**Java 序列化**

Java 序列化是一种将对象转换为字节流的过程，以便可以将对象保存到磁盘上，将其传输到网络上，或者将其存储在内存中，以后再进行反序列化，将字节流重新转换为对象。

序列化在 Java 中是通过 java.io.Serializable 接口来实现的，该接口没有任何方法，只是一个标记接口，用于标识类可以被序列化。

当你序列化对象时，你把它包装成一个特殊文件，可以保存、传输或存储。反序列化则是打开这个文件，读取序列化的数据，然后将其还原为对象，以便在程序中使用。

序列化是一种用于保存、传输和还原对象的方法，它使得对象可以在不同的计算机之间移动和共享，这对于分布式系统、数据存储和跨平台通信非常有用。

以下是 Java 序列化的基本概念和用法：

实现 Serializable 接口： 要使一个类可序列化，需要让该类实现 java.io.Serializable 接口，这告诉 Java 编译器这个类可以被序列化，例如：

实例

import java.io.Serializable;

public class MyClass implements Serializable {

// 类的成员和方法

}

序列化对象： 使用 ObjectOutputStream 类来将对象序列化为字节流，以下是一个简单的实例：

实例

MyClass obj = new MyClass();

try {

FileOutputStream fileOut = new FileOutputStream("object.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(obj);

out.close();

fileOut.close();

} catch (IOException e) {

e.printStackTrace();

}

上述代码将一个名为 "object.ser" 的文件中的 obj 对象序列化。

反序列化对象： 使用 ObjectInputStream 类来从字节流中反序列化对象，以下是一个简单的实例：

实例

MyClass obj = null;

try {

FileInputStream fileIn = new FileInputStream("object.ser");

ObjectInputStream in = new ObjectInputStream(fileIn);

obj = (MyClass) in.readObject();

in.close();

fileIn.close();

} catch (IOException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

上述代码从 "object.ser" 文件中读取字节流并将其反序列化为一个 MyClass 对象。

类 ObjectInputStream 和 ObjectOutputStream 是高层次的数据流，它们包含反序列化和序列化对象的方法。

ObjectOutputStream 类包含很多写方法来写各种数据类型，但是一个特别的方法例外：

public final void writeObject(Object x) throws IOException

上面的方法序列化一个对象，并将它发送到输出流。相似的 ObjectInputStream 类包含如下反序列化一个对象的方法：

public final Object readObject() throws IOException,

ClassNotFoundException

该方法从流中取出下一个对象，并将对象反序列化。它的返回值为Object，因此，你需要将它转换成合适的数据类型。

实例

为了演示序列化在 Java 中是怎样工作的，我将使用之前教程中提到的 Employee 类，假设我们定义了如下的 Employee 类，该类实现了Serializable 接口。

Employee.java 文件代码：

public class Employee implements java.io.Serializable

{

public String name;

public String address;

public transient int SSN;

public int number;

public void mailCheck()

{

System.out.println("Mailing a check to " + name

+ " " + address);

}

}

请注意，一个类的对象要想序列化成功，必须满足两个条件：

该类必须实现 java.io.Serializable 接口。

该类的所有属性必须是可序列化的。如果有一个属性不是可序列化的，则该属性必须注明是短暂的。

如果你想知道一个 Java 标准类是否是可序列化的，请查看该类的文档。检验一个类的实例是否能序列化十分简单， 只需要查看该类有没有实现 java.io.Serializable接口。