### [newInstance() 的参数版本与无参数版本详解](http://xiaohuafyle.iteye.com/blog/1607258)

通过反射创建新的类示例，有两种方式：   
Class.newInstance()   
Constructor.newInstance()   
  
以下对两种调用方式给以比较说明：   
Class.newInstance() 只能够调用无参的构造函数，即默认的构造函数；   
Constructor.newInstance() 可以根据传入的参数，调用任意构造构造函数。   
  
Class.newInstance() 抛出所有由被调用构造函数抛出的异常。   
  
Class.newInstance() 要求被调用的构造函数是可见的，也即必须是public类型的;   
Constructor.newInstance() 在特定的情况下，可以调用私有的构造函数。   
  
Class A（被调用的示例）：

1. **public** **class** A {
2. **private** A() {
3. System.out.println("A's constructor is called.");
4. }
6. **private** A(**int** a, **int** b) {
7. System.out.println("a:" + a + " b:" + b);
8. }
9. }

Class B（调用者）：

1. **public** **class** B {
2. **public** **static** **void** main(String[] args) {
3. B b=**new** B();
4. out.println("通过Class.NewInstance()调用私有构造函数:");
5. b.newInstanceByClassNewInstance();
6. out.println("通过Constructor.newInstance()调用私有构造函数:");
7. b.newInstanceByConstructorNewInstance();
8. }
9. /\*通过Class.NewInstance()创建新的类示例\*/
10. **private** **void** newInstanceByClassNewInstance(){
11. **try** {/\*当前包名为reflect，必须使用全路径\*/
12. A a=(A)Class.forName("reflect.A").newInstance();
13. } **catch** (Exception e) {
14. out.println("通过Class.NewInstance()调用私有构造函数【失败】");
15. }
16. }
18. /\*通过Constructor.newInstance()创建新的类示例\*/
19. **private** **void** newInstanceByConstructorNewInstance(){
20. **try** {/\*可以使用相对路径，同一个包中可以不用带包路径\*/
21. Class c=Class.forName("A");
22. /\*以下调用无参的、私有构造函数\*/
23. Constructor c0=c.getDeclaredConstructor();
24. c0.setAccessible(**true**);
25. A a0=(A)c0.newInstance();
26. /\*以下调用带参的、私有构造函数\*/
27. Constructor c1=c.getDeclaredConstructor(**new** Class[]{**int**.**class**,**int**.**class**});
28. c1.setAccessible(**true**);
29. A a1=(A)c1.newInstance(**new** Object[]{5,6});
30. } **catch** (Exception e) {
31. e.printStackTrace();
32. }
33. }
34. }

输入结果如下：   
通过Class.NewInstance()调用私有构造函数:   
通过Class.NewInstance()调用私有构造函数【失败】   
通过Constructor.newInstance()调用私有构造函数:   
A's constructor is called.   
a:5 b:6   
  
说明方法newInstanceByClassNewInstance调用失败，而方法newInstanceByConstructorNewInstance则调用成功。   
如果被调用的类的构造函数为默认的构造函数，采用Class.newInstance()则是比较好的选择，   
一句代码就OK；如果是老百姓调用被调用的类带参构造函数、私有构造函数，   
就需要采用Constractor.newInstance()，两种情况视使用情况而定。   
不过Java Totorial中推荐采用Constractor.newInstance()。