1. **定义注解，**例如：

package onjava.atunit;

import java.lang.annotation.\*;

@Target(ElementType.METHOD)

@Retention(RetentionPolicy.RUNTIME)

public @interface Test {}

@Target 定义你的注解可以应用在哪里（例如是方法还是字段）。

@Retention 定义了注解在哪里可用，在源代码中（SOURCE），class文件（CLASS）中或者是在运行(RUNTIME)

@Interited允许子类继承父类的注解

@Repeatable允许一个注解可以被使用一次或者多次

@Documented将此注解保存在 Javadoc 中(注释中使用)

**编写注解处理器**

Java 拓展了反射机制的 API 用于帮助你创造这类工具。同时他还提供了 javac 编译器钩子在编译时使用注解。

**注解元素**

- 所有基本类型（int、float、boolean等）

- String

- Class

- enum

- Annotation

- 以上类型的数组

任何非基本类型的元素，无论是在源代码声明时还是在注解接口中定义默认值时，都不能使用 null 作为其值。

**注解的作用：**

生成外部文件，试图或者是尝试完全代替xml配置文件，但是目前技术还做不到完全替代

**实现注解处理器**