用于注入数据的注解

例子：

accountDao1

```
@Repository("accountDao1")
public class AccountDaoImpl implements IAccountDao
   {
   public  void saveAccount(){
      System.out.println("保存了账户1111111111111");
   }
}
```

accountDao2

```
@Repository("accountDao1")
public class AccountDaoImpl implements IAccountDao
   {
   public  void saveAccount(){
      System.out.println("保存了账户22222222");
   }
}
```

一、Autowired

1、作用：(先根据bytype匹配；若有多个匹配类型，则按照byname)

自动按照类型注入。只要容器中有唯一的一个bean对象类型和要注入的变量类型匹配，就可以注入成功；如果ioc容器中没有任何bean的类型和要注入的变量类型匹配，则报错。

如果Ioc容器中有多个类型匹配时：根据变量名进行匹配，有则匹配成功；没有则报错

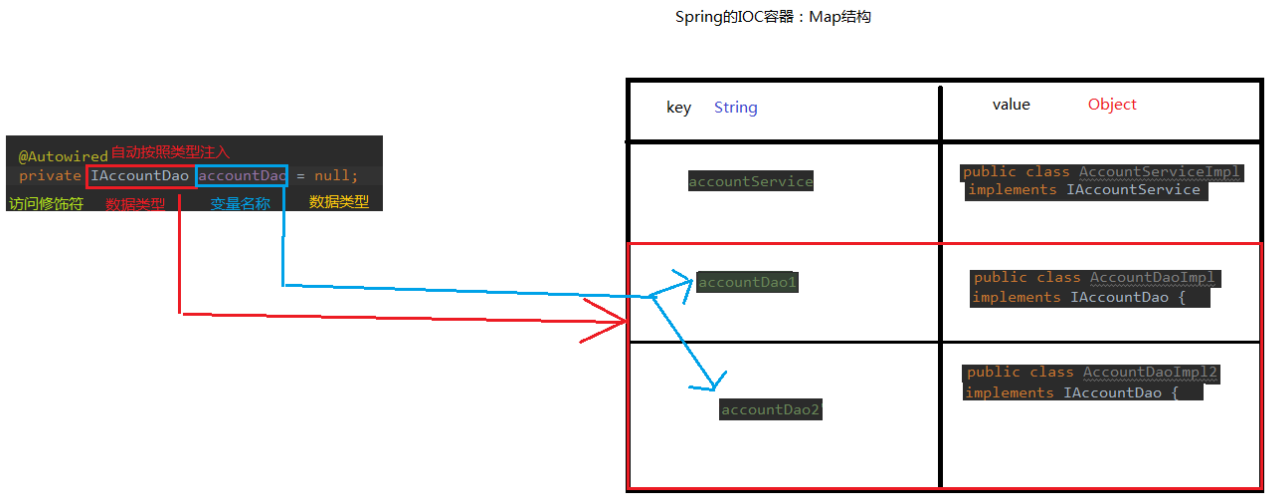
2、出现位置：

可以是变量上，也可以是方法上

3、 细节：

在使用注解注入时，set方法就不是必须的了。

4、图解



5、实例

```
@Autowired
   private IAccountDao accountDao = null;
```

二、 Qualifier:

1、 作用：

在按照类中注入的基础之上再按照名称注入。它在**给类成员注入时不能单独使用**。但是在给方法参数注入时可以（稍后我们讲）

2、 属性：

value：用于指定注入bean的id。

```
public class AccountServiceImpl implements IAccountService {
@Autowired
 @Qualifier("accountDao1")
   private IAccountDao accountDao = null;
   }
```

3、注意：

给类成员注解时要配合Autowired使用

三、 Resource

1、 作用：

直接按照bean的id注入。它可以独立使用

2、 属性：

name：用于指定bean的id。

注意：

以上三个注入都只能注入其他bean类型的数据，而基本类型和String类型无法使用上述注解实现。

另外，集合类型的注入只能通过XML来实现。

四、 Value

1、作用：

将配置文件的属性读出来，用于注入**基本类型和String类型**的数据

2、 属性：

value：用于指定数据的值。

它可以使用spring中SpEL(也就是spring的el表达式）SpEL的写法：${表达式}

```
@Value("${jdbc.driver}")
private String driver;
```

然后再