**JAVA集合类性能测试**

**一．Java集合类知识总结**：

1. **List集合总结**：

List继承了Collection,是有序的列表。实现类有ArrayList,LinkedList, vector,Stack等

1. ArrayList是基于数组实现的，是一个数组队列。实现了可变大小的数组，它允许所有元素，包括null，ArrayList没有同步，size(),isEmpty(),get(),set()方法运行时间为常数，但add（）方法开销为分摊的常数，添加n个元素需要O(n)的时间，其他的方法运行时间为线性。每个ArrayList实例都有一个容量（Capactity）,即用于存储元素的数据大小，这个容量可以随着不断的增加新元素而自动增加，
2. LinkedList是基于链表实现的，是一个双向循坏列表。可以被当做堆栈使用

LinkedList实现了List类接口，允许null元素，此外LinkedList提供额外的get,delete,insert方法在LinkedList的首部和尾部。这些操作使LinkedList可被用作堆栈（stack）,队列（queue）或双向队列（deque）。LinkedList没有同步方法，如果用多线程访问同一个List,则必须自己实现访问同步，一种解决办法是在创建List时构造一个同步的List

List list=Collection.synchronizedList(new LinkedList(…))

1. Vector是基于数组实现的，是一个矢量队列，是线程安全的。Vector非常类似ArrayList

当vector是同步由Vector创建的iterator是同一接口，但是，因为vector是同步的，当一个iterator被创建而且在被使用，另一个线程改变了Vector状态，这时调用iterrator的方法时将抛出ConcurrentModificationException,因此必须捕获该异常。

1. Stack是基于数组实现的，是栈，它继承于vector，特性是FILO(先进后出)
2. **Set接口：**

Set一种不包含重复元素的Collection,即任意的两个元素e1,e2都e1.equals(e2)=false,Set的构造函数有一个约束条件，传入的Collection参数不包含重复的元素。必须小心操作可变对象，如果一个Set中的可变元素改变了自身的状态导致Object.equals(Object)=true将导致一些问题

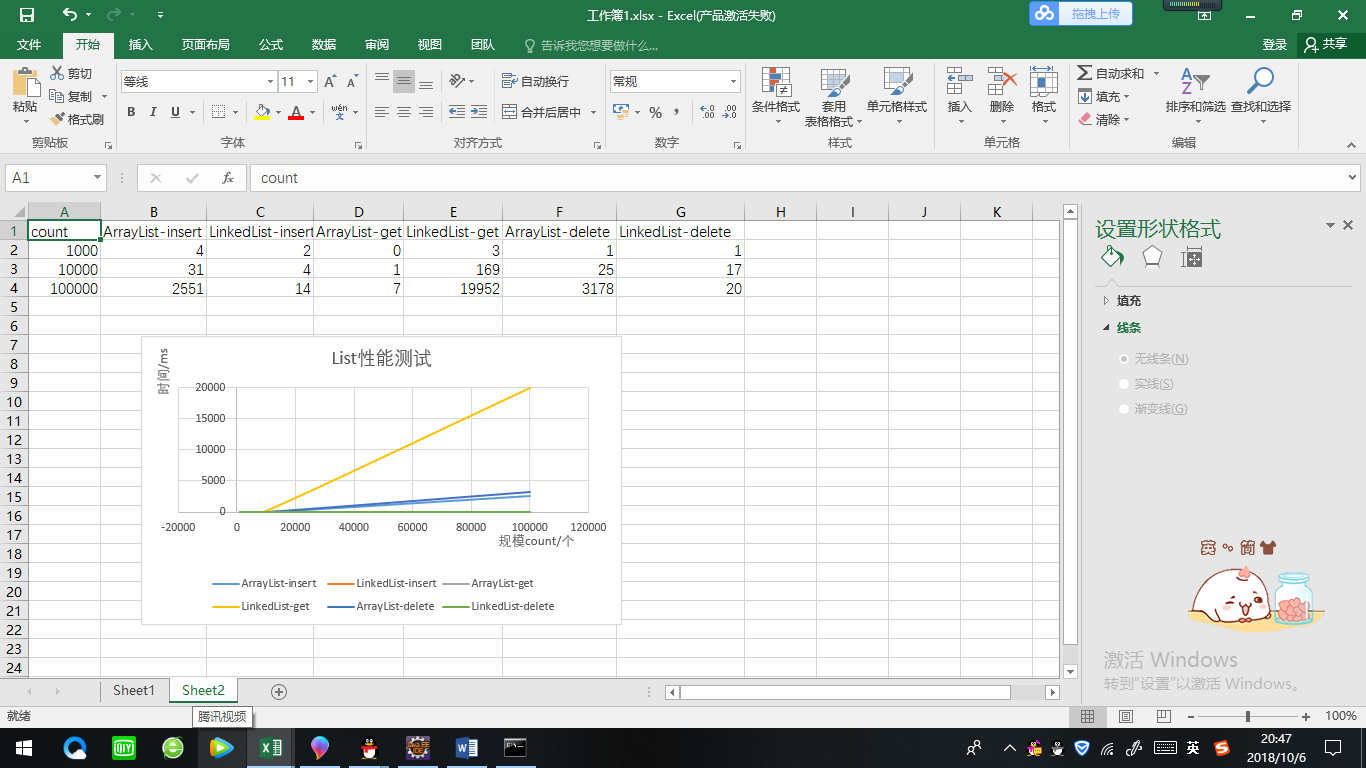
1. HashSet调用对象的hashCode(),获得哈希码，然后在集合中计算存放对象的位置，通过比较哈希码与equals()方法来判断是否重复，所以重载了equals()方法的同时也要重载hashCode()；
2. TreeSet继承sortedSet接口，能够对集合对象排序，默认排序方式是自然排序，但该方式只能实现Comparable接口的对象排序，
3. **Map接口：**

