# Java 反射

## 一、什么是反射

在运行状态下，对于任何一个类，都知道有那些方法和属性。进而对其调用。对于这种动态获取对象信息和调用对象方法的功能称之为反射

## 二、反射的方式

|  |
| --- |
| @Test  **public** **void** test1() { // 第一种  Person person = **new** Person();  Class clazz = person.getClass();  System.***out***.println(clazz.getName());  }  @Test  **public** **void** test2() { // 第二种  Class clazz = Person.**class**;  System.***out***.println(clazz.getName());  }  @Test  **public** **void** test3() { // 第三种  **try** {  Class clazz = Class.*forName*("com.xm.clazz.Person");  System.***out***.println(clazz.getName());  }**catch**(Exception e) {  e.printStackTrace();  }  } |

## **三、获取构造方法**

|  |
| --- |
| Class clazz = Class.*forName*("com.xm.clazz.Person");  /\*\* 获取所有公用的构造方法 \*/  Constructor[] ctrs = clazz.getConstructors();  **for**(Constructor ctr : ctrs) {  System.***out***.println(ctr);  }  /\*\* 获取所有的构造方法 \*/  Constructor[] dctrs = clazz.getDeclaredConstructors();  **for**(Constructor dctr : dctrs) {  System.***out***.println(dctr);  }  /\*\* 获取公用的无参构造方法 \*/  Constructor ctr = clazz.getConstructor(**null**);  System.***out***.println(ctr);  /\*\* 根据参数获取构造方法 \*/  Constructor dctr = clazz.getDeclaredConstructor(**new** Class[] {String.**class**,**int**.**class**});  System.***out***.println(dctr); |

## 四、获取类属性

|  |
| --- |
| Class clazz = Class.*forName*("com.xm.clazz.Person");  /\*\* 获取公共属性 \*/  Field[] fields = clazz.getFields();  **for**(Field field : fields) {  System.***out***.println(field);  }  System.***out***.println("-------------------------");  /\*\* 获取所有属性 \*/  Field [] dFields = clazz.getDeclaredFields();  **for**(Field dField : dFields) {  System.***out***.println(dField);  }  System.***out***.println("-------------------------");  /\*\* 获取userName属性并设置值 \*/  Field field = clazz.getDeclaredField("userName");  Object obj = clazz.getConstructor().newInstance();  field.setAccessible(**true**); // 暴力反射  field.set(obj,"zhangsan"); // 给属性设置值  Person person = (Person) obj;  System.***out***.println(person); |

## 五、获取类方法

|  |
| --- |
| Class clazz = Class.*forName*("com.xm.clazz.Person");  /\*\* 获取所有公用方法包含父类的 \*\*/  Method [] methods = clazz.getMethods();  **for**(Method method : methods) {  System.***out***.println(method);  }  /\*\* 获取本类的所有方法 \*\*/  Method[] dMethods = clazz.getDeclaredMethods();  **for**(Method dMethod : dMethods) {  System.***out***.println(dMethod);  }  Method method = clazz.getMethod("getUserName", **null**);  System.***out***.println(method);  Method dMethod = clazz.getDeclaredMethod("show",String.**class**,**int**.**class**);  Object obj = dMethod.invoke(clazz.getConstructor().newInstance(),"zhangsan",21);  System.***out***.println(obj); |

## 六、调用Mian方法

|  |
| --- |
| Class clazz = Class.*forName*("com.xm.clazz.Person");  Method method = clazz.getMethod("main",java.lang.String[].**class**);  System.***out***.println(method);  method.invoke(**null**,(Object) **new** String[]{"a"}); |