

<http://www.blogjava.net/sundc/archive/2013/06/08/400382.html>

这几天闲来无事。在网上看了一个题目，相信大家知道这个题目

```
static void change(String str){  
    str="welcome";  
}
```

```
public static void main(String[] args) {  
    String str = "123";  
    change(str);  
    System.out.println(str);  
}
```

str = 多少?

如果大家都知道。。

然后我想把最后输出要“welcome”怎么做?

于是改成这样

```
static String change(String str){  
    str="welcome";  
    return str;  
}
```

然后main里

str=change(str);这结果改了，但是改的代码太多。

我想只能改change方法。也不需要返回值，来解决问题。先看了下

String类发现 private final char value[]; 是私有的。把这个值改了不就OK了嘛

于是想到了反射：

代码如下：

```
static void change(String str){  
    try {  
        Class<?> clazz = str.getClass();  
        Field fields = clazz.getDeclaredField("value");  
        Object obj = fields.get(str);  
        char [] charValue = (char [] )obj;  
        System.out.println(charValue);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

发现打出结果 welcome 那么只要改掉

charValue就OK了吧。于是继续加代码 发现 modifiers "private final" 会出这个错，

要加上

```
field.setAccessible(true);
```

```

try {
    Class<?> clazz = str.getClass();
    Field field = clazz.getDeclaredField("value");
    field.setAccessible(true);
    Object obj = field.get(str);
    char [] charValue = (char [])obj;
    charValue = new char [3];
    for(int i=0;i<charValue.length;i++){
        charValue[i]='a';
    }
    field.set(str, charValue);
} catch (Exception e) {
    e.printStackTrace();
}

```

发现能满足我的要求，。值是被改了，然后我想改变长度，于是又动一下代码charValue = new char [4];

发现值不能改变 继续看String源码，发现有个count属性，。于是又了改了这个值代码如下

```

try {
    Class<?> clazz = str.getClass();
    Field field = clazz.getDeclaredField("value");
    field.setAccessible(true);
    Object obj = field.get(str);
    char [] charValue = (char [])obj;
    charValue = new char [4];
    for(int i=0;i<charValue.length;i++){
        charValue[i]='a';
    }
    Field field1 = clazz.getDeclaredField("count");
    field1.setAccessible(true);
    field1.set(str, charValue.length);
    field.set(str, charValue);
} catch (Exception e) {
    e.printStackTrace();
}
}

```

发现能解决问题。

我又试了下

然后我又试下Integer 发现一样可以

代码如下：

```

static void changeInteger(Integer a){
    try {
        Class<?> clazz = a.getClass();

```

```
Field field = clazz.getDeclaredField("value");  
field.setAccessible(true);  
field.set(a, 12312);  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

个人觉得只要Field.setAccessible(true);之后，即使是final关键字标示过得属性也可以有访问权限！这样的反射会改变JAVA的结构，甚至你的代码可维护性，你完全可以改别的代码里面的值，所以这只是一个简单的例子。只是验证通过反射能做一些让你无法想象的东西。。