问：什么是serialVersionUID？  
  
答：序列化运行时将每个可序列化类与版本号相关联，称为serialVersionUID，在反序列化期间使用该版本号来验证序列化对象的发送方和接收方是否已加载与该序列化兼容的该对象的类。如果接收者为对象加载了一个类，该类具有与相应发送者类不同的serialVersionUID，则反序列化将导致anInvalidClassException。可序列化类可以通过声明名为“serialVersionUID”的字段来显式声明其自己的serialVersionUID，该字段必须是static，final和long类型：  
  
private static final long serialVersionUID=6201378234876555585L;  
  
SerialVersionUID是每个类的唯一标识符，JVM使用它来比较类的版本，确保在反序列化期间加载序列化期间使用相同的类。您必须声明serialVersionUID，因为它为我们提供了更多控制。  
  
问：如果我不想声明serialVersionUID会发生什么  
  
答：如果可序列化类没有显式声明serialVersionUID，则序列化运行时将根据类的各个方面计算该类的默认serialVersionUID值，如Java（TM）对象序列化规范中所述。  
  
但是，强烈建议所有可序列化类显式声明serialVersionUID值，因为默认的serialVersionUID计算对类详细信息高度敏感，这些详细信息可能因编译器实现而异，因此在反序列化期间可能会导致意外的InvalidClassExceptions。因此，为了保证跨不同java编译器实现的一致serialVersionUID值，可序列化类必须声明显式serialVersionUID值。强烈建议显式serialVersionUID声明尽可能使用private修饰符，因为此类声明仅适用于立即声明的类-serialVersionUID字段不适用于继承成员。  
  
问：我为什么要使用serialVersionUID？  
  
答：无论您选择哪种序列化形式，在您编写的每个可序列化类中声明一个显式的串行版本UID。这消除了串行版本UID作为不兼容的潜在来源（第74项）。还有一个小的性能优势。如果未提供串行版本UID，则需要在运行时生成一个昂贵的计算。  
  
问：如果我想忽略这样的警告，我该怎么办？  
  
答：Window>Preferences>Java>Compiler>Errors/Warnings>潜在的编程问题（只有eclipse）