## 1 @SuppressWarning(“deprecation”)使用

使用@Deprecation的方法若再被调用，则为了不出现警告可以在该方法上面使用@SuppressWarning(“deprecation”)忽略警告

## 2 注解分类

### 按照运行机制分：

源码注解：只在源码中存在，编译成.class文件就不存在

编译时注解：在源码和.class文件中存在，如@SuppressWarning，@Deprecation

运行时注解：在运行阶段还起作用，甚至会影响运行逻辑的注解，如@Autowired

### 按照来源分：

来自JDK的注解

来自第三方的注解

我们自己定义的注解

元注解：给注解当注解

## 3 自定义注解的语法要求

@interface:表明是注解类

成员只能是无参无异常的

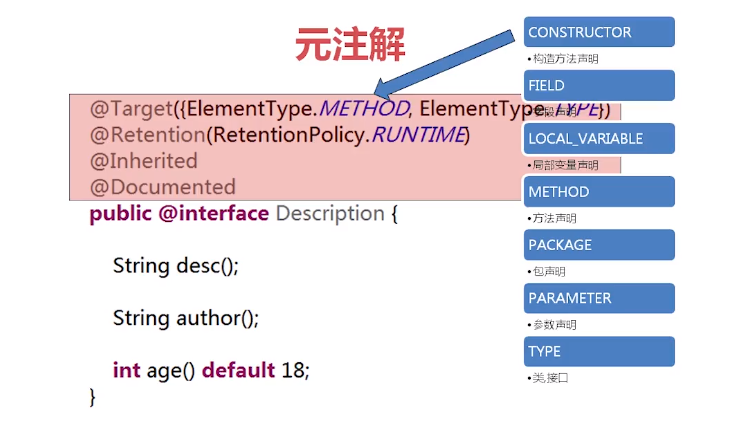
成员类型受限，合法的类型包括原始类型及String,Class,Annotation,Enumeration

如果注解只有一个成员，则成员名必须取名为value()，在使用时可以忽略成员名和赋值号（=）

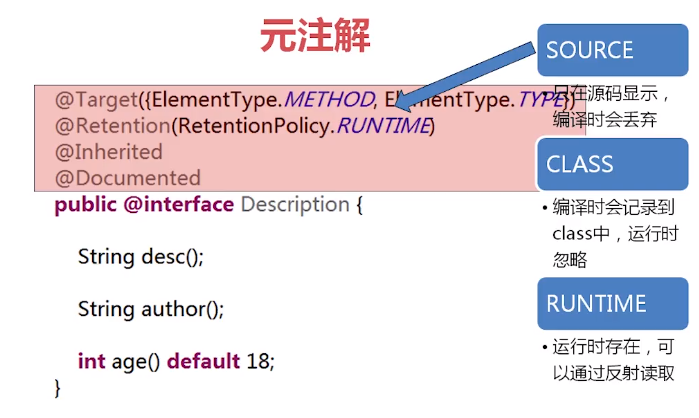
没有成员的注解类称为标识注解

元注解：

@Target:作用域



@Retention:生命周期



@Inherited:标识类注解，允许子类继承，不能用在接口上，而且只能继承类上面的注解，不能继承方法的。

@Documented:生成javadoc时会包含注解