Beaconfire Inc, Home Work, Week1 Day4.
By (Ping) Nalongsone Danddank.
ndanddank@gmail.com
wechatID: ndanddank
Short Answer:

1. Describe the Collections Type Hierarchy. What Are the Main Interfaces, and What Are the Differences Between Them?

-> Iterable interface is the top above all Collections interface, all interface like List, Queue, Set, are inheritance from it. And collection's classes implement to different interface follow by specified they are, like Ordered or not ordered, duplicated or unique, LinkedList or Array data structure.

2. What are List interface implementations and what are the differences between them and when to use what?

-> List interface implement from Collection interface, and classes that implement from List is ArrayList, LinkedList, Vector and Stack, these classes there are similar with Ordered and Duplicated element, but there are different storing in memory address, like ArrayList is consequence and arrange, we can access the element with index and contains runtime, but insert and delete element operation may are linear time . Another hand, LinkedList use pointer to link or connect each element, so it is linear time for get or access a element, but contains time for insert and delete.

3. What are Queue interface implementations and what are the differences and when to use what?

-> Queue is First In First Out data structure, that mean the classes implement from it, are Ordered, and focus on append element to end and take element from the head. One interface implement from it is Deque, and PriorityQueue class also implement too. However, There are some classes that implement from Deque interface are LinkedList and ArrayQue. PriorityQueue uses Tree Heap data structure to keep sorted Ordered element. LinkedList stores element by link each node by pointer and ArrayDeque is array that access by index.

4. What are Set interface implementations and what are the differences and when to use what?

-> Set is like a bucket to put element to it, and each element must be unique. So if we don't care about Ordering element or not, using Hashset class which implement from Set interface. LinkedHashSet is for store unique elements and also ordered when we access. And SortedSet interface is sorted ordered element sequence data structure and the TreeSet class implement from it.

5. Explain the structure of the Deque implementation of LinkedList and its usage.

-> Deque similar to Queue data structure, but support insert and remove both head and end node. The LinkedList that implement from it, is focus on adding or removing element for both head and tail.

6. What are the differences between HashMap, LinkedHashMap and TreeMap?
-> The difference of them is when we store or insert the key to them, HashMap use HashSet, so when insert, the order is not preserved. But LinkedHashMap, the insertion order is preserved because use Sort, and TreeMap's key use TreeHashSet class, so the element that store in TreeMap will be sorted ordering.

7. What is the hashCode() and equals() function?
-> these function are from Object class. hashCode() is return integer, and is use for identity for object instance of class, each object should have identity hash code. equals() is use for comparing to other object instance, the function should use hash Code to implement two object equal or not. And if 2 object equal by equals() function that mean their hash code mush same, but when there are same hash code, there may equal.

8. How Is HashMap Implemented in Java? How Does Its Implementation Use hashCode and equals Methods of Objects? What Is the Time Complexity of Putting and Getting an Element from Such Structure?
-> HashMap use HashSet to store keys by using hash function to calculate the index of key that store in the array data structure, the key must me unique, and when the storing process of the keys, the hash function provide the same index, then using Linked List data structure to store the key, and runtime for get and put for the key should be constrain time. However, from Java 8 version up when number of the values LinkedList to large, storing the keys could change to be Binary Search Tree, and time complexity for access is log(n).

9. What is Comparable and Comparator interface? What are differences between them and how to use them?
-> Comparable is for class that need the own instance object compare to other instance. And Comparator use for implementation by a comparator class that use for comparing two other instance objects.

10. What is Functional Interface? How do you create your own Functional Interface?
-> is an interface with only one abstract method. So we can create our own Functional Interface by defining the single abstract method.

11. What is Lambda Expression? Why does Lambda Expression work so closely with Functional Interface?
-> is a short form and replacement for an anonymous class. And it simplify the use of functional interface that declare single abstract method. So let us convenience to write a code on the place or block instead of create a new interface or class.