**JAVA 5以後用ConcurrentHashMap取代HashTable**

**Map aa=new ConcurrentHashMap();**

1. HashMap幾乎可以等價於Hashtable，除了HashMap是非synchronized的，並可以接受null(HashMap可以接受為null的鍵值(key)和值(value)，而Hashtable則不行)。
2. HashMap是非synchronized，而Hashtable是synchronized，這意味著Hashtable是執行緒安全的，多個執行緒可以共用一個Hashtable；而如果沒有正確的同步的話，多個執行緒是不能共用HashMap的。Java 5提供了ConcurrentHashMap，它是HashTable的替代，比HashTable的擴展性更好。
3. 另一個區別是HashMap的迭代器(Iterator)是fail-fast反覆運算器，而Hashtable的enumerator反覆運算器不是fail-fast的。所以當有其它執行緒改變了HashMap的結構（增加或者移除元素），將會拋出ConcurrentModificationException，但反覆運算器本身的remove()方法移除元素則不會拋出ConcurrentModificationException異常。但這並不是一個一定發生的行為，要看JVM。這條同樣也是Enumeration和Iterator的區別。
4. 由於Hashtable是執行緒安全的也是synchronized，所以在單執行緒環境下它比HashMap要慢。如果你不需要同步，只需要單一執行緒，那麼使用HashMap性能要好過Hashtable。
5. HashMap不能保證隨著時間的推移Map中的元素次序是不變的。

|  |  |  |
| --- | --- | --- |
|  | HashTable | HashMap |
| Synchronized | V(多線程能同時取用) | X(只能供單一線程使用) |
| Thread Safty | V | X |
| 效能 | X | V |
|  |  |  |
|  |  |  |
|  |  |  |