CS 150 Topics List

Chapter 21

1. Sets (21.2)
   1. A collection that contains no duplicate items. That is, no two elements e1 and e2 can be in the set such that e1.equals(e2) is true.
   2. HashSet
      1. Use the element class’s hashCode() method to place items in the set. In other words, the element’s index (location) in the set is determined by its hash code.
      2. The details of making hash codes is not covered here.
      3. The hash codes of two objects must be the same if the two objects are equal.
      4. The hash codes of two unequal objects may be the same, but this should be avoided as much as possible.
   3. LinkedHashSet
      1. Uses a linked-list implementation that supports ordering the set.
      2. Elements in the set are sequenced based on the order in which they were added.
   4. TreeSet
      1. A non-hash set that implements SortedSet and NavigableSet.
      2. This causes the set to be stored in a sorted order
      3. Navigable allows for the use of lower(e), floor(e), ceiling(e), and higher(e).
      4. These sets can be set up to use Comparable or a Comparator for the sorting decisions.
2. Sets vs Lists (21.3)
   1. Sets are more complicated to implement and use more space in RAM than Lists, but they are more efficient to use when storing non-duplicate elements.
   2. Lists are useful for using indexes to access elements.
3. Maps (21.5)
   1. Maps are container objects that store key/value pairs called *entries*.
   2. Maps store both the list of values and the list of keys.
   3. Maps have update methods (clear, put, putAll, and remove)
   4. Maps have query methods (containsKey, containsValue, isEmpty, and size).
   5. Like Sets, Maps have three concrete subclasses that act in the same way, just with Maps instead of Sets:
      1. HashMap
      2. LinkedHashMap
      3. TreeMap
4. Singleton and Unmodifiable methods for Sets, Lists, and Maps. (21.7)