

传奇

永久域名 <http://legendswallow.javaeye.com>**legendswallow**

人生重要的不是所站的位置，而是所朝的方向
2010-04-01 通过网页

[>>更多闲聊](#)

浏览: 4404 次

性别:

来自: 北京



我现在离线

[详细资料](#)[留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)

[waiting](#)[tinygone](#)[a890323](#)[bitzhl](#)

博客分类

- [全部博客 \(54\)](#)
- [JS \(6\)](#)
- [Java \(20\)](#)
- [DB \(8\)](#)
- [C \(1\)](#)
- [Python \(1\)](#)
- [Ophone \(0\)](#)
- [Web \(1\)](#)
- [算法 \(1\)](#)
- [反思 \(5\)](#)

[jQuery操作表格，设置第一列行号](#) | [velocity 判断对象是否为空](#)

2010-03-09

[java注解](#)

文章分类:[Java编程](#)

Annotation是Java5、6只后的新特征（中文称之为注解），并且越来越多的得到了应用，比如Spring、Hibernate3、Struts2、iBatis3、JPA、JUnit等等都得到了广泛应用，通过使用注解，代码的灵活性大大提高。

这些都是使用别人定义的注解，一般注解的使用都是在一些基础框架或者类库中来定义的，因此很少见过有人自己去写一个注解出来并使用在程序中，因此注解的使用常常给人感觉很神秘，这就为你揭开注解的面纱。

注解的神秘之处在于：通过类似注释的方式，可以控制程序的一些行为，运行时的状态，可以为成员赋值，做配置信息等等，与常规编码思维大相径庭。

只用别人定义好的注解是搞不懂这些问题的，要想真正知道注解内部的秘密，要自己定义注解，然后在程序中获取注解信息，拿到注解信息后，就可以为我所用了。

下面我简单演示下三类注解的用法：类注解、方法注解、字段（也称之为域）注解的定义与适用，并看看如何获取注解的信息。

一、定义注解

```
package lavasoft.anntest;
```

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
```

```
/**
 * 类注解
 *
 * @author leizhimin 2009-12-18 14:15:46
 */
```

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface MyAnnotation4Class {
    public String msg();
}
```

```
package lavasoft.anntest;
```

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
```

```
/**
 * 方法注解
 *
 * @author leizhimin 2009-12-18 14:16:05
 */
```

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface MyAnnotation4Method {
```

我的留言板 [>>更多留言](#)

- 抄袭别人的，自己感觉也不错！
你好

-- by [legendswallow](#)

- 很喜欢你的这句话，学习了！
作为系统中的个体，最重要的动作不是“往上爬”，而是“专 ...

-- by [vera_sq](#)

其他分类

- [我的收藏](#) (79)
- [我的论坛主题贴](#) (1)
- [我的所有论坛贴](#) (2)
- [我的精华良好贴](#) (0)

最近加入圈子

- [JBPM @net](#)
- [JSF](#)
- [函数式编程之道](#)
- [RedSaga满江红](#)
- [权限管理](#)

存档

- [2010-04](#) (5)
- [2010-03](#) (18)
- [2010-02](#) (11)
- [更多存档...](#)

最新评论

- [java注解](#)
很好，终于对神秘的注解有了一点了解了

-- by [wen.owen](#)

- [学习java不得不学的技术](#)
没任何权威性合理性，只是把日常web开发的技术罗列了一遍

-- by [linliangyi2007](#)

- [学习java不得不学的技术](#)
现在都要会啊。竞争激烈的说！

-- by [topcode](#)

- [学习java不得不学的技术](#)
我觉得还是精通一方面就行了~

-- by [condeywadl](#)

- [学习java不得不学的技术](#)
艺多不压身啊 不过，需要精通一两门

```
public String msg1();

public String msg2();
}
```

```
package lavasoft.anntest;
```

