```
控制反转(IoC、Inversion of Control),是一个概念,是一种思想。指将传统上由
码直接操控的对象调用权变给容器,通过容器来实现对象的衰配和管理。
转就是对对象控制权的转移,从程序代码本身反转到了外部容器。Spring是容器

    1.仇敵抢技, Dependency Lookup. DL. 答為提供问调接口和上下文产境给相伴, 程序代码则需要提出技方式。2.依赖往入,Dependency Injection. Dl. 程序代码不衡定位查询。这些工作由容器自行完成
    默认bean的创建方式。调用bean的无参数的构造方法

                                                                                                                                                                                         bean的装配: 即bean对象的创建
                                                                                                                                                                                                                                           | monf(G); | monf(G);
                                                                                                                                                                                bean的生命始末
                                                                                                                                                                            Spring守器配置文件, name:是构造方法的形参名

<br/>
<br/>
chean id="myStudent" class="com.beans.Student"><br/>
<onstructor-arg name="myname" value="是三"><br/>
<onstructor-arg name="myXuexiao" ref="mySchoot" ><br/>
<br/>
<br/>
cbean>
                                                                                                                                    | 构造注入 |
                                                                                                                  简单类型属性联值 @Value 放在属性定义或者set方法上面
@Component(Peanfilld*)
public class Student {
    @Value*Zhangsan*)
    private String name;
                                                                                                                            接类型注入(byType),符合条件的对象只有一个
《Autowired 即《Resource 故在属性证义和set方法上面。
或值免疫和出异常。《Autowired 即《Resource 故在属性证义和set方法上面。
或值免疫和出异常。《Autowired 即以Spring能变投资的。《Resource》是由中的
《Gaussing》(《Resource》是由中的
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(《Resource》),是有一个
《Resource》(Resource》),是有一个
《Resource》(Resource》),是有一个
《Resource》(Resource》),是有一个
《Resource》(Resource》),是有一个
《Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》),是一个
《Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resource》(Resour
                                                                         使用注解的DI
                                                                                                                               @PreDestroy
public void tearDown() {
//对象销级之前要执行的方法
}
                                                                         Discontainment (GGIB) 代明的生成原用是生或目标表的子类。而于类及增强过的,这个子类对象就是代理对象。
所以,我们 GGIB 生成态层代理。要是目标表态观影略被需求,即不能是 final 的 类。
Finance rehance rewards - rew Ethance();
enhance residence - rew Ethance();
enhance residence(); 使从Mandediscreptor模目的对象);
enhancer.create(); //创程代理对象
                                                                                     enhancer.create(),的错误代理对象

AopR念,AOP(Aspect Orient Programming),则何初则编程。从一种思想
到面,给程序动态带到的功能。日志,事务常作为如而使用。

AOP (Aspect Orient Programming) ,则何初则编程。从一种思想
到面,给程序动态带到的功能。日志,事务常作为如而使用。

AOP 该反及采用动态代理发现的,两种代理,JDK 的动态代理
                                                                                                                   Spring学器処置文件、
<-- 注册目标对象~

- 注册目标对象~

- 注册目标对象~

- 生光的是小de* dassa* com.bjpowernode.SomeService!mpt*><bbean>

- 生光的過度者。

- 中島 Massa* com.bjpowernode.MyAspect*><bbean>

- 上配温索ppiliplia可能性

- app aspect|-autoproxy />
                                                                                                         Jaca代码。
public class MyAspect {
//增置基地的增强功能
public void benominato}(
System.out println("切面夹中的装置地面");
                                                                                                                                                                                                                                                                                                                  事务的特点。
度于柱。一致性,隔离性。持入性
Spring中的事务是定义在业务方法上的。业务方法中调用Dao执行数据库操作
事务管理器。FlatformTransactionManager 按订对象。定义事务的操作行为。规义、封准、不同数据库访付技术。事务规义、封准的实现由实现实完成。开发人员只需关心使用数据库
访问技术的实现实即可。
Spring4
                                                                                                                                                                                                                                                                                                                                           隔离规则。
DEFAULT: 采用 DB 款认前等务隔离微则。MyGgl 的款认为 REPEATABLE READ;
OGAB於以为 READ_COMMITTED.
READ_LINCOMMITTED. 该老规定,未解社经师发问题。
READ_LONMITTED. 该老规定,未解社经师发问题。
REPEATABLE READ,可能发验,解决批准。不可能发验与对话。
REPEATABLE READ,可能发验,解决批准。不可意发验,存在效验
SERNALZABLE,每行化,不存在非发问题

传播行为。
                                                                                                                                           事务定义接口TransactionDefinition 中定义了事务檔述相关的三类常量:
事务隔离级例、事务传播行为。事务款从超时时限。
Spring事务的控制就是合理的设置这个三个特性的值
                                                                                                                                                                                                                                                                                                                                            Spring事务的默认操作是。发生运行时异常时回滚,幸运行时异常提交(可以设置回滚)
                                                                                      事务是通过AOP的环核通知实现的,AOP思想的实现是动态代理、Service类有实现接口默认是jdk动态代理。
无接口使用Gglb动态代理,有接口的Service也可以设置使用Gglb动态代理
                                                                                                                                                                                    Spring框架中事务的设置(注解和AspectJ)
                                                                                                                                                                                                                                                                                                                《Nc.dalucius》
《Nc.dalucius》
《Ho.dalucius》
《Ho.dalucius
                                                                                                                                      整合需要的配置文件

    事業以后的核素定父的Spring的等界调整理
    主要科器等解决力法、Spring的等等共需的量一次、故例SprindContext作用域。
    直察器可以手工程性、也可以使用框架提供的ContextLoaderListener
    企業等差別をからContext、存在人気をwhetContext用用域
    web.xmi注册服用器・
    contextLoaderListenerとの対象が表現を対象がある。
    contextLoaderListenerとの対象が表現を対象がある。
    contextLoaderListenerを使用力式
    ContextLoaderListener使用力式
    (ContextLoaderListener使用力式
    (ContextLoaderListener使用力式
    (ContextLoaderListener使用力式
    (ContextLoaderListener使用力式
    (ContextLoaderListener性別分表)
    (ContextLoaderListener性別分表)
    (ContextLoaderListener性別分表)

                                                                                                                                                                                                                                                                    //使用框架提供的工具类获取容器对象
WebApplicationContext ac = null;
ac =WebApplicationContextUtils.getRequir
```