



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”

1、什么是java序列化，如何实现java序列化？

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。序列化的实现：将需要被序列化的类实现Serializable接口，该接口没有需要实现的方法，implements Serializable只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个ObjectOutputStream(对象流)对象，接着，使用ObjectOutputStream对象的writeObject(Object obj)方法就可以将参数为obj的对象写出(即保存其状态)，要恢复的话则用输入流。

2、保存(持久化)对象及其状态到内存或者磁盘

Java 平台允许我们在内存中创建可复用的 Java 对象，但一般情况下，只有当 JVM 处于运行时，这些对象才可能存在，即，这些对象的生命周期不会比 JVM 的生命周期更长。但在现实应用中，就可能要求在 JVM 停止运行之后能够保存(持久化)指定的对象，并在将来重新读取被保存的对象。Java 对象序列化就能够帮助我们实现该功能。

3、序列化对象以字节数组保持-静态成员不保存

使用 Java 对象序列化，在保存对象时，会把其状态保存为一组字节，在未来，再将这些字节组装成对象。必须注意地是，对象序列化保存的是对象的“状态”，即它的成员变量。由此可知，对象序列化不会关注类中的静态变量。

4、序列化用户远程对象传输

除了在持久化对象时会用到对象序列化之外，当使用 RMI(远程方法调用)，或在网络中传递对象时，都会用到对象序列化。Java序列化API为处理对象序列化提供了一个标准机制，该API简单易用。

5、Serializable 实现序列化

在 Java 中，只要一个类实现了 `java.io.Serializable` 接口，那么它就可以被序列化。

`ObjectOutputStream` 和 `ObjectInputStream` 对对象进行序列化及反序列化通过 `ObjectOutputStream` 和 `ObjectInputStream` 对对象进行序列化及反序列化。

6、writeObject 和 readObject 自定义序列化策略

在类中增加 `writeObject` 和 `readObject` 方法可以实现自定义序列化策略。

7、序列化 ID

虚拟机是否允许反序列化，不仅取决于类路径和功能代码是否一致，一个非常重要的一点是两个类的序列化 ID 是否一致（就是 `private static final long serialVersionUID`）。

8、序列化并不保存静态变量

序列化子类说明 要想将父类对象也序列化，就需要让父类也实现 `Serializable` 接口。

9、Transient 关键字阻止该变量被序列化到文件中

1. 在变量声明前加上 `Transient` 关键字，可以阻止该变量被序列化到文件中，在被反序列化后，`transient` 变量的值被设为初始值，如 `int` 型的是 0，对象型的是 `null`。
2. 服务器端给客户端发送序列化对象数据，对象中有一些数据是敏感的，比如密码字符串等，希望对该密码字段在序列化时，进行加密，而客户端如果拥有解密的密钥，只有在客户端进行反序列化时，才可以对密码进行读取，这样可以一定程度保证序列化对象的数据安全。

10、序列化（深 clone 一中实现）

在 Java 语言里深复制一个对象，常常可以先使对象实现 `Serializable` 接口，然后把对象（实际上只是对象的一个拷贝）写到一个流里，再从流里读出来，便可以重建对象。