Map没有继承Collection接口，Map提供Key到value的映射，一个Map中不能包含相同的kay,每个key只能映射一个value。Map接口提供了3种集合的视图，Map的内容可以被当做一组key集合，一组value集合或一组key-value映射。

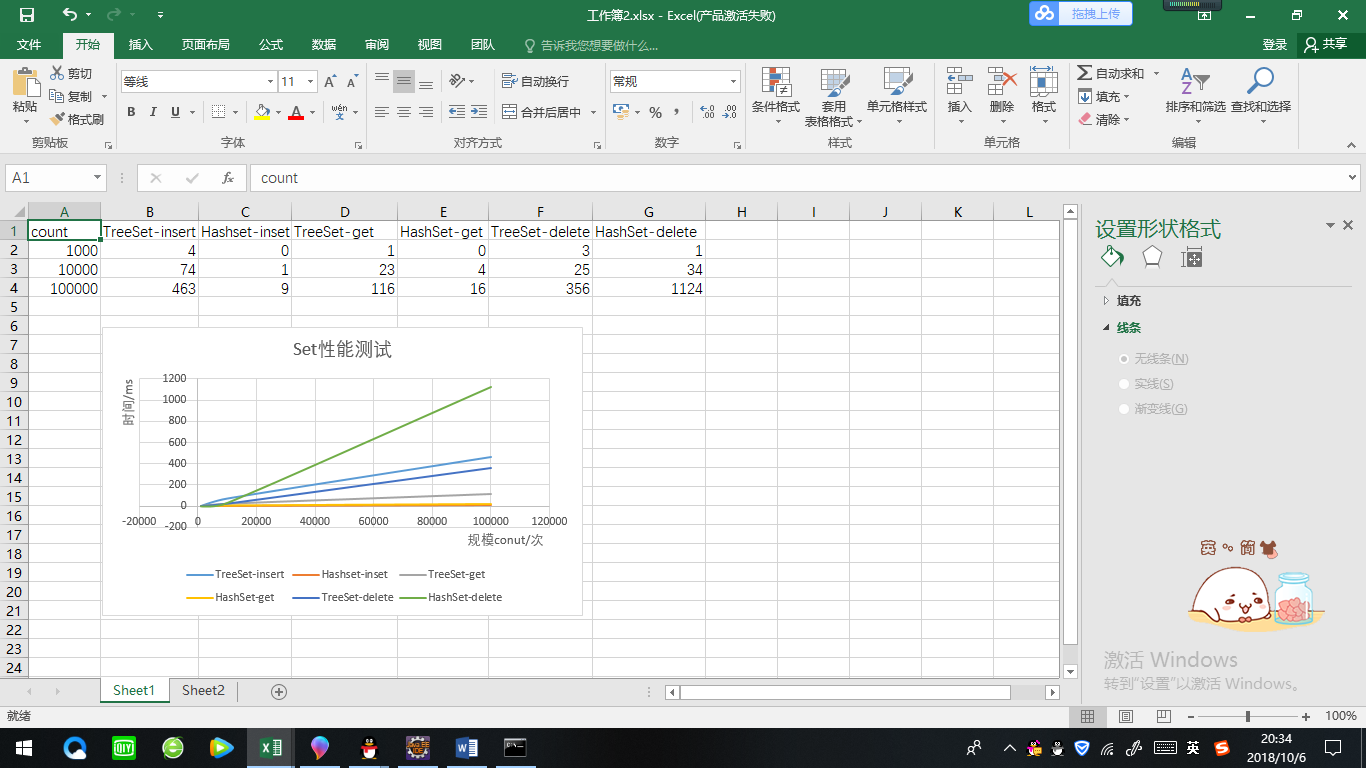
1. HashMap是非同步的，并且允许null，即null value和null key 但是将HashMap视为Collection时，其迭代操作时间开销和HashMap的容量成正比。
2. HashTable继承Map接口，实现了一个key--value映射的哈希表。任何非空的对象都可作为key或者value。添加数据使用put(key,value),取出数据使用get(key)，这两个基本操作的时间开销为常数。HashTable通过initial caoacity和load factor两个参数调整性能。通常缺省的load factor 0.75较好地实现了时间和空间的均衡。增大了load factor可以节省空间但相应的查找时间将增大，这回影响像get和put这样的操。HashTable是同步的，

