[**深入理解Java：注解（Annotation）--注解处理器**](http://www.cnblogs.com/peida/archive/2013/04/26/3038503.html)

　　如果没有用来读取注解的方法和工作，那么注解也就不会比注释更有用处了。使用注解的过程中，很重要的一部分就是创建于使用注解处理器。Java SE5扩展了反射机制的API，以帮助程序员快速的构造自定义注解处理器。

**注解处理器类库(java.lang.reflect.AnnotatedElement)：**

　　Java使用Annotation接口来代表程序元素前面的注解，该接口是所有Annotation类型的父接口。除此之外，Java在java.lang.reflect 包下新增了AnnotatedElement接口，该接口代表程序中可以接受注解的程序元素，该接口主要有如下几个实现类：

　　Class：类定义  
　　Constructor：构造器定义  
　　Field：累的成员变量定义  
　　Method：类的方法定义  
　　Package：类的包定义

　　java.lang.reflect 包下主要包含一些实现反射功能的工具类，实际上，java.lang.reflect 包所有提供的反射API扩充了读取运行时Annotation信息的能力。当一个Annotation类型被定义为运行时的Annotation后，该注解才能是运行时可见，当class文件被装载时被保存在class文件中的Annotation才会被虚拟机读取。  
　　AnnotatedElement 接口是所有程序元素（Class、Method和Constructor）的父接口，所以程序通过反射获取了某个类的AnnotatedElement对象之后，程序就可以调用该对象的如下四个个方法来访问Annotation信息：

　　方法1：<T extends Annotation> T getAnnotation(Class<T> annotationClass): 返回改程序元素上存在的、指定类型的注解，如果该类型注解不存在，则返回null。  
　　方法2：Annotation[] getAnnotations():返回该程序元素上存在的所有注解。  
　　方法3：boolean is AnnotationPresent(Class<?extends Annotation> annotationClass):判断该程序元素上是否包含指定类型的注解，存在则返回true，否则返回false.  
　　方法4：Annotation[] getDeclaredAnnotations()：返回直接存在于此元素上的所有注释。与此接口中的其他方法不同，该方法将忽略继承的注释。（如果没有注释直接存在于此元素上，则返回长度为零的一个数组。）该方法的调用者可以随意修改返回的数组；这不会对其他调用者返回的数组产生任何影响。

　　一个简单的注解处理器：



/\*\*\*\*\*\*\*\*\*\*\*注解声明\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* 水果名称注解

\* @author peida

\*

\*/

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

@Documented

public @interface FruitName {

String value() default "";

}

/\*\*

\* 水果颜色注解

\* @author peida

\*

\*/

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

@Documented

public @interface FruitColor {

/\*\*

\* 颜色枚举

\* @author peida

\*

\*/

public enum Color{ BULE,RED,GREEN};

/\*\*

\* 颜色属性

\* @return

\*/

Color fruitColor() default Color.GREEN;

}

/\*\*

\* 水果供应者注解

\* @author peida

\*

\*/

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

@Documented

public @interface FruitProvider {

/\*\*

\* 供应商编号

\* @return

\*/

public int id() default -1;

/\*\*

\* 供应商名称

\* @return

\*/

public String name() default "";

/\*\*

\* 供应商地址

\* @return

\*/

public String address() default "";

}

/\*\*\*\*\*\*\*\*\*\*\*注解使用\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class Apple {

@FruitName("Apple")

private String appleName;

@FruitColor(fruitColor=Color.RED)

private String appleColor;

@FruitProvider(id=1,name="陕西红富士集团",address="陕西省西安市延安路89号红富士大厦")

private String appleProvider;

public void setAppleColor(String appleColor) {

this.appleColor = appleColor;

}

public String getAppleColor() {

return appleColor;

}

public void setAppleName(String appleName) {

this.appleName = appleName;

}

public String getAppleName() {

return appleName;

}

public void setAppleProvider(String appleProvider) {

this.appleProvider = appleProvider;

}

public String getAppleProvider() {

return appleProvider;

}

public void displayName(){

System.out.println("水果的名字是：苹果");

}

}

/\*\*\*\*\*\*\*\*\*\*\*注解处理器\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class FruitInfoUtil {

public static void getFruitInfo(Class<?> clazz){

String strFruitName=" 水果名称：";

String strFruitColor=" 水果颜色：";

String strFruitProvicer="供应商信息：";

Field[] fields = clazz.getDeclaredFields();

for(Field field :fields){

if(field.isAnnotationPresent(FruitName.class)){

FruitName fruitName = (FruitName) field.getAnnotation(FruitName.class);

strFruitName=strFruitName+fruitName.value();

System.out.println(strFruitName);

}

else if(field.isAnnotationPresent(FruitColor.class)){

FruitColor fruitColor= (FruitColor) field.getAnnotation(FruitColor.class);

strFruitColor=strFruitColor+fruitColor.fruitColor().toString();

System.out.println(strFruitColor);

}

else if(field.isAnnotationPresent(FruitProvider.class)){

FruitProvider fruitProvider= (FruitProvider) field.getAnnotation(FruitProvider.class);

strFruitProvicer=" 供应商编号："+fruitProvider.id()+" 供应商名称："+fruitProvider.name()+" 供应商地址："+fruitProvider.address();

System.out.println(strFruitProvicer);

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*输出结果\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class FruitRun {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

FruitInfoUtil.getFruitInfo(Apple.class);

}

}

====================================

水果名称：Apple

水果颜色：RED

供应商编号：1 供应商名称：陕西红富士集团 供应商地址：陕西省西安市延安路89号红富士大厦



 　　Java注解的基础知识点（见下面导图）基本都过了一遍，下一篇我们通过设计一个基于注解的简单的ORM框架，来综合应用和进一步加深对注解的各个知识点的理解和运用。

