|  |  |
| --- | --- |
| List | Set |
| Insertion order is maintained | Insertion order is not maintained |
| Duplicate elements are allowed | Duplicate elements are not allowed |
| Null values are allowed | Null value is allowed only once |
| List contains classes   1. ArrayList 2. LinkedList 3. Vector 4. Stack | Set Contains Interfaces and classes   1. HashSet 2. LinkedHashSet 3. TreeSet 4. NavigableSet Interface 5. SortedSet Interface |

|  |  |
| --- | --- |
| ArrayList | LinkedList |
| Underlying Data structure of ArrayList growable and resizable array | Underlying Data Structure of LinkedList is double LinkedList; |
| ArrayList is best choice for Retrieval Operaton | LinkedList is best option for insertion and deletion |
| In arrayList no new methods are added, It uses collection and List interface methods | In LinkedList there are 6 new methods are added and also used Collection and list interface methods. |

|  |  |
| --- | --- |
| ArrayList | Vector |
| All methods are non-synchronized | All methods are synchronized |
| Performance is high | Performance is low |
| Multiple threads are allowed at the same time | Single thread is allowed at the same time |
| It is introduced in java 1.2 version | It is introduced in java 1.0 version |
| ArrayList is not legacy class | Vector is legacy class |

|  |  |
| --- | --- |
| LinkedList | Vector |
| All methods are non-synchronized | All methods are synchronized |
| Performance is high | Performance is low |
| Multiple threads are allowed at the same time | Single thread is allowed at the same time |
| Underlying Data structure is Double Linked List | Underlying dataStructure is growable and resizble array |
| ArrayList is not legacy class | Vector is legacy class |

|  |  |
| --- | --- |
| HashSet | LinkedHashSet |
| Underlying data structure is HashTable | Underlying Data structure is LinkedList and HashTable |
| It does not maintain insertion order | It maintains insertion order |
| It is implemented class of Set Interface | It is a child class of HashSet class |

|  |  |
| --- | --- |
| HashSet | TreeSet |
| Underlying data structure is HashTable | Underlying data structure is Balanced Tree |
| It does not maintain insertion order | It insert the values as per ascending order or alphabetical order |
| Heterogeneous values are allowed | Heterogeneous values are not allowed |
| Null value is allowed only once | Null value is not allowed |
| It is a implemented class of Set interface | It is an implemented class of NavigableSet Interface. |

|  |  |
| --- | --- |
| LinkedHashSet | TreeSet |
| Underlying data structure is LinkedList and HashTable | Underlying data structure is Balanced Tree |
| Heterogeneous values are allowed | Heterogeneous values are not allowed |
| Null value is allowed only once | Null value is not allowed |
| Insertion order is maintained | It insert the values as per ascending order or alphabetical order |
| It is child class of HashSet Class | It is implemented class of NavigableSet Interface |

|  |  |
| --- | --- |
| HashMap | HashTable |
| All methods are non-synchronized | All methods are synchronized |
| Performance is high | Performance is low |
| Multiple threads are allowed at the same time | Single thread is allowed at the same time. |
| Null key is allowed only once | Null key is not allowed |
| Null values are allowed multiple times | Null values are not allowed |
| Default capacity is 16 | Default capacity is 11 |
| It does not maintain insertion order | It insert values as per Hash Code |
| Child class is LinkedHashMap class | Child class is Properties Class |

|  |  |
| --- | --- |
| HashMap | LinkedHashMap |
| Insertion order is not maintained | Insertion order is maintained |
| Underlying data structure is HashTable | Underlying data structure is HashTable and LinkedList. |
| It is implemented class of Map Interface | It is child class of HashMap |

|  |  |
| --- | --- |
| HashMap | TreeMap |
| Underlying data structure is HashTable | Underlying data structure is Red-Black Tree |
| Insertion order is not maintained | It insert the Keys as per ascending or alphabetical order |
| It is implemented class of Map Interface | It is Implemented class of NavigableMap Interface |
| We can add heterogeneous Key-value | We can add Keys are homogeneous but values can be heterogeneous. |
| There is no new method is present is HashMap Class, We always use Map Interface methods | There is no new method present in TreeMap class, We can use methods from Map Interface, SortedMap Interface, NavigableMap Interface |

|  |  |
| --- | --- |
| HashTable | TreeMap |
| Underlying data structure is HashTable | Underlying data structure is Red-Black Tree |
| Default capacity is 11 | Default capacity is 16 |
| Insert the values as per Hash Code. | We can insert values as per ascending order or alphabetical order of Key. |
| All methods are synchronized | All methods are non-synchronized |
| Performance is low | Performance is high |
| Single thread is allowed at same time | Multiple threads are allowed at same time |
| We can add the heterogeneous key value | We can add keys are homogeneous but values are heterogeneous. |
| There is no new method is present is HashTable Class, We always use Map Interface methods. | There is no new method present in TreeMap class, We can use methods from Map Interface, SortedMap Interface, NavigableMap Interface |