public void f() {}. } /**类Business** public class Business { IDependency d. public Business() {} public void doSth() { d.f(). } public void setDependency(IDependency d) { this.d = d. } }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com