Practical Exercise 5

Exercise Questions

1. Can you use a *for-each* loop to traverse the elements in any instance of `Collection`?

2. When using a for-each loop to traverse all the elements in a collection, do you need to use the `next()` or `hasNext()` methods in an iterator?

3. ***Use iterators on linked lists***
   Write and test a Java program that stores 5 million integers in a linked list and test the time to traverse the list using an *iterator* (*for-each loop*) vs. using the `get(index)` method (i.e. compare on running time).
   [Note that in Java, there is a method `System.currentTimeMillis()`, which returns the current time in milliseconds.]

4. Suppose that list1 is a list that contains the strings red, yellow and green, and that list2 is another list that contains the strings red, yellow and blue. Write and test a Java program to display the content of list1 and list after each of the following execution:
   1. list1.addAll(list2)
   2. list1.add(list2)
   3. list1.removeAll(list2)
   4. list1.remove(list2)
   5. list1.retainAll(list2)
   6. list1.clear()

5. How do you add a new element to a stack? How do you remove an element from a stack? How do you find the size of a stack?

6. Is java.util.Queue a subinterface of java.util.Collection, java.util.Set, or java.util.List? Does LinkedList implement Queue?

7. How do you create a priority queue for integer? By default, how are elements ordered in a priority queue? Is the element with the least value assigned the highest priority in a priority queue?

8. How do you create a priority queue that reverse the natural order of the elements?

9. (Perform set operations on priority queue)
   Create two priority queues:
   [“George”, “Jim”, “John”, “Blake”, “Kevin”, “Michael”] and
   [“George”, “Katie”, “Kevin”, “Michelle”, “Ryan”], and find their union, difference, and intersection.