**反序列化对象**

下面的 DeserializeDemo 程序实例了反序列化，/tmp/employee.ser 存储了 Employee 对象。

DeserializeDemo.java 文件代码：

import java.io.\*;

public class DeserializeDemo

{

public static void main(String [] args)

{

Employee e = null;

try

{

FileInputStream fileIn = new FileInputStream("/tmp/employee.ser");

ObjectInputStream in = new ObjectInputStream(fileIn);

e = (Employee) in.readObject();

in.close();

fileIn.close();

}catch(IOException i)

{

i.printStackTrace();

return;

}catch(ClassNotFoundException c)

{

System.out.println("Employee class not found");

c.printStackTrace();

return;

}

System.out.println("Deserialized Employee...");

System.out.println("Name: " + e.name);

System.out.println("Address: " + e.address);

System.out.println("SSN: " + e.SSN);

System.out.println("Number: " + e.number);

}

}

以上程序编译运行结果如下所示：

Deserialized Employee...

Name: Reyan Ali

Address:Phokka Kuan, Ambehta Peer

SSN: 0

Number:101

这里要注意以下要点：

readObject() 方法中的 try/catch代码块尝试捕获 ClassNotFoundException 异常。对于 JVM 可以反序列化对象，它必须是能够找到字节码的类。如果JVM在反序列化对象的过程中找不到该类，则抛出一个 ClassNotFoundException 异常。

注意，readObject() 方法的返回值被转化成 Employee 引用。

当对象被序列化时，属性 SSN 的值为 111222333，但是因为该属性是短暂的，该值没有被发送到输出流。所以反序列化后 Employee 对象的 SSN 属性为 0。