链滴

java 深拷贝与浅拷贝的一些思考

作者：TWanGT

原文链接：https://ld246.com/article/1566358304034

来源网站：链滴

概念介绍

什么是浅拷贝

拷贝就是将数据复制 对象a 到另一个 对象b , 但是遇到包装类型的数据时, 浅拷贝 只是将对象的引用制了, 并没有新建一个对象, 所以导致修改a对象中x的值会导致b对象的x的值也发生改变,> 有时候为节省内存开销 浅拷贝 还是有很大的应用价值的

什么是深拷贝

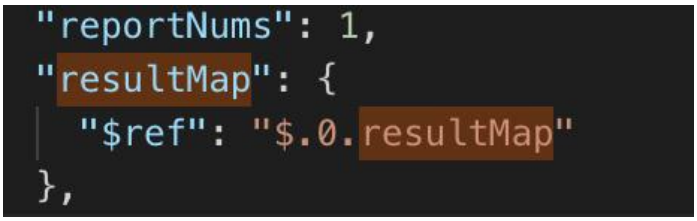
深拷贝拷贝从字面意思理解就是将拷贝深入到包装对象, 将 对象a 实实在在的新建一份 对象b (相当于样的东西占用2份内存), 这样 对象a 和 对象b 所做的修改就相互独立了

举例

深拷贝 和 浅拷贝 可以通过看对象的内存地址来区分(可以借助 IDE 工具的 debug 功能

这里我通过将对象用 json 输出来看下

浅拷贝:



深拷贝:

实现方式

使用BeanUtils

这个是 浅拷贝, 复制基本类型或包装类型的引用, 适用于对象转换(同名字段), 或者将多个对象的字段中到一个对象中的时候使用, 由于是 浅拷贝 所以修改其中一次, 会导致别的地方的值都同步修改, 不适于需要保留多个版本的情况

```
BeanUtils.copyProperties(source, target);
```

实现Cloneable接口

在类上实现克隆接口 implements Cloneable

并重写一下 clone 接口

如果字段是复杂类型, 需要该类型也实现 Cloneable 接口

```
@Override
public Object clone() throws CloneNotSupportedException {
return (xxx) super.clone();
}
```

在需要克隆的地方调用一下

```
obj.clone();
```

使用CloneUtils

需要复杂类型都实现 Cloneable 接口

```
CloneUtils.clone(object);
```

特殊的map类型

大多数的 map 类型都没有实现 Serializable 或者 Cloneable 接口(HashMap 和 LinkedHashMap 两者都实现了)

所以如果用 clone() 或者 CloneUtils.clone() 可能还是 浅拷贝

这时我们就需要修改我们使用的 map 类型来克隆, 或者通过别的序列化方式(例如: json)来重新生成象

这块我就不赘述了, 一个比较详细的参考

通过序列化

这个应该是最简单方便的方法了, 不过性能开销是最大的, 因为需要 序列化 一次, 然后再 反序列化 一次, 不过对于现在大多数情况下 cpu 资源过剩的倒也还可以接口 (高并发场景的话就最好不要用这个了...)

如果要序列化的话就需要实现 Serializable 接口

或者通过 json序列化 的方式来实现