Assignment # 20

Q1) What are the Limitations of Arrays?

Limitations with array:

- o Array is a homogeneous data structure i.e only one type of elements can be stored.
- o Array is static in nature i.e size of an array cannot be modified.
- o Insert and delete operations require more shifting of elements.
- etc.

Q2) What is Legacy class?

Java vendor has given some classes and interfaces before Java 2 to solve the limitiations of an array. These classes are called as legacy classes.

Q3) What are the Legacy classes and interfaces available?

Following are the 5 legacy classes available:

1. Stack
2. Vector
3. Dictionary
4. Hashtable
5. Properties

Q4) What is Legacy Cursor?

Enumeration is a legacy cursor that is used to access data from legacy classes one by one.

Q5) What are the limitations with Legacy classes?

Limitations with legacy classes:

- o There is no specific scenario defined that in which condition which classes can be used.
- o Almost all the methods defined in the legacy classes are synchronized. Because of this, the same object cannot be accessed by multiple threads concurrently.

Q6) What is Collection Framework?

- o Java Vendor has provided some new API (known as Collection Framework API) from Java 2 to solve the problems associated with Legacy classes.
- o Collection Framework provides mechanism to store and manipulate the group of elements.

Q7) What are the advantages of using Collection Framework?

The advantages of using the Collection Framework are:

- o Interfaces
- o Implementation Classes
- o Algorithms implementations
    - o inserting, deleting, searching, sorting and shuffling etc

1. Reduces programming effort
2. Improves performance
3. Increases flexibility
4. Better support for large datasets
5. Code reusability
6. Easier debugging
7. Improved code readability
8. Support for generics
9. Extensive API
10. Backward compatibility

Q8) What are the top level interfaces available in Collection Framework?

Following are the top level interfaces available in Collection Framework API:

- o Collection
- o Map
- o Iterator

Q9) What is the difference between add() and addAll() method?

Add() :

- add method adds one elements to the collection
- Returns a boolean value indicating whether the element was added successfully.

AddAll() :

- adds all the elements from the specified collection to the current collection.
- Returns a boolean value indicating whether any of the elements were added successfully.

Q10) What is the difference between removeAll() and retainAll() method?

removeAll() : it removes all the elements which are present in the specified collection from the current collection.

retainAll () : it retains all the elements all the elements which are present in the specifoed collection from the current collection and removes remaining elements

Q11) What is List in Java?

List is an interface available in java.util package and it is extending java.util.Collection

interface.

Q12) Can I add duplicate elements in List subclasses?

Yes, you can add duplicate elements in List subclasses, such as ArrayList, LinkedList, etc.

Q13) What are the concrete subclasses of List interface?

List interface has following concrete sub classes:

o ArrayList

o LinkedList

o Vector

o Stack

Q14) What is the extra functionality provided in List subclasses?

List subclasses provide extra functionality as follows:

- **ArrayList**: Fast random access and dynamic resizing.
- **LinkedList**: Efficient insertions/deletions and additional methods from Deque and Queue.
- **Vector**: Thread safety through synchronized methods.
- **Stack**: LIFO operations with methods like push(), pop(), and peek().

Q15) Why ArrayList is not recommendable for insert and delete operations?

It is fast to access the elements, but slow to insert and delete the elements.

Q16) What is the use of Stack class?

The `Stack` class is used to implement a **Last In, First Out (LIFO)** structure. It allows you to add elements with `push()`, remove the top element with `pop()`, and view the top element with `peek()`. It's useful for tasks like undo operations, expression evaluation, and depth-first search.

Q17) What are the differences between Array and ArrayList?

Array	ArrayList
It is programming construct to store/represent the colletion of elements.	It is a class introduced from Java 2 to manage the collection of elements.
Array size is static (fixed length).	ArrayList is dynamic (variable length).
Need to specify the size while creating the array object.	No need to specify the size while creating the ArrayList object.
Need to specify the type of the elements while creating the array object.	No need to specify the type of element. (From Java 5 using GENERICS, you can restrict type of element also.)
Primitives can be stored in array	Primitives can't be stored in ArrayList.
Need to use length property to get the number of elements.	Need to use size() method to get the number of elements.
Need to use index notation for managing elements.	Need to use methods for managing the elements.
Use Arrays class methods to perform the operations.	Use Collections class methods to perform the operations.

Q18) What are the differences between ArrayList and LinkedList?

ArrayList	LinkedList
Elements of an ArrayList will be stored internally using index representation.	Elements of an LinkedList will be stored internally using node representation.
ArrayList elements can be accessed randomly because it implements RandomAccess Interface.	LinkedList elements can not be accessed randomly because it doesn't implement RandomAccess Interface.
Insertion and deletion operations are expensive.	Insertion and deletion operations are inexpensive.
It occupies less memory.	It occupies more memory.

Q19) What are the differences between ArrayList and Vector?