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
```

```
/**
 * 字段注解
 *
 * @author leizhimin 2009-12-18 15:23:12
 */
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
public @interface MyAnnotation4Field {

    public String commont();

    public boolean request();
}
```

二、写一个类，用上这些注解

```
package lavasoft.anntest;
```

```
/**
 * 一个普通的Java类
 */
@MyAnnotation4Class(msg = "测试类注解信息")
class TestClass {

    @MyAnnotation4Field(commont = "成员变量的注解信息", request = true)
    private String testfield;

    @MyAnnotation4Method(msg1 = "测试方法注解信息1", msg2 = "测试方法注解信息2")
    public void testMethod() {
        System.out.println("Hello World! ");
    }
}
```

三、测试注解

为了使用注解，需要通过反射获取注解的对象。通过注解对象来操作注解信息。

```
package lavasoft.anntest;
```

```
import java.lang.reflect.Field;
import java.lang.reflect.Method;
```

```
/**
 * 测试类
 *
 * @author leizhimin 2009-12-18 14:13:02
 */
public class TestOptAnnotation {

    public static void main(String[] args) throws NoSuchMethodException, NoSuchFieldException {
        TestClass t = new TestClass();
        System.out.println("-----MyAnnotation4Class注解信息-----");
        MyAnnotation4Class an4clazz = t.getClass().getAnnotation(MyAnnotation4Class.class);
```

-- by [yangjueshe](#)

评论排行榜

- [学习java不得不学的技术](#)
- [java注解](#)
- [关于龙芯的一片文章，写的极好](#)
- [实例恢复](#)
- [sys和system用户的区别](#)



[\[什么是RSS?\]](#)

```
System.out.println(an4clazz.msg());
System.out.println("-----MyAnnotation4Method注解信息-----");
Method method = t.getClass().getMethod("testMethod", new Class[0]);
MyAnnotation4Method an4method = method.getAnnotation(MyAnnotation4Method.class);
System.out.println(an4method.msg1());
System.out.println(an4method.msg2());
System.out.println("-----MyAnnotation4Field注解信息-----");
Field field = t.getClass().getDeclaredField("testfield");
MyAnnotation4Field an4field = field.getAnnotation(MyAnnotation4Field.class);
System.out.println(an4field.common());
System.out.println(an4field.request());

    }
}
```

运行结果：

-----MyAnnotation4Class注解信息-----

测试类注解信息

-----MyAnnotation4Method注解信息-----

测试方法注解信息1

测试方法注解信息2

-----MyAnnotation4Field注解信息-----

成员变量的注解信息

true

Process finished with exit code 0

四、总结

看完上面的测试过程，Annotation已经不再神秘了，这里总结下自定义注解的使用方法：

- 1、自定义注解，注意注解的时空范围，简单说就是注解针对的目标（类、方法、字段），以及注解的时效（运行时、或者源码中有效）。
- 2、要获取注解的信息，必须通过Java的反射技术来获取Annotation对象，因为你除此之外没有别的获取注解对象的方法。
- 3、获取了注解对象，就可以调用注解的方法来获取相对应的值了。为基础框架所用。
- 4、当然，注解也可以没有定义成员，这样注解就成了一个标记符号了。

理解了这些，注解就不再神秘了。

[圆三繙鑽勛亾壘 鍍狃負鐵下續杓炔螻](#)

錫虫敵鏑伴漆罇棋施鏑岫堞滌 惡錄嬪愜蹇 緬或笱漢諱翰鏑出數琛岍姪閭估慘甯岫湼鏑
www.google.cn

[jQuery操作表格，设置第一列行号](#) | [velocity 判断对象是否为空](#)

19:13 | [浏览 \(519\)](#) | [评论 \(1\)](#) | 分类: [Java](#) | [相关推荐](#)

评论

1 楼 [wen.owen](#) 2010-04-09 [引用](#)

很好，终于对神秘的注解有一点点了解了👍

发表评论

表情图标

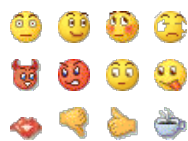


字体颜色:

字体大小:

对齐:

提示：选择您需要装饰的文字，按上列按钮即可添加上相应的标签



您还没有登录，请[登录](#)后发表评论(快捷键 **Alt+S** / **Ctrl+Enter**)