

一、反射作用

- i. 反编译：将 class 文件反编译成 Java 文件。
- ii. 访问对象：访问 Java 对象中的属性、方法、构造方法等。

二、需要使用的类

- i. java.lang.Class - 类，某个具体的类或接口
- ii. java.lang.reflect.Constructor - 反射构造方法
- iii. java.lang.reflect.Field - 反射属性
- iv. java.lang.reflect.Method - 反射方法
- v. java.lang.reflect.Modifier - 访问修饰符的信息

三、获取 Class 对象的三种方法

- i. Class.forName 方式，会让ClassLoader装载类，并进行类的初始化
- ii. .class 方式，ClassLoader 装载入内存，不对类进行类的初始化操作
- iii. getClass 方式，返回类对象运行时真正所指的对象，所属类型的Class对象

四、获取构造函数(四个方法)

- i. getConstructor(Class[] params) 根据构造函数的参数，返回一个具体的具有public属性的构造函数
- ii. getConstructors() 返回所有具有public属性的构造函数数组
- iii. getDeclaredConstructor(Class[] params) 根据构造函数的参数，返回一个具体的构造函数（不分public和非public属性）
- iv. getDeclaredConstructors() 返回该类中所有的构造函数数组（不分public和非public属性）

五、获取反射对象

- i. newInstance() 方法返回的是一个泛型T，我们要强转成相应的反射类。
- ii. 反射创建对象，可以使用Class.newInstance() 和 Constructor.newInstance() 两种方式，而前者必须要求反射类中存在一个无参的构造方法，并且有访问权限；后者适应各种类型的构造方法，无论是否有参数都可以调用，这是需要 setAccessible() 控制访问验证即可。一般建议是使用后者

六、获取反射类的方法(四个方法)

- i. `getMethod(String name, Class[] params)` 根据方法名和参数，返回一个具体的具有public属性的方法
- ii. `getMethods()` 返回所有具有public属性的方法数组
- iii. `getDeclaredMethod(String name, Class[] params)` 根据方法名和参数，返回一个具体的方法（不分public和非public属性）
- iv. `getDeclaredMethods()` 返回该类中的所有的方法数组（不分public和非public属性）

`getMethods()`：用于获取类的所有的public修饰域的成员方法，包括从父类继承的public方法和实现接口的public方法；

`getDeclaredMethods()`：用于获取在当前类中定义的所有的成员方法和实现的接口方法，不包括从父类继承的方法。

七、获取反射类的属性(四个方法)

- i. `getField(String name)` 根据变量名，返回一个具体的具有public属性的成员变量
- ii. `getFields()` 返回具有public属性的成员变量的数组
- iii. `getDeclaredField(String name)` 根据变量名，返回一个成员变量（不分public和非public属性）
- iv. `getDelcaredField()` 返回所有成员变量组成的数组（不分public和非public属性）

八、获取类、属性、方法的修饰符

通过 `int getModifiers()` 方法获得修饰符，方法返回是一个 `int` 型数值，每个修饰符都对应一个int型。