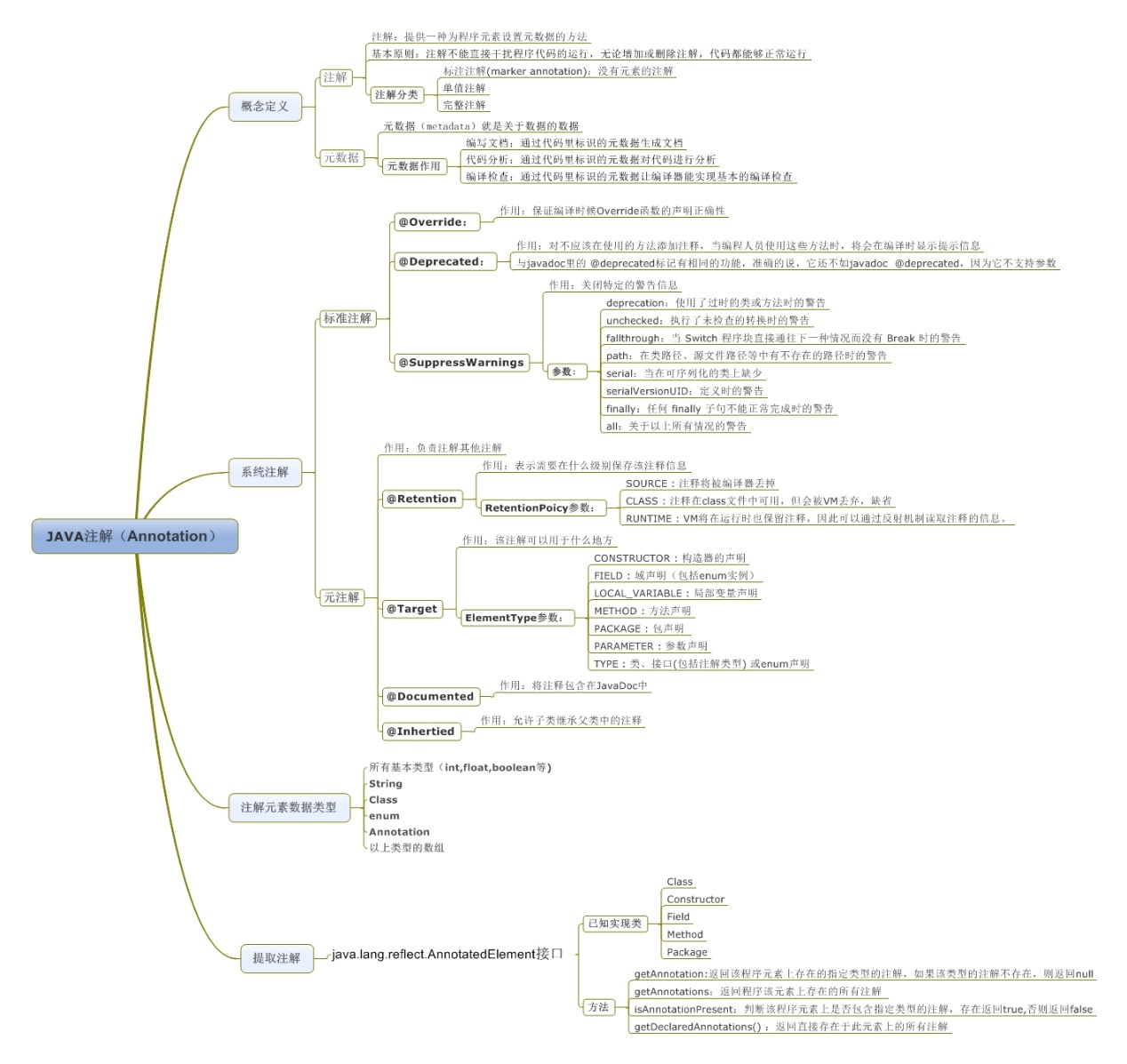
# TODO

# 脑图



**1.java元注解**

|  |
| --- |
| **[java]** [view plain](http://blog.csdn.net/lu597203933/article/details/38666303),java.lang.annotation.RetentionPolicy.java  **public enum RetentionPolicy {**/\*保留策略\*/  **SOURCE,**/\*这个Annotation类型的信息只会保留在程序.java源码里，并不会保留在编译好的.class文件里面\*/  **CLASS,**/\*这个Annotation类型的信息保留在程序源码.java、.class文件里面,在执行的时候，并不会加载到虚拟机(JVM)中去.注意一下。当你没有设定一个Annotation类型的Retention值时，系统默认值是CLASS.\*/  **RUNTIME**/\*在源码、编译好的.class文件中保留信息，在执行的时候会把这一些信息加载到JVM中去的\*/  **}** |

eg:@Override里面的Retention设为SOURCE,编译成功了就不要这一些检查的信息;相反,@Deprecated里面的Retention设为RUNTIME,表示除了在编译时会警告我们使用了哪个被Deprecated的方法,在执行的时候也可以查出该方法是否被Deprecated.

|  |
| --- |
| **[java] view plain,java.lang.annotation.RetentionPolicy.java**  **@Documented**  **@Retention(RetentionPolicy.RUNTIME)**  **@Target(ElementType.ANNOTATION\_TYPE)**  **public @interface Retention {**  **RetentionPolicy value();**  **}** |
| **[java]** [view plain](http://blog.csdn.net/lu597203933/article/details/38666303),java.lang.annotation.ElementType.java  **public enum ElementType {**  **TYPE**/\* Class, interface (including annotation type), or enum declaration \*/  **, FIELD, METHOD, PARAMETER, CONSTRUCTOR,**  **LOCAL\_VARIABLE, ANNOTATION\_TYPE,PACKAGE**  **}** |

|  |
| --- |
