·1、课程名称:反射与 Annotation。 JAVA培训先锋 JAVA 应用开发详解 第16章: Annotation — 反射与Annotation MLDN 软件教学研发部

MLDN

魔乐科技

JAVA就业专家

www.MLDNJAVA.cn\_

第(1)页 共(6)页 **可以登录 http://BBS.MLDN.CN进行技术交流**← 第(1)页 共(6)页 可以登录 http://BBS.MLDN.CN进行技术交流→ 魔 乐 科 技 JAVA就业专家 www.MLDNJAVA.cn ·2、本课程预计讲解的知识点。

可以通过Java反射机制取得使用的全部Annotation 可以通过Java反射机制取得指定的Annotation E-MAIL: mldnqa@163.com

本章目标

·3、具体内容。

容。

public <A extends Annotation> A 1 getAnnotation(Class<A> annotationClass) 2 public Annotation[] getAnnotations() 3 public Annotation[] getDeclaredAnnotations() 4 public boolean isAnnotation() public boolean isAnnotationPresent(Class<? 5 extends Annotation> annotationClass)

通过反射取得Annotation

方法

package org.lxh.demo16.reflectannotation;

√ public class SimpleBeanOne {← @SuppressWarnings("unchecked") ← @Deprecated↔ @Override₽ public String toString(){← return "Hello LiXingHua!!!" ;₽ }+/ };₽

E-MAIL: mldnqa@163.com

范围,所以此时通过反射的话只能取得一个。↩ import java.lang.reflect.Method ; public class ReflectDemo01{←

Class  $<?> c = null ; \leftarrow$ for(Annotation a:an){ # 使用 foreach 输出↓ System.out.println(a) ;₽ }₩

}+¹

};₽

public @interface MyDefaultAnnotationReflect{← public String key() default "LXH" ;₽ public String value() default "李兴华" ;↩ }₽ 以上的 Annotation 范围是在运行时依然有效,下面定义一个类使用此 Annotation.↩ package org.lxh.demo16.reflectannotation;€ public class SimpleBeanTwo{← @SuppressWarnings("unchecked") ← @Deprecated↔ @Override₽ @MyDefaultAnnotationReflect(key="MLDN",value="www.mldnjava.cn")↔ public String toString(){+1 return "Hello LiXingHua!!!" ;+1 }₽ };₽ import java.lang.annotation.Annotation; import java.lang.reflect.Method ; public class ReflectDemo02{←

package org.lxh.demo16.reflectannotation;€

import java.lang.annotation.RetentionPolicy; @Retention(value=RetentionPolicy.RUNTIME)

import java.lang.annotation.Retention ;

}+/ }≠¹ ∙4、总结₄

是这种习惯是不是 www.MLDW.cn

·5、预习任务。

下一章内容

E-MAIL: mldnqa@163.com

www.MLDW.cn

www.MLDW.cn

Method toM = c.getMethod("toString") ♯ 找到 toString()方法↩ if(toM.isAnnotationPresent(MyDefaultAnnotationReflect.class)){← #判断是否是指定的 Annotation₽ MyDefaultAnnotationReflect mda = null ;₽ mda = toM.getAnnotation(MyDefaultAnnotationReflect.class) ;// 得到指定的 Annotation↵ String key = mda.key(); # 取出设置的 key+ String value = mda.value(); # 取出设置的 value↵ System.out.println("key = " + key)  $\neq$ System.out.println("value = " + value) ; $\leftarrow$ Annotation 在实际的开发中,不管如何使用,其最终肯定是结合反射机制,也就是说可以通过 Annotation 设置一些内容到一个方法上去,以完成特定的功能。→ 日后再学习 EJB 3.0 中可以发现更多 Annotation 应用。www.mldn.cn 上下载 EJB 3.0 的教程。↩

下面使用反射取得指定的 Annotation,因为现在唯一设置内容的就是 MyDefaultAnnotationRelfect。↩ Class  $<?> c = null ; \leftarrow$ c = Class.forName("org.lxh.demo16.reflectannotation.SimpleBeanTwo");← 第(4)页 共(6)页 可以登录 http://BBS.MLDN.CN进行技术交流→

c = Class.forName("org.lxh.demo16.reflectannotation.SimpleBeanOne");← Method toM = c.getMethod("toString") ♯ 找到 toString()方法↩ Annotation an[] = toM.getAnnotations(); # 取得全部的 Annotation↓ 此时,已经取得了一个 Annotation。 ~ 以上的操作代码实际上是通过三个系统内建的 Annotation完成的,那么也可以自定义一个 Annotation.↩

类型 普通 普通 普通 普通 普通

魔 乐 科 技 JAVA就业专家 www.MLDNJAVA.cn\_ 存在以下几种与Annotation操作有关的方法

如果要想让一个Annotation起作用,则必须结合反射机制。在Class类中 描述 如果在一个元素中存在注释,则取得全部注 返回此元素上的所有注释 返回直接存放在此元素上的所有注释 判断元素是否表示一个注释 判断一个元素上是否存在注释

www.MLDNJAVA.cn

同时设置了三个 Annotation,那么此时注意,只有 Depreacted 的 Anntation 的定义范围是 RUNTIME 第(3)页 共(6)页 可以登录 http://BBS.MLDN.CN进行技术交流→

www.MLDNJAVA.cn 一个 Annotation 如果要是想让其变得有意义,则必须结合反射机制取得 Annotation 中设置的全部内 第(2)页 共(6)页 可以登录 http://BBS.MLDN.CN进行技术交流→

No.

import java.lang.annotation.Annotation; 魔 乐 科 技 JAVA就业专家 www.MLDNJAVA.cn\_

〃此 Annotation 在类执行时依然有效ゼ

魔 乐 科 技 JAVA就业专家 www.MLDNJAVA.cn\_

第(5)页 共(6)页 **可以登录 http://BBS.MLDN.CN进行技术交流**₽

MLDN 魔乐科技 JAVA就业专家 www.MLDNJAVA.cn\_

第(5)页 共(6)页 **可以登录 http://BBS.MLDN.CN进行技术交流**₽

深入Annotation

第(6)页 共(6)页

www.MLDNJAVA.cn

可以登录 http://BBS.MLDN.CN进行技术交流→