[**java提高篇（五）-----使用序列化实现对象的拷贝**](http://blog.csdn.net/chenssy/article/details/12952063)

标签： [java](http://www.csdn.net/tag/java" \t "_blank)[java提高篇](http://www.csdn.net/tag/java%e6%8f%90%e9%ab%98%e7%af%87)

2013-10-22 19:15 5748人阅读 [评论](http://blog.csdn.net/chenssy/article/details/12952063#comments)(9) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/chenssy/article/details/12952063#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

【JAVA开发】-----Java提高篇（36） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)](http://blog.csdn.net/chenssy/article/details/12952063" \o "系统根据文章中H1到H6标签自动生成文章目录)[[-]](http://blog.csdn.net/chenssy/article/details/12952063)

1. [一浅拷贝问题](http://blog.csdn.net/chenssy/article/details/12952063#t0)
2. [二利用序列化实现对象的拷贝](http://blog.csdn.net/chenssy/article/details/12952063#t1)

我们知道在**[Java](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "_blank)**中存在这个接口Cloneable，实现该接口的类都会具备被拷贝的能力，同时拷贝是在内存中进行，在性能方面比我们直接通过new生成对象来的快，特别是在大对象的生成上，使得性能的提升非常明显。然而我们知道拷贝分为深拷贝和浅拷贝之分，但是浅拷贝存在对象属性拷贝不彻底问题。关于深拷贝、浅拷贝的请参考这里：[渐析java的浅拷贝和深拷贝](http://www.cnblogs.com/chenssy/p/3308489.html" \t "_blank)

**一、浅拷贝问题**

我们先看如下代码：

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. **public** **class** Person **implements** Cloneable{
2. /\*\* 姓名 \*\*/
3. **private** String name;
5. /\*\* 电子邮件 \*\*/
6. **private** Email email;
8. **public** String getName() {
9. **return** name;
10. }
12. **public** **void** setName(String name) {
13. **this**.name = name;
14. }
16. **public** Email getEmail() {
17. **return** email;
18. }
20. **public** **void** setEmail(Email email) {
21. **this**.email = email;
22. }
24. **public** Person(String name,Email email){
25. **this**.name  = name;
26. **this**.email = email;
27. }
29. **public** Person(String name){
30. **this**.name = name;
31. }
33. **protected** Person clone() {
34. Person person = **null**;
35. **try** {
36. person = (Person) **super**.clone();
37. } **catch** (CloneNotSupportedException e) {
38. e.printStackTrace();
39. }
41. **return** person;
42. }
43. }
45. **public** **class** Client {
46. **public** **static** **void** main(String[] args) {
47. //写封邮件
48. Email email = **new** Email("请参加会议","请与今天12:30到二会议室参加会议...");
50. Person person1 =  **new** Person("张三",email);
52. Person person2 =  person1.clone();
53. person2.setName("李四");
54. Person person3 =  person1.clone();
55. person3.setName("王五");
57. System.out.println(person1.getName() + "的邮件内容是：" + person1.getEmail().getContent());
58. System.out.println(person2.getName() + "的邮件内容是：" + person2.getEmail().getContent());
59. System.out.println(person3.getName() + "的邮件内容是：" + person3.getEmail().getContent());
60. }
61. }
62. --------------------
63. Output:
64. 张三的邮件内容是：请与今天12:30到二会议室参加会议...
65. 李四的邮件内容是：请与今天12:30到二会议室参加会议...
66. 王五的邮件内容是：请与今天12:30到二会议室参加会议...

在该应用程序中，首先定义一封邮件，然后将该邮件发给张三、李四、王五三个人，由于他们是使用相同的邮件，并且仅有名字不同，所以使用张三该对象类拷贝李四、王五对象然后更改下名字即可。程序一直到这里都没有错，但是如果我们需要张三提前30分钟到，即把邮件的内容修改下：

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. **public** **class** Client {
2. **public** **static** **void** main(String[] args) {
3. //写封邮件
4. Email email = **new** Email("请参加会议","请与今天12:30到二会议室参加会议...");
6. Person person1 =  **new** Person("张三",email);
8. Person person2 =  person1.clone();
9. person2.setName("李四");
10. Person person3 =  person1.clone();
11. person3.setName("王五");
13. person1.getEmail().setContent("请与今天12:00到二会议室参加会议...");
15. System.out.println(person1.getName() + "的邮件内容是：" + person1.getEmail().getContent());
16. System.out.println(person2.getName() + "的邮件内容是：" + person2.getEmail().getContent());
17. System.out.println(person3.getName() + "的邮件内容是：" + person3.getEmail().getContent());
18. }
19. }

在这里同样是使用张三该对象实现对李四、王五拷贝，最后将张三的邮件内容改变为：请与今天12:00到二会议室参加会议...。但是结果是：

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. 张三的邮件内容是：请与今天12:00到二会议室参加会议...
2. 李四的邮件内容是：请与今天12:00到二会议室参加会议...
3. 王五的邮件内容是：请与今天12:00到二会议室参加会议...

