. Draw Collections Framework Class Diagram

Collections

List Set Shored List

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

Abstract List Abstract Set

Concrete Class Concrete Class

ArrayList Vector HashSet TreeSet

LinkedList Stack LinkedHashSet

----------------------------------------------------------

Queue Map Shorted Map

Abstract List

Concrete Class Navigable Map

Priority Queue LinkedMap

Priority Blocking Queue LinkedHashMap

Blocking Queue TreeMap

2. What is HashMap and Map?

HashMap is concrete class extends abstract class implements map

HashMap contains values based on the key. Contains only unique elements. It may have one null key and multiple null values.It maintains on order.

Import java.util.HashMap

Map is an interface and has HashMap , LinkedHashMap and TreeMap. Map contains values on the basis of Key value pair.Each key and value pair is known as an entry. Map contains only unique keys.

3.Difference between HashMap and HashTable? Can we make hashmap synchronized?

HashMap: HashMap is non synchronized.It is not thread safe and can`t be shared between many threads without proper synchronization code

HashMap allows one null key and multiple null values

HashMap is fast

HashMap inherits abstractmap class

Hashtable:Hashtable is synchronized. It is thread safe and can be shared with many threads.

Hashtable doesn`t allow any null key or value.

Hastable is slow.

4.Difference between Vector and ArrayList?

Vector:Vector is synchronized and increments 100% and doubles the array size,vector is slow because it is synchronized

ArrayList:Arraylist is not synchronized and arraylist increments 50% of current array size.Arraylist is fast because it is non-synchronized.

5.What is an Iterator.

Iterator is an interface which belongs to collection API.It allows us to traverse the collection,access the data element and remove the data elements of the collection.

It has public interface Iterator and had three methods.

hasNext: it returns true if iterator has more element to iterate.

Next(): it returns the next element in the clollection until the hasNext()return true.  Until there is no next element.

Remove(): It removes the current element in the collection. This method throuws ‘IllegaStateException’ if this function is called before next() is invoked.

6.List vs Set vs Map. Purposes and definitions.

         Set, List and Map are three important interfaces of Java collection framework and Difference between Set, List, and Map in Java is one of the most frequently asked [Java Collection interview question](http://java67.blogspot.com/2012/09/java-collection-interview-questions.html). Sometimes this question is asked as When to use List, Set and Map in Java. Clearly, the interviewer is looking to know that whether you are familiar with fundamentals of Java collection framework or not. In order to decide when to use List, Set or Map, you need to know what are these interfaces and what functionality they provide. [List in Java](http://java67.blogspot.com/2012/07/sort-list-ascending-descending-order-set-arraylist.html) provides ordered and indexed collection which may contain duplicates

7.Pros and cons of ArrayList and LinkedList

**Advantages of using arrays:**

* Easier to use and access
* Faster access to the elements

**Disadvantages of using arrays:**

* Fixed size - the size of the array is static
* One block allocation - if you don't have enough memory to provide a single block (but you have sufficient scattered memory blocks) to allocate the space for the array then you'll need to defragment and other similar stuff to first create a free block of that size. So you may like to term it as improper utilization of memory :-)
* Complex position-based insertion - if you want to insert an element at a position already covered by some other element then you got to shift right by one position all the elements to the right of that position. This will vacate the position for you to insert the new element at the desired position. The more elements you have to the right of the desired position, the more expensive the process will be.

**Advantages of using Linked Lists:**

* Flexibility - insert at (or delete from) any position in contant time
* No single allocation of memory needed - fragmented memory can be put to a better use
* Dynamic allocation - the size is not required to be known in advance

**Disadvantages of using Linked Lists:**

* Complex to use and access - relatively complex as compared to arrays
* No constant time access to the elements - simply because it doesn't involve the simple arithmetic used by arrays to compute the memory address, so relatively inefficient as compared to arrays

8.TreeSet vs LinkedHashSet

**TreeSet**, **LinkedHashSet** and HashSet in Java are three Set implementation in collection framework and like many others they are also used to store objects. **TreeSet** is a SortedSet implementation which allows it to keep elements in the sorted order defined by either Comparable or Comparator interface.

9.What are relationships between equals and hash codes?

Object class provides two methods hashcode() and equals() to represent the identity of an object. It is a common convention that if one method is overridden then other should also be implemented.

Before explaining why lets see what is the contract between these two methods holds. As per the Java API documentation:

* When hashcode() is invoked on the same object more than once during an execution of a Java application, the hashcode() method must consistently return the same integer, provided no information used in equals() comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application.
* If two objects are equal according to the equals(object) method, then calling the hashCode() method on each of the two objects must produce the same integer result.
* It is NOT required that if two objects are unequal according to the equals(Java.lang.Object) method, then calling the hashCode() method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

10.What are the advantages of ArrayList over arrays ?

The major difference between the two is that arrays are fixed length data **structure**. But, ArrayList is variable length, which means ArrayList can grow or shrink its size dynamically. ArrayList is backed by an array. ArrayList can roughly increment its size by 1.5 times.