二．**Java类集合性能测试结果**：

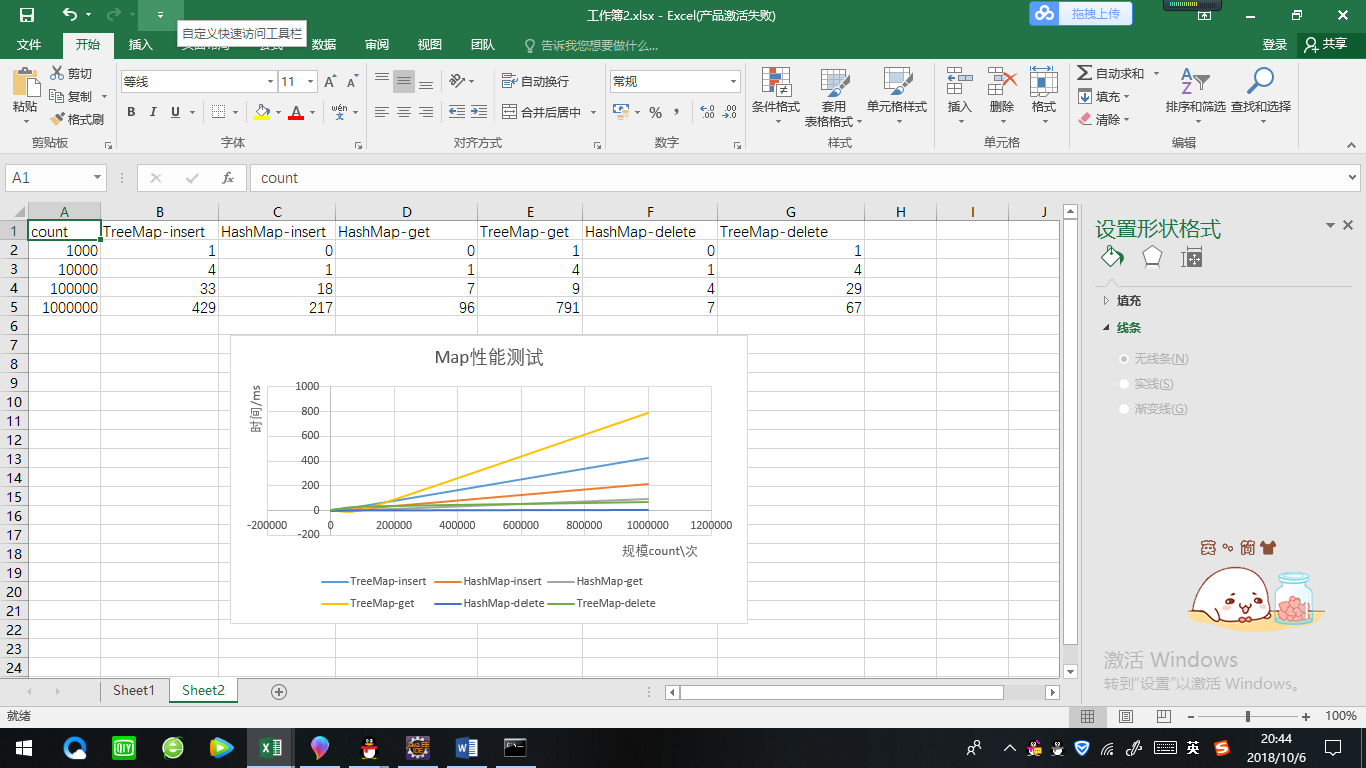
1. **集合类List性能测试：**



1. **集合类Set性能测试**

