

27 反射

JAVAAEE高级

一：反射描述

1、反射是被视为动态语言的关键，反射机制允许程序在执行期借助于API取得任何类的内部信息，并能直接操作任意对象的内部属性及方法

1、反射机制提供的功能

- 1、判断任意一个对象所属的类
- 2、构造任意一个类的对象
- 3、判断任意一个类所具有的成员变量和方法
- 4、调用任意一个对象的成员变量和方法
- 5、生成动态代理

2、主要相关的API

java.lang.Class	代表一个类
java.lang.reflect.Method	代表类的方法
java.lang.reflect.Field	代表类的成员变量
java.lang.reflect.Constructor	代表类的构造方法

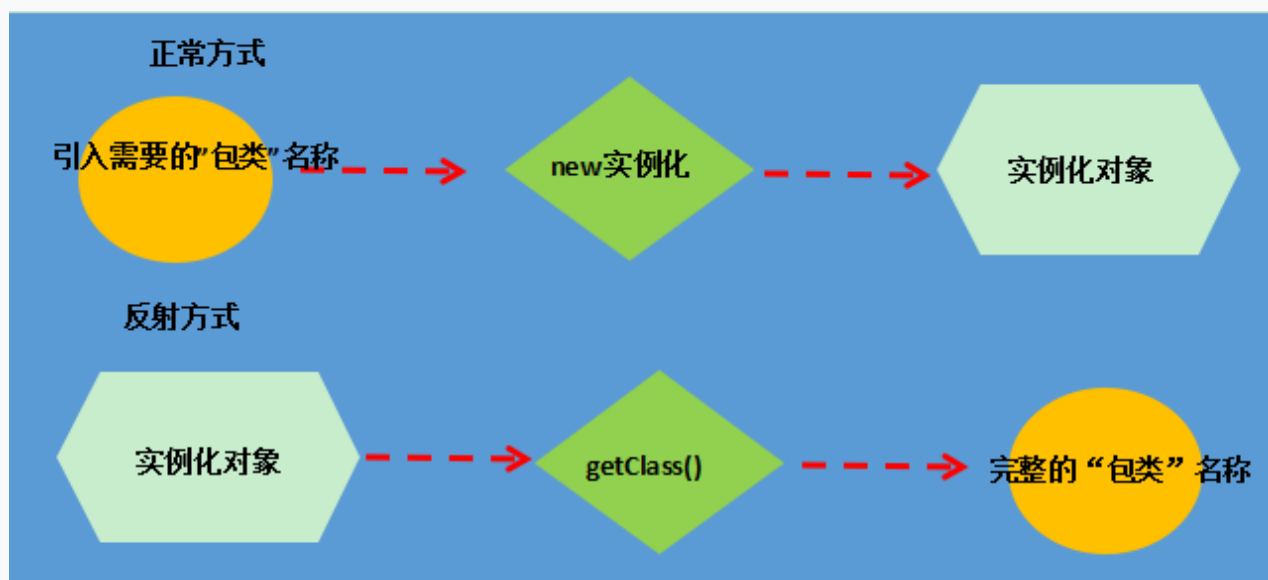
二：Class 类

- 1、在Object类中定义了以下的方法，此方法将被所有子类继承

```
1. public final Class getClass()
```

2、以上的方法返回值的类型是一个Class类，此类是Java反射的源头

3、实际上所谓反射从程序的运行结果来看也很好理解，即：可以通过对象反射求出类的名称



4、相当于对照镜子后可以得到的信息：某个类的属性、方法和构造器、某个类到底实现了哪些接口

5、对于每个类而言，JRE 都为其保留一个不变的 Class 类型的对象

6、这个 Class 对象包含了特定某个类的有关信息

7、Class本身也是一个类、对象只能由系统建立对象

8、一个类在 JVM 中只会有一个Class实例

9、一个Class对象对应的是一个加载到JVM中的一个.class文件

10、每个类的实例都会记得自己是由哪个 Class 实例所生成

11、通过Class可以完整地得到一个类中的完整结构

1、Class类的常用方法

方法	描述
static Class.forName(String name)	返回指定类名 name 的 Class 对象
Object newInstance()	调用缺省构造函数，返回该Class对象的一个实例
getName()	返回此Class对象所表示的实体（类、接口、数组类、基本类型或void）名称
Class getSuperClass()	返回当前Class对象的父类的Class对象
Class [] getInterfaces()	获取当前Class对象的接口
ClassLoader getClassLoader()	返回该类的类加载器
Class getSuperclass()	返回表示此Class所表示的实体的超类的Class
Constructor[] getConstructors()	返回一个包含某些Constructor对象的数组
Field[] getDeclaredFields()	返回Field对象的一个数组
Method getMethod(String name,Class ... paramTypes)	返回一个Method对象，此对象的形参类型为 paramType

2、实例化Class类对象方法

- 1、如已知具体的类，通过类的class属性获取该方法
最为安全可靠，程序性能最高

```
1.    Class clazz = String.class;
```

- 2、如已知某个类的实例，调用该实例的getClass()方法获取Class对象

```
1.    Person p = new Person();  
2.    Class cl = p.getClass();
```

- 3、如已知一个类的全类名，且该类在类路径下，可通过Class类的静态方法forName()获取

```
1.      Class c2 = Class.forName("java.lang.String");
```

```
1.      System.out.println(c2 == clazz);  
2.      System.out.println(c2 == cl);
```