**Advanced Programming**

Java Collections

Agha Bazaid Khan

16L-4211

**ArrayList vs Vector**

|  |  |
| --- | --- |
| **ArrayList** | **Vector** |
| * ArrayList increments 50% of current array size if the number of elements exceeds from its capacity. * ArrayList is fast because it is **non-synchronized.** * ArrayList uses the Iterator interface to traverse the elements. * ArrayList is not a legacy class. It is introduced in JDK 1.2. | * Vector increments 100% means doubles the array size if the total number of elements exceeds than its capacity. * Vector is slow because it is synchronized, i.e., in a multithreading environment, it holds the other threads in runnable or non-runnable state until current thread releases the lock of the object. * A Vector can use the Iterator interface or Enumeration interface to traverse the elements. * Vector is a legacy class. |

**HashSet vs SortedSet**

|  |  |
| --- | --- |
| **HashSet** | **SortedSet** |
| * HashSet stores the elements by using a mechanism called hashing. * HashSet allows null value. * HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their hashcode. * HashSet is the best approach for search operations. | * SortedSet stores the elements by using a mechanism called sorting. * SortedSet does not allow null value. * The elements are ordered either by using a natural ordering or by using a Comparator. * SortedSet are reasonable approach for search operations. |

**HashSet vs TreeSet**

|  |  |
| --- | --- |
| **HashSet** | **TreeSet** |
| * HashSet is faster. * HashSet allows null value. * HashSet is backed by HashMap. * HashSet doesn't guarantee any order. | * TreeSet is relatively slower. * TreeSet doesn't allow null Object and throw NullPointerException. * TreeSet is backed by TreeMap in Java. * TreeSet maintains objects in the Sorted order defined by either Comparable or Comparator method in Java. |

**Array vs List**

|  |  |
| --- | --- |
| **Array** | **List** |
| * Array is collection of homogeneous elements. * For Array memory allocated is static and continuous. * User need not have to keep in track of next memory allocation. * Easier and faster to access. | * List is collection of heterogeneous elements. * For List memory allocated is dynamic and Random. * User has to keep in Track of next location where memory is allocated. * Not fast in accessing items. |

**List vs Set**

|  |  |
| --- | --- |
| **List** | **Set** |
| * List is an ordered collection it maintains the insertion order. * List allows duplicates. * List allows any number of null values. * List interface has one legacy class called Vector. | * Set is an unordered collection, it doesn’t maintain any order. * Set doesn’t allow duplicate elements. * Set can have only a single null value at most. * Set interface does not have any legacy class. |

**NavigableSet vs NavigableMap**

|  |  |
| --- | --- |
| **NavigableSet** | **NavigableMap** |
| * A NavigableSet extends the SortedSet interface and as well as NavigableMap interface. * The NavigableSet interface represents a Set that is sorted in terms of a client. * It associates to an unordered collection of distinct elements (i.e it doesn't store duplicates). | * A NavigableMap extends the SortedMap interface which represents a sorted map. * The NavigableMap represents a Map. * Associates its elements with certain keys so that these elements could be obtained by that keys. |