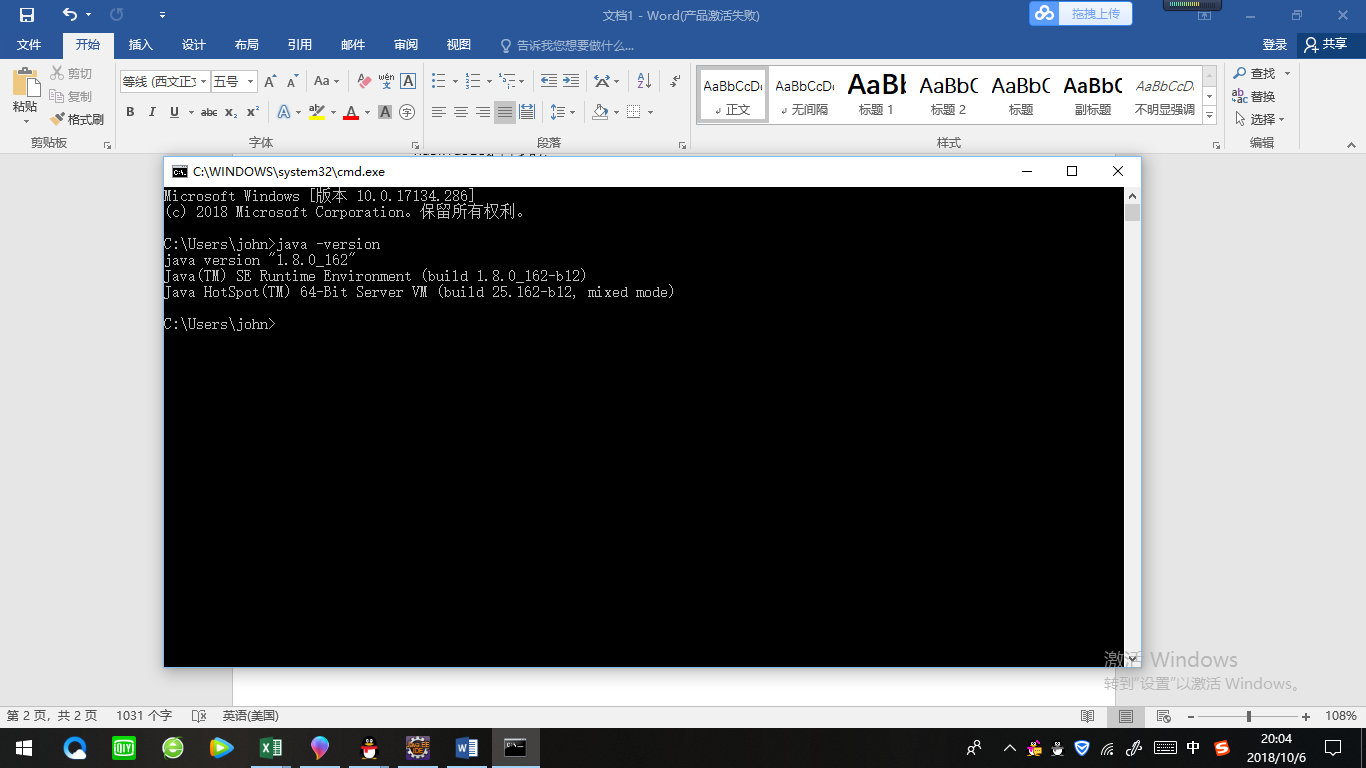


1. **集合类Map性能测试**



**1.硬件配备**：

（1）JDK版本



（2）CPU型号和内存

