**JAVA注解：**

@Override的产生是为了解决误把重写写成重载的情况

@Deprecated - 标记过时方法。如果使用该方法，会报编译警告。

@SuppressWarnings - 指示编译器去忽略注解中声明的警告。

@Retention - 标识这个注解怎么保存，是只在代码中，还是编入class文件中，或者是在运行时可以通过反射访问。

@Documented - 标记这些注解是否包含在用户文档中。

@Target - 标记这个注解应该是哪种 Java 成员。

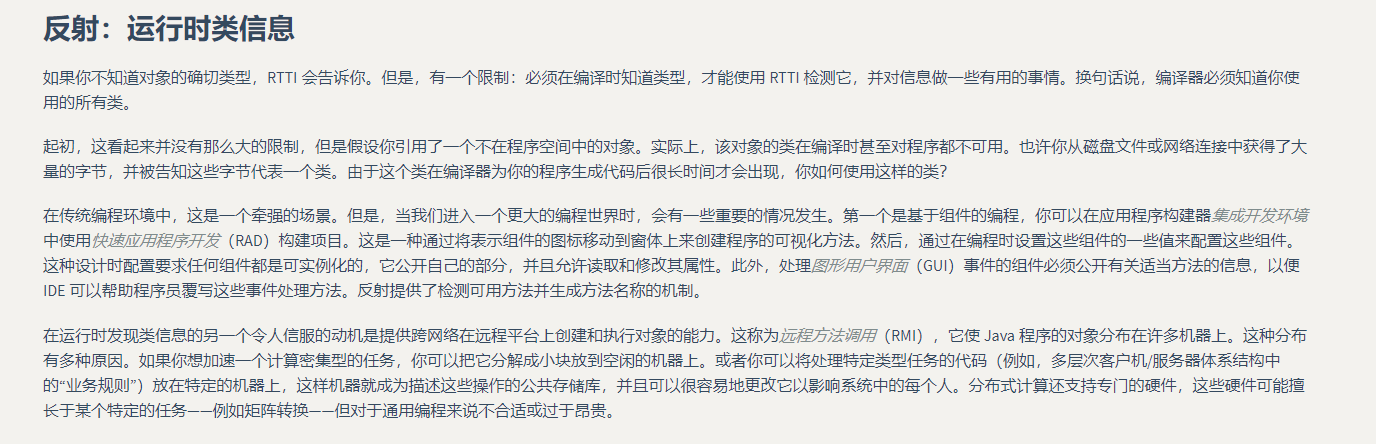
@Inherited - 标记这个注解是继承于哪个注解类(默认 注解并没有继承于任何子类)

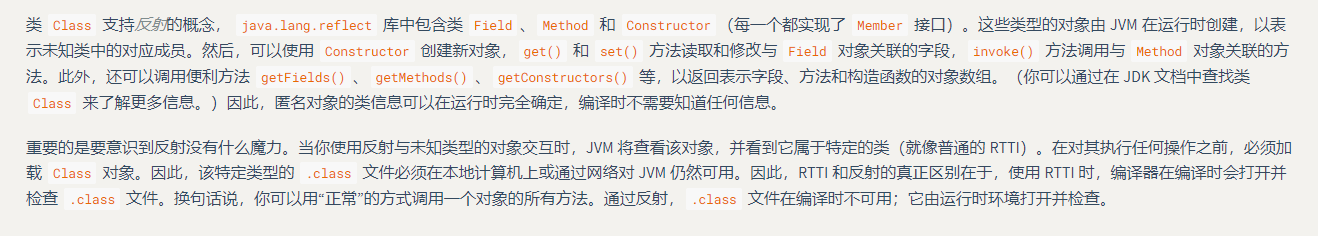
@SafeVarargs - Java 7 开始支持，忽略任何使用参数为泛型变量的方法或构造函数调用产生的警告。

@FunctionalInterface - Java 8 开始支持，标识一个匿名函数或函数式接口。

@Repeatable - Java 8 开始支持，标识某注解可以在同一个声明上使用多次。

**反射的定义以及常用方法：**





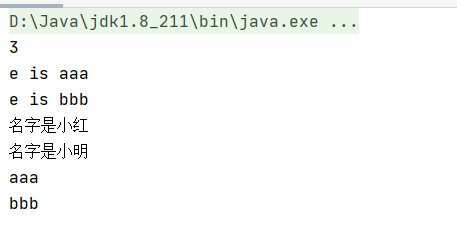
**函数式接口：**

1. 函数式接口(Functional Interface)就是一个有且仅有一个抽象方法，但是可以有多个非抽象方法的接口。
2. 函数式接口可以被隐式转换为 lambda 表达式。



匿名函数3种方式实现：





匿名函数方法：



