

1、课程名称：反射与 Annotation



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

2、本课程预计讲解的知识点



本章目标

- ◆ 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

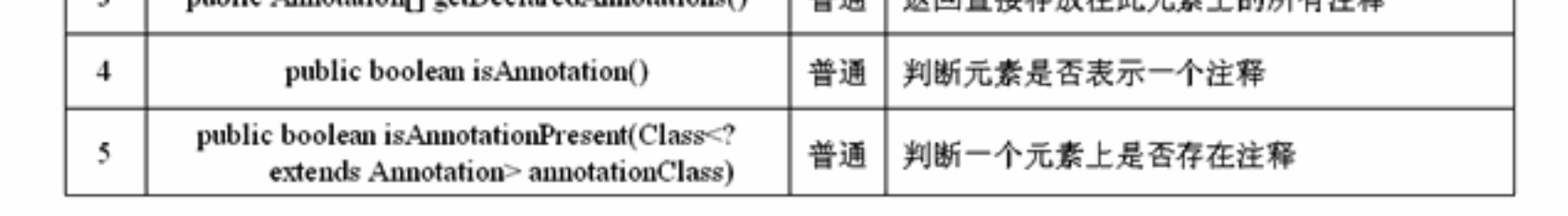
E-MAIL: mldnqa@163.com

www.MLDNJAVA.cn

3、具体内容

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 中设置的全部内容。

第 (2) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流



- ◆ 如果要想让一个Annotation起作用，则必须结合反射机制。在Class类中存在以下几种与Annotation操作有关的方法

No.	方法	类型	描述
1	public <A extends Annotation> A getAnnotation(Class<A> annotationClass)	普通	如果在一个元素中存在注释，则取得全部注释
2	public Annotation[] getAnnotations()	普通	返回此元素上的所有注释
3	public Annotation[] getDeclaredAnnotations()	普通	返回直接存放在此元素上的所有注释
4	public boolean isAnnotation()	普通	判断元素是否表示一个注释
5	public boolean isAnnotationPresent(Class<? extends Annotation> annotationClass)	普通	判断一个元素上是否存在注释

E-MAIL: mldnqa@163.com

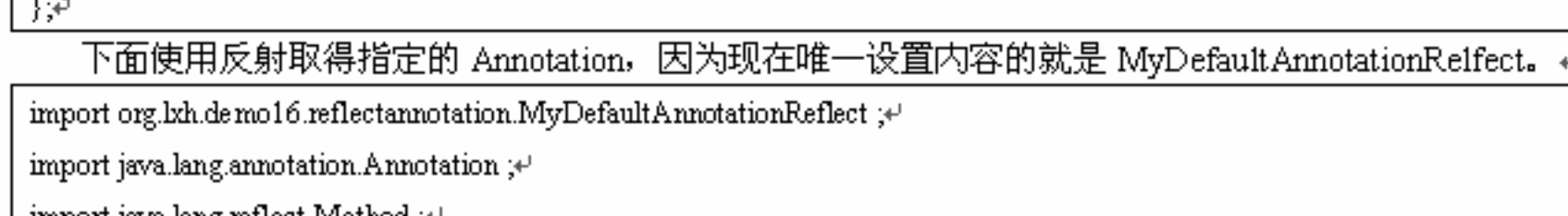
www.MLDNJAVA.cn

```
package org.lzh.demo16.reflectannotation ;  
public class SimpleBeanOne {  
    @SuppressWarnings("unchecked")  
    @Deprecated  
    @Override  
    public String toString(){  
        return "Hello LiXingHua!!!" ;  
    }  
};
```

同时设置了三个 Annotation，那么此时注意，只有 Deprecated 的 Annotation 的定义范围是 RUNTIME 范围，所以此时通过反射的话只能取得一个。

```
import java.lang.annotation.Annotation ;  
import java.lang.reflect.Method ;  
public class ReflectDemo01 {  
    public static void main(String args[]) throws Exception{ // 所有异常抛出
```

第 (3) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流



- ◆ 如果要想让一个Annotation起作用，则必须结合反射机制。在Class类中存在以下几种与Annotation操作有关的方法

No.	方法	类型	描述
1	public <A extends Annotation> A getAnnotation(Class<A> annotationClass)	普通	如果在一个元素中存在注释，则取得全部注释
2	public Annotation[] getAnnotations()	普通	返回此元素上的所有注释
3	public Annotation[] getDeclaredAnnotations()	普通	返回直接存放在此元素上的所有注释
4	public boolean isAnnotation()	普通	判断元素是否表示一个注释
5	public boolean isAnnotationPresent(Class<? extends Annotation> annotationClass)	普通	判断一个元素上是否存在注释

E-MAIL: mldnqa@163.com

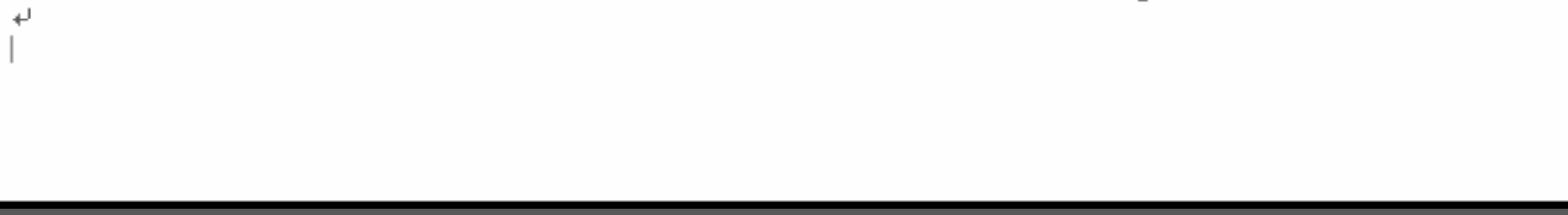
www.MLDNJAVA.cn

```
package org.lzh.demo16.reflectannotation ;  
public class SimpleBeanOne {  
    @SuppressWarnings("unchecked")  
    @Deprecated  
    @Override  
    public String toString(){  
        return "Hello LiXingHua!!!" ;  
    }  
};  
  
package org.lzh.demo16.reflectannotation ;  
import java.lang.annotation.Annotation ;  
import java.lang.annotation.Retention ;  
import java.lang.annotation.RetentionPolicy ;  
@Retention(value=RetentionPolicy.RUNTIME) // 此 Annotation 在类执行时依然有效  
public @interface MyDefaultAnnotationReflect {  
    public String key() default "LXH" ;  
    public String value() default "李兴华" ;  
}
```

