# JAVA注解

## 注解的分类

### 按照运行机制分

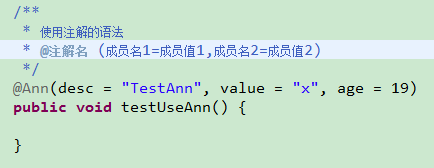
1. 源 码 注解 只存在在源码中，编译为class文件就没有了
2. 编译时注解 在源码和class文件都都存在
3. 运行时注解 在运行阶段还起作用，甚至会影响运行逻辑的注解

## 自定义注解

### 语法要求



### 自定义注解的使用



### 注解的继承@Inherited

该元注解只能 注解在类上，并且不支持接口的注解，只支持继承

并且 类只能继承 类的类注解 不能继承类的方法注解

## 注解的解析 需要通过反射来进行解析

是否含有注解 c.isAnnotation(Ann.**class**);

获取这个注解 c.getAnnotation(Ann.**class**);

# JAVA反射

## Class类

### 类是对象,类是java.lang.Class类的实类对象

### 表示Class对象

#### Class c1 =Foo.class 任何一个类都有一个隐含的成员静态成员变量

#### Class c2= foo.getClass()类的对象通过getClass方法

c1 ==c2

#### Class c3=Class.*forName*(“com.xx.Foo”);

c2==c3

#### 通过类的类型来创建对象

c1.newInstance();

### 类的动态加载 Class.*forName*(‘com.xx.xx’) 在运行时刻加载

### 获取类的方法信息

获取所有方法信息

Method[] methods = c1.getMethods();

获取自身声明的所有方法，不包括继承

Method[] methods2 = c1.getDeclaredMethods();

获取方法的返回类型的类类型

methods[i].getReturnType().getName();

获取方法的名称

methods[i].getName();

获取方法的所有参数的类

Class[] parameterTypes = methods2[i].getParameterTypes();

### 获取类的成员变量

获取所有的public 成员变量

Field[] fields = c1.getFields();

获取所有自己申明的成员变量

Field[] declaredFields = c1.getDeclaredFields();

获取方法的名称

field.getName();

获取方法的返回类型的名称

field.getType().getName();

### 获取类的构造函数

获取类的所有public构造函数

Constructor[] constructors = c1.getConstructors();

获取类的所有构造函数

Constructor[] declaredC = c1.getDeclaredConstructors();

获取构造函数的的参数

constructor.getParameterTypes();