



Java教程

Java快速入门

面向对象编程

异常处理

反射

注解

使用注解

定义注解

处理注解

泛型

集合

IO

日期与时间

单元测试

正则表达式

加密与安全

多线程

Maven基础

最后更新: 2019/2/12 08:47 / 阅读: 6468777

Java语言使用 `@interface` 语法来定义注解（`Annotation`），它的格式如下：

```
public @interface Report {  
    int type() default 0;  
    String level() default "info";  
    String value() default "";  
}
```

注解的参数类似无参数方法，可以用 `default` 设定一个默认值（强烈推荐）。最常用的参数应当命名为 `value`。

元注解

有一些注解可以修饰其他注解，这些注解就称为元注解（meta annotation）。Java标准库已经定义了一些元注解，我们只需要使用元注解，通常不需要自己去编写元注解。

@Target

最常用的元注解是 `@Target`。使用 `@Target` 可以定义 `Annotation` 能够被应用于源码的哪些位置：

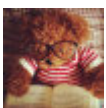
- 类或接口： `ElementType.TYPE` ；
- 字段： `ElementType.FIELD` ；
- 方法： `ElementType.METHOD` ；
- 构造方法： `ElementType.CONSTRUCTOR` ；

[文章](#)[Java教程](#)[Python教程](#)[JavaScript教程](#)[区块链教程](#)[SQL教程](#)[更多](#)[登录](#)[函数式编程](#)[设计模式](#)[Web开发](#)[Spring开发](#)[Spring Boot开发](#)[Spring Cloud开发](#)

关注公众号不定期领红包:



关注微博获取实时动态:



廖雪峰

北京 朝阳区

加关注

```
@Target(ElementType.METHOD)
public @interface Report {
    int type() default 0;
    String level() default "info";
    String value() default "";
}
```

定义注解 `@Report` 可用在方法或字段上, 可以把 `@Target` 注解参数变为数组 `{ ElementType.METHOD, ElementType.FIELD }`:

```
@Target({
    ElementType.METHOD,
    ElementType.FIELD
})
public @interface Report {
    ...
}
```

实际上 `@Target` 定义的 `value` 是 `ElementType[]` 数组, 只有一个元素时, 可以省略数组的写法。

@Retention

另一个重要的元注解 `@Retention` 定义了 `Annotation` 的生命周期:

- 仅编译期: `RetentionPolicy.SOURCE` ;
- 仅class文件: `RetentionPolicy.CLASS` ;
- 运行期: `RetentionPolicy.RUNTIME` 。



香港云服务免费使用 12 个月 + 永久免费
CDN (每月1TB) + 3个月免费VPS/容器/数据库

广告

```
@Retention(RetentionPolicy.RUNTIME)
public @interface Report {
    int type() default 0;
    String level() default "info";
    String value() default "";
}
```

@Repeatable

使用 `@Repeatable` 这个元注解可以定义 `Annotation` 是否可重复。这个注解应用不是特别广泛。

```
@Repeatable(Reports.class)
@Target(ElementType.TYPE)
public @interface Report {
    int type() default 0;
    String level() default "info";
    String value() default "";
}

@Target(ElementType.TYPE)
public @interface Reports {
    Report[] value();
}
```

经过 `@Repeatable` 修饰后，在某个类型声明处，就可以添加多个 `@Report` 注解：



Looking to
publish your
Sustainability
research fast?

Discover
Sustainability



@Inherited

使用 `@Inherited` 定义子类是否可继承父类定义的 `Annotation`。`@Inherited` 仅针对 `@Target(ElementType.TYPE)` 类型的 `annotation` 有效，并且仅针对 `class` 的继承，对 `interface` 的继承无效：

```
@Inherited
@Target(ElementType.TYPE)
public @interface Report {
    int type() default 0;
    String level() default "info";
    String value() default "";
}
```

在使用的時候，如果一个类用到了 `@Report`：

```
@Report(type=1)
public class Person {
}
```

则它的子类默认也定义了该注解：

```
public class Student extends Person {
}
```

如何定义Annotation



```
public @interface Report {  
}
```

第二步，添加参数、默认值：

```
public @interface Report {  
    int type() default 0;  
    String level() default "info";  
    String value() default "";  
}
```

把最常用的参数定义为 `value()`，推荐所有参数都尽量设置默认值。

第三步，用元注解配置注解：

```
@Target(ElementType.TYPE)  
@Retention(RetentionPolicy.RUNTIME)  
public @interface Report {  
    int type() default 0;  
    String level() default "info";  
    String value() default "";  
}
```

其中，必须设置 `@Target` 和 `@Retention`，`@Retention` 一般设置为 `RUNTIME`，因为我们自定义的注解通常要求在运行期读取。一般情况下，不必写 `@Inherited` 和 `@Repeatable`。

小结



必须设置 `@Target` 来指定 `Annotation` 可以应用的范围，

应当设置 `@Retention(RetentionPolicy.RUNTIME)` 便于运行期读取该 `Annotation` 。

读后有收获可以支付宝请作者喝咖啡：



还可以分享给朋友：

[分享到微博](#)[← 上一页](#)[下一页 →](#)



📖 文章

☕ Java教程

🔗 Python教程

📦 JavaScript教程

₿ 区块链教程

🗄️ SQL教程

▼ 更多

🔍 Search

👤 登录



JS Diagram Library

Flowcharts, org charts, and more

GoJS

评论

发表评论

Sign in to make a comment

[📖 文章](#)[☕ Java教程](#)[🔗 Python教程](#)[📦 JavaScript教程](#)[₿ 区块链教程](#)[🗄️ SQL教程](#)[▼ 更多](#)[👤 登录](#)

京ICP备13005808号-11 本网站内容全部为原创，谢绝转载。友情链接: [中华诗词](#) – [阿里云](#) – [SICP](#) – [4closure](#)