**Problem**

class Dao {

private mySQLSession;

Dao() { //如何解耦？成百上千的依赖关系能否集中管理？

mySQLSession = new mySQLSession();

}

}

**Inversion of Control / Dependency Injection**

·对象间的依赖关系，由IoC容器集中管理被依赖对象的生命周期。

·控制被反转之后，获得依赖对象的过程由自身管理变为由IoC容器动态注入。

**Dependency inversion principle**

·上层模块不应该依赖底层模块，它们都应该依赖于抽象。

·抽象不应该依赖于细节，细节应该依赖于抽象。

注入方式:

·constructor

·setter

IoC容器：一个bean对象池，根据配置或注解获取类的信息。

注解：

1. BeanFactory
2. SpringMVC