java反射使用getConstructors的newInstance方法，抛出异常wrong number of arguments

在项目开发过程中，调用远程方法时抛出了异常wrong number of arguments，经过调试定位后发现，抛出异常的代码如下：

Class<?> stub= Class.forName(“Stub类的全类型名”);

Class<?> locator= Class.forName(“Locator类的全类型名”);

Service service = Object.getConstructors[1].newInstance(要访问URL的实例, locator);//在执行这一步时抛出了异常wrong number of arguments

经过调试发现，获取到的构造器中，索引为1的构造器入参并不是两个，而是一个，而后查看了该object所对应的类中的构造器，发现该类有三个构造器，一个无参数构造器，一个包含一个参数的构造器，还有一个包含两个参数的构造器。那么哪个索引所对应的构造器是我们想要的呢？

经过查阅资料发现，getConstructors方法返回的构造器并没有一个特定的顺序，也可以说是无序的，随机的，所以索引1位置的构造器并不能保证每次都是我们想要使用的构造器。

那之前为什么没有出现问题呢？在以前的代码中我们使用的jdk1.6\_85,一直没有更换过jdk，在固定jdk版本的编译下，产生的class中的构造器顺序可能是固定的，因此表现出来可能就是我们代码中构造器正序或者倒序后的元素下标，使用固定下标的构造器就不会出问题。

为什么这次会出现调用不通的问题？因为我们升级了jdk1.8\_191，导致编译时生成的class中构造器顺序变化，不再是下标为1的构造器，从而导致wrong number of arguments,即参数数量错误。

由此可知，在使用反射获取构造器并生成实例时，一定不要仅仅依赖构造器数组的元素下标来获取指定构造器；如果仅仅依赖下标，那么在构造器数量有变动或者构造器数量无变动，但是换用了其他版本的jdk或者环境时，都有可能导致获取不到指定的构造器，从而在创建实例时抛出wrong number of arguments异常。

怎么解决？

根据参数数量的不同获取指定构造器：

例：

int length = stub.getConstructors.length;

for (int i = 0; i< length; i++)

{

int count = stub.getConstructors[i].getParameterCount();

if(count == 2)

{

stub.getConstructors[i].newInstance(要访问URL的实例, locator);

}

}

但是这种方法不能够解决构造器参数数量相同的类。