

Collection Framework and Maps

Assignment Solution

Question 1: What is the collection framework in Java?

Answer 1: The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects.

- Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.
- Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).
- The Collection framework represents a unified architecture for storing and manipulating a group of objects. It has:
 - Interfaces and its implementations, i.e., classes
 - Algorithm

Question 2: What is the difference between ArrayList and LinkedList?

Answer 2:

ArrayList	LinkedList
1) ArrayList internally uses a dynamic array to store the elements.	LinkedList internally uses a doubly linked list to store the elements.
2) Manipulation with ArrayList is slow because it internally uses an array. If any element is removed from the array, all the other elements are shifted in memory.	Manipulation with LinkedList is faster than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory.
3) An ArrayList class can act as a list only because it implements List only.	LinkedList class can act as a list and queue both because it implements List and Deque interfaces.
4) ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.
5) The memory location for the elements of an ArrayList is contiguous.	The location for the elements of a linked list is not contagious.

Question 3: What is the difference between Iterator and ListIterator?

Answer 3:

Iterator	ListIterator
----------	--------------

Can traverse elements present in Collection only in the forward direction.	Can traverse elements present in Collection both in forward and backward directions.
Helps to traverse Map, List and Set.	Can only traverse List and not the other two.
Indexes cannot be obtained by using Iterator.	It has methods like nextIndex() and previousIndex() to obtain indexes of elements at any time while traversing List.
Cannot modify or replace elements present in Collection	We can modify or replace elements with the help of set(E e)
Cannot add elements and it throws ConcurrentModificationException.	Can easily add elements to a collection at any time.
Certain methods of Iterator are next(), remove() and hasNext().	Certain methods of ListIterator are next(), previous(), hasNext(), hasPrevious(), add(E e).

Question 4: What is the difference between Iterator and Enumeration?

Answer 4:

Iterator	Enumeration
Iterator is a universal cursor as it is applicable for all the collection classes.	Enumeration is not a universal cursor as it applies only to legacy classes.
Iterator has the remove() method.	Enumeration does not have the remove() method.
Iterator can do modifications (e.g using remove() method it removes the element from the Collection during traversal).	Enumeration interface acts as a read only interface, one can not do any modifications to Collection while traversing the elements of the Collection.
Iterator is not a legacy interface. Iterator can be used for the traversal of HashMap, LinkedList, ArrayList, HashSet, TreeMap, TreeSet .	Enumeration is a legacy interface which is used for traversing Vector, Hashtable.

Question 5: What is the difference between List and Set?

Answer 5:

List	Set
1. The List is an indexed sequence.	1. The Set is an non-indexed sequence.
2. List allows duplicate elements	2. Set doesn't allow duplicate elements.
3. Elements by their position can be accessed.	3. Position access to elements is not allowed.
4. Multiple null elements can be stored.	4. Null element can store only once.
5. List implementations are ArrayList, LinkedList, Vector, Stack	5. Set implementations are HashSet, LinkedHashSet.

Question 6: What is the difference between HashSet and TreeSet?

Answer 6:

Parameters	HashSet	TreeSet
Ordering or Sorting	It does not provide a guarantee to sort the data.	It provides a guarantee to sort the data. The sorting depends on the supplied Comparator.
Null Objects	In HashSet, only an element can be null.	It does not allow null elements.
Comparison	It uses hashCode() or equals() method for comparison.	It uses compare() or compareTo() method for comparison.
Performance	It is faster than TreeSet.	It is slower in comparison to HashSet.
Implementation	Internally it uses HashMap to store its elements.	Internally it uses TreeMap to store its elements.
Data Structure	HashSet is backed up by a hash table.	TreeSet is backed up by a Red-black Tree.
Values Stored	It allows only heterogeneous value.	It allows only homogeneous value.

Question 7: What is the difference between Array and ArrayList?

Answer 7:

Base	Array	ArrayList
Dimensionality	It can be single-dimensional or multidimensional	It can only be single-dimensional
Traversing Elements	For and for each generally is used for iterating over arrays	Here iterator is used to traverse over ArrayList
Length	length keyword can give the total size of the array.	size() method is used to compute the size of ArrayList.
Size	It is static and of fixed length	It is dynamic and can be increased or decreased in size when required.
Speed	It is faster as above we see it of fixed size	It is relatively slower because of its dynamic nature
Primitive Datatype Storage	Primitive data types can be stored directly unlikely objects	Primitive data types are not directly added unlikely arrays, they are added indirectly with help of autoboxing and unboxing
Generics	They cannot be added here hence type unsafe	They can be added here hence making ArrayList type-safe.
Adding Elements	Assignment operator only serves the purpose	Here a special method is used known as add() method

Question 8: What is a Map in Java?

Answer 8: A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.

- A Map is useful if you have to search, update or delete elements on the basis of a key.
- Java Map Hierarchy: There are two interfaces for implementing Map in java: Map and SortedMap, and three classes: HashMap, LinkedHashMap, and TreeMap.

Question 9: What are the commonly used implementations of maps in java?

Answer 9: There are three commonly used implementations of maps in Java:

- **HashMap:** HashMap is the most commonly used implementation of the Map interface. It is an unordered collection that stores key-value pairs in a hash table. One of the benefits of using HashMap is that it allows for quick retrieval of values based on their keys.
- **TreeMap:** TreeMap is an implementation of the Map interface that stores key-value pairs in a sorted order. The keys are sorted based on their natural ordering or by a comparator provided at the time of creation.
- **LinkedHashMap:** LinkedHashMap is a hybrid implementation of the Map interface that maintains the elements in the order in which they were inserted. It provides constant-time performance for the basic operations, such as get and put.

Question 10: What is the difference between HashMap and TreeMap?

Answer 10:

Sr No	HashMap	TreeMap
1.	It does not provide any order for elements.	It provides orders for elements.
2.	It's speed is fast.	It's speed is slow.
3.	It allows one key as null and also allows multiple values.	It does not allow key as null but it allows multiple null values.
4.	It consumes more memory space.	It consumes less memory space.
5.	It has only basic features.	It has advanced features.
6.	For comparing keys, equals() is used.	For comparing keys, compare or compareTo() is used.
7.	It's complexity is O(1).	It's complexity is O(log n).

Question 11: How do you check if a key exists in a map in Java?

Answer 11: To check if a key exists in a map in Java, you can use the containsKey() method. The containsKey() method takes a key as a parameter and returns true if the key is present in the map, or false if it is not.