

比较Hashtable类和HashMap类的不同之处 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/267/2021_2022__E6_AF_94_E8_BE_83Hash_c104_267209.htm Hashtable和HashMap类有三个重要的不同之处。第一个不同主要是历史原因。Hashtable是基于陈旧的Dictionary类的，HashMap是Java 1.2引进的Map接口的一个实现。第二个不同也许最重要的不同是Hashtable的方法是同步的，而HashMap的方法不是。这就意味着，虽然你可以不用采取任何特殊的行为就可以在一个多线程的应用程序中用一个Hashtable，但你必须同样地为一个HashMap提供外同步。一个方便的方法就是利用Collections类的静态的synchronizedMap()方法，它创建一个线程安全的Map对象，并把它作为一个封装的对象来返回。这个方法可以让你同步访问潜在的HashMap。这么做的结果就是当你不需要同步时，你不能切断Hashtable中的同步（比如在一个单线程的应用程序中），而且同步增加了很多处理费用。第三点不同只有HashMap可以让你将空值作为一个表的条目的key或value。HashMap中只有一条记录可以是一个空的key，但任意数量的条目可以是空的value。这就是说，如果在表中没有发现搜索键，或者如果发现了搜索键，但它是一个空的值，那么get()将返回null。如果有必要，用containsKey()方法来区别这两种情况。一些资料建议，当需要同步时，用Hashtable，反之用HashMap。但是，因为在需要时，HashMap可以被同步，HashMap的功能比Hashtable的功能更多，而且它不是基于一个陈旧的类的，所以有人认为，在各种情况下，HashMap都优先于Hashtable。关于Properties 有时候，你可

能想用一个hashtable来映射key的字符串到value的字符串。DOS、Windows和Unix中的环境字符串就有一些例子，如key的字符串PATH被映射到value的字符串C:\WINDOWS.C:\WINDOWS\SYSTEM。Hashtables是表示这些的一个简单的方法，但Java提供了另外一种方法。Java.util.Properties类是Hashtable的一个子类，设计用于String keys和values。Properties对象的用法同Hashtable的用法相象，但是类增加了两个节省时间的方法，你应该知道。Store()方法把一个Properties对象的内容以一种可读的形式保存到一个文件中。Load()方法正好相反，用来读取文件，并设定Properties对象来包含keys和values。注意，因为Properties扩展了Hashtable，你可以用超类的put()方法来添加不是String对象的keys和values。这是不可取的。另外，如果你将store()用于一个不包含String对象的Properties对象，store()将失败。作为put()和get()的替代，你应该用setProperty()和getProperty()，它们用String参数。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com