| **[java]** [view plain](http://blog.csdn.net/lu597203933/article/details/38666303),java.lang.annotation.Target.java  **@Documented**  **@Retention(RetentionPolicy.RUNTIME)**  **@Target(ElementType.ANNOTATION\_TYPE)**  **public @interface Target {**  **ElementType[] value();**  **}** |

 ＠Target里面的ElementType是用来指定Annotation类型可以用在哪一些元素上的.说明一下：TYPE(类型), FIELD(属性), METHOD(方法), PARAMETER(参数), CONSTRUCTOR(构造函数),LOCAL\_VARIABLE(局部变量), ANNOTATION\_TYPE,PACKAGE(包),其中的TYPE(类型)是指可以用在Class,Interface,Enum和Annotation类型上.

 另外,从1的源代码可以看出,@Target自己也用了自己来声明自己,只能用在ANNOTATION\_TYPE之上.

 如果一个Annotation类型没有指明@Target使用在哪些元素上,那么它可以使用在任何元素之上,这里的元素指的是上面的八种类型

|  |
| --- |
| **[java]** [view plain](http://blog.csdn.net/lu597203933/article/details/38666303),java.lang.annotation.**Documented**.java  **@Documented**  **@Retention(RetentionPolicy.RUNTIME)**  **@Target(ElementType.ANNOTATION\_TYPE)**  **public @interface Documented {**  **}** |

使用了@Documented,@Documented的目的就是让这一个Annotation类型的信息能够显示在javaAPI说明文档上;没有添加的话，使用javadoc生成API文档的时候就会找不到这一个类型生成的信息

|  |
| --- |
| **[java]** [view plain](http://blog.csdn.net/lu597203933/article/details/38666303),java.lang.annotation.**Inherited**.java  **@Documented**  **@Retention(RetentionPolicy.RUNTIME)**  **@Target(ElementType.ANNOTATION\_TYPE)**  **public @interface Inherited {**  **}** |

把Annotation的数据继承给子类，那么就会用到@Inherited这一个Annotation类型.

2.

参考：

1.[三个annotation类型(RetentionPolicy，ElementType)](http://blog.csdn.net/laixiaonian/article/details/8814100)

<http://blog.csdn.net/laixiaonian/article/details/8814100>

# REF