11.Principle of storing data in a  hashtable

In [computing](https://en.wikipedia.org/wiki/Computing), a **hash table** (**hash map**) is a [data structure](https://en.wikipedia.org/wiki/Data_structure) that implements an [associative array](https://en.wikipedia.org/wiki/Associative_array) [abstract data type](https://en.wikipedia.org/wiki/Abstract_data_type), a structure that can map [keys](https://en.wikipedia.org/wiki/Unique_key) to [values](https://en.wikipedia.org/wiki/Value_(computer_science)). A hash table uses a [hash function](https://en.wikipedia.org/wiki/Hash_function) to compute an *index* into an array of *buckets* or *slots*, from which the desired value can be found.

Ideally, the hash function will assign each key to a unique bucket, but most hash table designs employ an imperfect hash function, which might cause hash [*collisions*](https://en.wikipedia.org/wiki/Collision_(computer_science)) where the hash function generates the same index for more than one key. Such collisions must be accommodated in some way.

12.Differences between Hashtable, ConcurrentHashMap and Collections.synchronizedMap()

13.How are hash codes computed?

14. Is it possible that hashcode is not unique?

15. Can we put two elements with equal hash code to one hash map?

1. It is perfectly legal for two objects to have the same hashcode.
2. If two objects are equal (using the equals() method) then they have the same hashcode.
3. If two objects are not equal then they cannot have the same hashcode  
     
      
     
      
     
   16.Iterator and modification of a List. ConcurentModificationException.  
     
      
     
   java.util.ConcurrentModificationException is a very common exception when working with java collection classes. Java Collection classes are fail-fast, which means if the Collection will be changed while some thread is traversing over it usingiterator, the iterator.next() will throwConcurrentModificationException  
     
      
     
   17. What is   the significance of ListIterator? What Is the difference b/w Iterator and ListIterator?  
     
   The basic difference between Iterator and ListIterator is that both being cursor,Iterator can traverse elements in a collection only in forward direction. On the other hand, the ListIterator can traverse in both forward and backward directions. Usingiterator you can not add any element to a collection.  
     
    18.What is the Collections API?

Collections in java is a framework that provides an architecture to store and manipulate the group of objects.

All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion etc. can be performed by Java Collections.

Java Collection simply means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet etc).

19.How can we access elements of a collections?

Elements of a collections can be access by using

Iterator Interface

Using Listiterator interface

Using for-each loop

20.What is   the difference between a queue anda stack?

|  |  |  |
| --- | --- | --- |
| # | STACK | QUEUE |
| 1 | Objects are inserted and removed at the same end. | Objects are inserted and removed from different ends. |
| 2 | In stacks only one pointer is used. It points to the top of the stack. | In queues, two different pointers are used for front and rear ends. |
| 3 | In stacks, the last inserted object is first to come out. | In queues, the object inserted first is first deleted. |
| 4 | Stacks follow Last In First Out (LIFO) order. | Queues following First In First Out (FIFO) order. |
| 5 | Stack operations are called push and pop. | Queue operations are called enqueue and dequeue. |
| 6 | Stacks are visualized as vertical collections. | Queues are visualized as horizontal collections. |
| 7 | Collection of dinner plates at a wedding reception is an example of stack. | People standing in a file to board a bus is an example of queue. |

21. What is   the Properties class?

Properties is a subclass of Hashtable. It is used to maintain lists of values in which the key is a String and the value is also a String. The Properties class is used by many other Java classes. For example, it is the type of object returned by System.getProperties( ) when obtaining environmental values.

22. Which implementation of   the List interface provides for the fastest insertion of a new element into the middle of the list?

23. How can we use hashset in collection interface?

Java HashSet class is used to create a collection that uses a hash table for storage. It inherits the AbstractSet class and implements Set interface. The important points about Java HashSet class are: HashSet stores the elements by using a mechanism called hashing.

25. Can you limit the initial capacity Of vector in java?

A vector has its default capacity which is 10 elements, here size is different from capacity, after 10 element if we enter one element the capacity of vector changes to 20 were as size is 11 only, for example if you have entered 21 elements in a vector, then if you print v.size it results 21 but v.capacity it results 30 If you have any further queries pls let me know

26.  What method should the key class Of Hashmap override?

In order to be used as a HashMap key the class has to implement these two methods to reflect "equality" of two objects. ... and expect them to be equal and work in a HashMap , you have to override hashCode() so that it returns the same value in both instances, and equals() returns true when comparing

27. What is the difference between Enumeration and Iterator?

     Enumuration and Iterator are two interfaces in java.util package.Using enumeration , you can only traverse the collection object, but using Iterator you can also remove an element while traversing the collection.

Iterator : hasNext(), next(), remove()

Enumeration : hasMoreElements(), hasElement(),

28. Collections class and Arrays class

                Collections Class: class consists exclusively of static methods that operate on or return collections.

                It contains polymorphism algorithms that operate on collections

                The methods of this class all throw a NUllPointerException.

                Array Class : contains a static factory that allows arrays to be viewed as list.

                                This class contains various methods for manipulating arrays

                                The methods in this class throw a NullPointerException