同时设置了三个 Annotation，那么此时注意，只有 Deprecated 的 Annotation 的定义范围是 RUNTIME 范围，所以此时通过反射的话只能取得一个。

```
import java.lang.annotation.Annotation ;  
import java.lang.reflect.Method ;  
public class ReflectDemo01 {  
    public static void main(String args[]) throws Exception{ // 所有异常抛出
```

第 (3) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流



- ◆ 如果要想让一个Annotation起作用，则必须结合反射机制。在Class类中存在以下几种与Annotation操作有关的方法

No.	方法	类型	描述
1	public <A extends Annotation> A getAnnotation(Class<A> annotationClass)	普通	如果在一个元素中存在注释，则取得全部注释
2	public Annotation[] getAnnotations()	普通	返回此元素上的所有注释
3	public Annotation[] getDeclaredAnnotations()	普通	返回直接存放在此元素上的所有注释
4	public boolean isAnnotation()	普通	判断元素是否表示一个注释
5	public boolean isAnnotationPresent(Class<? extends Annotation> annotationClass)	普通	判断一个元素上是否存在注释

E-MAIL: mldnqa@163.com

www.MLDNJAVA.cn

```
package org.lzh.demo16.reflectannotation ;  
public class SimpleBeanOne {  
    @SuppressWarnings("unchecked")  
    @Deprecated  
    @Override  
    public String toString(){  
        return "Hello LiXingHua!!!" ;  
    }  
};  
  
package org.lzh.demo16.reflectannotation ;  
import java.lang.annotation.Annotation ;  
import java.lang.annotation.Retention ;  
import java.lang.annotation.RetentionPolicy ;  
@Retention(value=RetentionPolicy.RUNTIME) // 此 Annotation 在类执行时依然有效  
public @interface MyDefaultAnnotationReflect {  
    public String key() default "LXH" ;  
    public String value() default "李兴华" ;  
}
```

同时设置了三个 Annotation，那么此时注意，只有 Deprecated 的 Annotation 的定义范围是 RUNTIME 范围，所以此时通过反射的话只能取得一个。

```
import java.lang.annotation.Annotation ;  
import java.lang.reflect.Method ;  
public class ReflectDemo01 {  
    public static void main(String args[]) throws Exception{ // 所有异常抛出
```

第 (3) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流

5、预习任务

下一内容

E-MAIL: mldnqa@163.com

www.MLDNJAVA.cn

```
package org.lzh.demo16.reflectannotation ;  
public class SimpleBeanOne {  
    @SuppressWarnings("unchecked")  
    @Deprecated  
    @Override  
    public String toString(){  
        return "Hello LiXingHua!!!" ;  
    }  
};  
  
package org.lzh.demo16.reflectannotation ;  
import java.lang.annotation.Annotation ;  
import java.lang.annotation.Retention ;  
import java.lang.annotation.RetentionPolicy ;  
@Retention(value=RetentionPolicy.RUNTIME) // 此 Annotation 在类执行时依然有效  
public @interface MyDefaultAnnotationReflect {  
    public String key() default "LXH" ;  
    public String value() default "李兴华" ;  
}
```

同时设置了三个 Annotation，那么此时注意，只有 Deprecated 的 Annotation 的定义范围是 RUNTIME 范围，所以此时通过反射的话只能取得一个。

```
import java.lang.annotation.Annotation ;  
import java.lang.reflect.Method ;  
public class ReflectDemo01 {  
    public static void main(String args[]) throws Exception{ // 所有异常抛出
```

第 (3) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流

下一内容

E-MAIL: mldnqa@163.com

www.MLDNJAVA.cn

```
package org.lzh.demo16.reflectannotation ;  
public class SimpleBeanOne {  
    @SuppressWarnings("unchecked")  
    @Deprecated  
    @Override  
    public String toString(){  
        return "Hello LiXingHua!!!" ;  
    }  
};  
  
package org.lzh.demo16.reflectannotation ;  
import java.lang.annotation.Annotation ;  
import java.lang.annotation.Retention ;  
import java.lang.annotation.RetentionPolicy ;  
@Retention(value=RetentionPolicy.RUNTIME) // 此 Annotation 在类执行时依然有效  
public @interface MyDefaultAnnotationReflect {  
    public String key() default "LXH" ;  
    public String value() default "李兴华" ;  
}
```

同时设置了三个 Annotation，那么此时注意，只有 Deprecated 的 Annotation 的定义范围是 RUNTIME 范围，所以此时通过反射的话只能取得一个。

```
import java.lang.annotation.Annotation ;  
import java.lang.reflect.Method ;  
public class ReflectDemo01 {  
    public static void main(String args[]) throws Exception{ // 所有异常抛出
```

第 (3) 页 共 (6) 页 可以登录 <http://BBS.MLDN.CN> 进行技术交流