<http://blog.csdn.net/u011637069/article/details/52093990>

版权声明：本文为博主原创文章，未经博主允许不得转载。 <http://blog.csdn.net/u011637069/article/details/52093990>

Java中Class类简介

  1. 在面向对象的世界里，万事万物皆对象。（java语言中，静态的成员、普通数据类型除外)  
   类是不是对象呢?类是(哪个类的对象呢?)谁的对象呢?  
   -- 类是对象，类是java.lang.Class类的实例对象

  2. 这个对象到底如何表示

  3 . Class.forName("类的全称")  
       不仅表示了，类的类类型，还代表了动态加载类  
       请大家区分编译、运行  
       编译时刻加载类是静态加载类、运行时刻加载类是动态加载类

  4. 基本的数据类型  
      void关键字  都存在类类型

  5. Class类的基本API操作

**[java]** [view plain](http://blog.csdn.net/u011637069/article/details/52093990) [copy](http://blog.csdn.net/u011637069/article/details/52093990)

1. **package** reflect;
3. **public** **class** ClassDemo1 {
4. **public** **static** **void** main(String[] args) {
5. //Foo的实例对象如何表示
6. Foo foo1 = **new** Foo();//foo1就表示出来了.
7. //Foo这个类 也是一个实例对象，Class类的实例对象,如何表示呢
8. //任何一个类都是Class的实例对象，这个实例对象有三种表示方式
10. //第一种表示方式--->实际在告诉我们任何一个类都有一个隐含的静态成员变量class
11. Class c1 = Foo.**class**;
13. //第二中表达方式  已经知道该类的对象通过getClass方法
14. Class c2 = foo1.getClass();
16. /\*官网 c1 ,c2 表示了Foo类的类类型(class type)
17. \* 万事万物皆对象，
18. \* 类也是对象，是Class类的实例对象
19. \* 这个对象我们称为该类的类类型
20. \*
21. \*/
23. //不管c1  or c2都代表了Foo类的类类型，一个类只可能是Class类的一个实例对象
24. System.out.println(c1 == c2);
26. //第三种表达方式
27. Class c3 = **null**;
28. **try** {
29. c3 = Class.forName("reflect.Foo");
30. } **catch** (ClassNotFoundException e) {
31. // TODO Auto-generated catch block
32. e.printStackTrace();
33. }
34. System.out.println(c2==c3);
36. //我们完全可以通过类的类类型创建该类的对象实例---->通过c1 or c2 or c3创建Foo的实例对象
37. **try** {
38. Foo foo = (Foo)c1.newInstance();//需要有无参数的构造方法
39. foo.print();
40. } **catch** (InstantiationException e) {
41. // TODO Auto-generated catch block
42. e.printStackTrace();
43. } **catch** (IllegalAccessException e) {
44. // TODO Auto-generated catch block
45. e.printStackTrace();
46. }

49. }
50. }
51. **class** Foo{
53. **void** print(){
54. System.out.println("foo");
55. }
56. }

在 面向对象的世界里，万事万物皆对象。基本类型、void、类、方法、成员变量等都是对象，他们都是java.lang.Class的一个实例对象。