* JDK的元Annotation
* JDK的元Annotation**用于修饰其他Annotation定义**。
* JDK5.0提供了专门在注解上的注解类型，分别是：
  + Retention
  + Target
  + Documented
  + Inherited
* **@Retention**：只能用于修饰一个Annotation定义，用于指定该Annotation可以保留多长时间，@Rentention包含一个RetentionPolicy类型的成员变量，使用@Rentention时必须为该value成员变量指定值：
  + RetentionPolicy.SOURCE：编译器直接丢弃这种策略的注释
  + RetentionPolicy.CLASS：编译器将把注释记录在class文件中。当运行Java程序时，JVM不会保留注解。这是默认值
  + RetentionPolicy.RUNTIME：编译器将把注释记录在class文件中。当运行Java程序时，JVM会保留注释.程序可以通过反射获取该注释
* **@Target**：用于修饰Annotation定义，用于**指定被修饰的Annotation能用于修饰哪些程序元素**。@Target也包含一个名为value的成员变量。
* @Documented:用于指定被该元Annotation修饰的Annotation类将被javadoc工具提取成文档。
  + 定义为Documented的注解必须设置Retention值为RUNTIME。
* @Inherited:被它修饰的Annotation将具有**继承性**。如果某个类使用了被@Inherited修饰的Annotation,，则其子类将自动具有该注解。
  + 实际应用中，使用较少。