**Spring核心思想**

**Spring IOC**

**定义：**

把创建对象的工作交给容器

控制没有反转时

//B ,C implement A

A a = new B();

当我们想换成C的实现时，那我们就得

把所有出现A a = new B()的地方改成A a = new C()，这样及其不方便

当我们有了IOC

我们通过配置XML来管理类与类的关系，容器根据XML帮我们创建对象

一开始时是：

<bean name=”a” class=”..B.class”/>

当我们需要修改时，就只需要改为：

<bean name=”a” class=”..C.class”/>

这样修改就非常方便

**Spring依赖注入**

**定义：**

把类与类的关系交给容器来管理（比如a 类依赖b类这种情形）

**好处：**

容易维护

当没有依赖注入时：

//B ,C implement A

// D类依赖A类

Class D{

Private A a;

}

我们需要使用D类时

A a = new B();

D d = new D();

d.setA(a)

当我们需要换一下a的实现方式，完了

我们需要到处把

A a = new B();

D d = new D();

d.setA(a)

改成：

A a = new C();

D d = new D();

d.setA(a)

但我们有了,依赖注入时

一开始时这样：

<bean id="a " class="..B .class" />

<bean id="d" class="..D.class" >

<property name = "a" ref = "a"></property>

</bean>

但修改时：

<bean id="a " class="..C .class" />

<bean id="d" class="..D.class" >

<property name = "a" ref = "a"></property>

</bean>

修改简单

**Spring切面编程**

**定义：**把函数当成西瓜，刀就是切面（定义着与业务无关的方法），切点就是下刀的位置，当我们把到刀切到西瓜的最上边是（函数在执行前），切面定义的方法就是前置方法，当我们把到刀切到西瓜的最尾部（函数执行完），切面定义的方法就是后置方法，around方法就是前后都切，afterReturning就是在函数返回前切。