Generics

Employee item

Push(Employee item)

Employee pop()

Integer item

Push(Integer item)

Integer pop()

String item

Push(String item)

String pop()

s i emp

SampleStack<String> s = new SampleStack<String>();

SampleStack<Integer> i = new SampleStack<Integer>();

x

Collection (Interface)

Set (Interface) List (Interface)

SortedSet (Interface)

HashSet LinkedHashSet TreeSet ArrayList Vector LinkedList

Set : Unordered collection, does not allow duplicates

HashSet : Unordered, internally uses hashing, best choice if the frequent operation is Search Operation

LinkedHashSet : maintains insertion order, internally uses doubly linked list

TreeSet : Sorted ( guarantees natural order i.e., ascending or alphabetical)

List : Ordered, allows duplicates

ArrayList : Resizable array

Best choice if frequent operation is retrieval (random access)

Worst choice if frequent operation is insertion or deletion in the middle (since internally requires several shift operations)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

1. 99

A[99] – directly accesses the last element

Vector : thread safe (contains synchronized methods), similar to ArrayList, can dynamically shrink & grow

LinkedList : underlying data structure is double linkedlist

used for implementing Stacks & Queues

Best choice if frequent operation is insertion or deletion in the middle

Worst choice if frequent operation is retrieval (sequential access)

Map (Interface)

SortedMap (Interface)

HashMap LinkedHashMap Hashtable TreeMap

Map : supports to work with key-value pairs

HashMap : Unordered, faster

LinkedHashMap : Ordered, internally uses doubly linked list, slower than HashMap

TreeMap : Sorted in natural order

Hashtable : thread safe (contains synchronized methods)