这里我们就疑惑了为什么李四和王五的邮件内容也发送了改变呢？让他们提前30分钟到人家会有意见的！

其实出现问题的关键就在于clone()方法上，我们知道该clone()方法是使用Object类的clone()方法，但是该方法存在一个缺陷，它并不会将对象的所有属性全部拷贝过来，而是有选择性的拷贝，基本规则如下：

1、 基本类型

如果变量是基本很类型，则拷贝其值，比如int、float等。

2、 对象

如果变量是一个实例对象，则拷贝其地址引用，也就是说此时新对象与原来对象是公用该实例变量。

3、 String字符串

若变量为String字符串，则拷贝其地址引用。但是在修改时，它会从字符串池中重新生成一个新的字符串，原有紫都城对象保持不变。

基于上面上面的规则，我们很容易发现问题的所在，他们三者公用一个对象，张三修改了该邮件内容，则李四和王五也会修改，所以才会出现上面的情况。对于这种情况我们还是可以解决的，只需要在clone()方法里面新建一个对象，然后张三引用该对象即可：

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. **protected** Person clone() {
2. Person person = **null**;
3. **try** {
4. person = (Person) **super**.clone();
5. person.setEmail(**new** Email(person.getEmail().getObject(),person.getEmail().getContent()));
6. } **catch** (CloneNotSupportedException e) {
7. e.printStackTrace();
8. }
10. **return** person;
11. }

所以：浅拷贝只是Java提供的一种简单的拷贝机制，不便于直接使用。

对于上面的解决方案还是存在一个问题，若我们系统中存在大量的对象是通过拷贝生成的，如果我们每一个类都写一个clone()方法，并将还需要进行深拷贝，新建大量的对象，这个工程是非常大的，这里我们可以利用序列化来实现对象的拷贝。

**二、利用序列化实现对象的拷贝**

如何利用序列化来完成对象的拷贝呢？在内存中通过字节流的拷贝是比较容易实现的。把母对象写入到一个字节流中，再从字节流中将其读出来，这样就可以创建一个新的对象了，并且该新对象与母对象之间并不存在引用共享的问题，真正实现对象的深拷贝。

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. **public** **class** CloneUtils {
2. @SuppressWarnings("unchecked")
3. **public** **static** <T **extends** Serializable> T clone(T obj){
4. T cloneObj = **null**;
5. **try** {
6. //写入字节流
7. ByteArrayOutputStream out = **new** ByteArrayOutputStream();
8. ObjectOutputStream obs = **new** ObjectOutputStream(out);
9. obs.writeObject(obj);
10. obs.close();
12. //分配内存，写入原始对象，生成新对象
13. ByteArrayInputStream ios = **new** ByteArrayInputStream(out.toByteArray());
14. ObjectInputStream ois = **new** ObjectInputStream(ios);
15. //返回生成的新对象
16. cloneObj = (T) ois.readObject();
17. ois.close();
18. } **catch** (Exception e) {
19. e.printStackTrace();
20. }
21. **return** cloneObj;
22. }
23. }

使用该工具类的对象必须要实现Serializable接口，否则是没有办法实现克隆的。

**[html]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. public class Person implements Serializable{
2. private static final long serialVersionUID = 2631590509760908280L;
4. ..................
5. //去除clone()方法
7. }
9. public class Email implements Serializable{
10. private static final long serialVersionUID = 1267293988171991494L;
12. ....................
13. }

所以使用该工具类的对象只要实现Serializable接口就可实现对象的克隆，无须继承Cloneable接口实现clone()方法。

**[java]** [view plain](http://blog.csdn.net/chenssy/article/details/12952063) [copy](http://blog.csdn.net/chenssy/article/details/12952063)

[print?](http://blog.csdn.net/chenssy/article/details/12952063)[在CODE上查看代码片](https://code.csdn.net/snippets/129612)

1. **public** **class** Client {
2. **public** **static** **void** main(String[] args) {
3. //写封邮件
4. Email email = **new** Email("请参加会议","请与今天12:30到二会议室参加会议...");
6. Person person1 =  **new** Person("张三",email);
8. Person person2 =  CloneUtils.clone(person1);
9. person2.setName("李四");
10. Person person3 =  CloneUtils.clone(person1);
11. person3.setName("王五");
12. person1.getEmail().setContent("请与今天12:00到二会议室参加会议...");
14. System.out.println(person1.getName() + "的邮件内容是：" + person1.getEmail().getContent());
15. System.out.println(person2.getName() + "的邮件内容是：" + person2.getEmail().getContent());
16. System.out.println(person3.getName() + "的邮件内容是：" + person3.getEmail().getContent());
17. }
18. }
19. -------------------
20. Output:
21. 张三的邮件内容是：请与今天12:00到二会议室参加会议...
22. 李四的邮件内容是：请与今天12:30到二会议室参加会议...
23. 王五的邮件内容是：请与今天12:30到二会议室参加会议...

**巩固基础，提高技术，不惧困难，攀登高峰！！！！！！**

**参考文献《编写高质量代码 改善Java程序的151个建议》----秦小波**