Vector	ArrayList
It is legacy class reimplemented in Collection Framework.	It is a class introduced from Java 2 in Collection Framework.
Methods are synchronized.	Methods are non synchronized
Object can't be accessed by multiple threads concurrently.	Object can be by multiple threads concurrently.
It is thread safe.	It is not thread safe.
You can use Enumeration, Iterator, ListIterator to access the data	You can't use Enumeration. Only Iterator and ListIterator can be used to access the data.

Q20) What is Enumeration?

- Enumaration is an interface available in java.util package.
- It is a cursor to access the elements from legacy classes one by one.

Q21) How can I get the instance of Enumeration interface?

You can get an instance of the Enumeration interface by calling the elements() method on a Vector or the keys() method on a Hashtable.

Q22) What are the methods of Enumeration interface?

Enumeration interface contains following methods:

1. boolean hasMoreElements():

   Checks whether next element is available or not.

2. Object nextElement():

   Moves the pointer and returns elements from collection.

Q23) What are the Collection classes can use Enumeration to access elements?

The collection classes that can use Enumeration to access elements are:

1. **Vector**
2. **Hashtable**
3. **Stack** (which extends Vector)
4. **Properties** (extends Hashtable)

Q24) What is Iterator?

- Iterator is an interface available in java.util package.
- It is a cursor to access the elements from Collection subclasses one by one.Q25) How can I get the instance of Iterator interface subclass?

Q26) What are the methods of Iterator interface?

Iterator interface contains following methods:

1. boolean hasNext():

   Checks whether next element is available or not.

2. Object next() :

   Moves the pointer and returns elements from collection.

3. void remove():

   Removes the current elements from Collections.

Q27) What are the Collection classes can use Iterator to access elements?

The collection classes that can use Iterator to access elements include:

1. ArrayList
2. LinkedList
3. HashSet
4. TreeSet
5. PriorityQueue
6. Vector
7. HashMap

8. TreeMap

Q28) What are the differences between Iterator and Enumeration?

Enumeration	Iterator
Enumeration is the Legacy interface.	Iterator is Collection Framework interface.
Enumeration is used to access the elements from the Legacy class object (Vector, Stack, Hashtable etc).	Iterator is used to access the elements from the Collection Framework classes (ArrayList, LinkedList, HashSet, LinkedHashSet etc).
You can't remove the elements by using the Enumeration.	You can remove the elements from the collection by using the Iterator.
You can modify the collection using collection object while accessing the data using Enumeration.	You can not modify the collection using collection object while accessing the data using Iterator. (java.util.ConcurrentModificationException)

Q29) What is ListIterator?

ListIterator is an interface which is extending Iterator interface.

Q30) What are the methods of ListIterator interface?

The methods of the `ListIterator` interface in Java are:

1. **add(E e)**: Inserts an element at the current position.
2. **hasNext()**: Checks if there are more elements in the forward direction.
3. **hasPrevious()**: Checks if there are more elements in the backward direction.
4. **next()**: Returns the next element and advances the iterator.
5. **previous()**: Returns the previous element and moves the iterator back.
6. **nextIndex()**: Returns the index of the next element, or the size if at the end.
7. **previousIndex()**: Returns the index of the previous element, or -1 if at the start.
8. **remove()**: Removes the last element returned by the iterator.
9. **set(E e)**: Replaces the last element returned with the specified element.

Q31) How can I get the instance of ListIterator interface subclass?

You can get an instance of the ListIterator interface by calling the listIterator() method on a List implementation, like ArrayList or LinkedList.

Q32) What are the Collection classes can use ListIterator to access elements?

The collection classes that can use ListIterator to access elements include:

1. **ArrayList**
2. **LinkedList**
3. **Vector**
4. **Stack**

Q33) What are the differences between Iterator and ListIterator?

Iterator	ListIterator
It is used to access the elements from the subclasses of Collection interface.	It is used to access the elements from the subclasses of List interface only.
Elements can be accessed only in forward direction.	Elements can be accessed both in forward and reverse direction.
Elements can be accessed only once from first to last.	Elements can be accessed multiple times.
Elements of collection can not be replaced using Iterator.	Elements of collection can be replaced using ListIterator.
New element can not be added to collection using Iterator.	New element can be added to collection using ListIterator.
Index of an element can not be accessed using Iterator.	Index of an element can be accessed using ListIterator.
You can access the elements of collection from beginning only.	You can access the elements of collection from beginning or from any required location.

Q34) How can I access the elements in reverse order from List subclasses?

You can access the elements in reverse order from List subclasses by:

1. Using ListIterator and calling hasPrevious() and previous() methods.
2. Using descendingIterator() method (if available).
3. Reversing the list using Collections.reverse() and then iterating over it.

Simplest way:

```
List<String> list = new ArrayList<>();
// add elements to the list
for (int i = list.size() - 1; i >= 0; i--) {
   String element = list.get(i);
   // process element
}
```

Q35) How can I access the elements from the specified index from List subclasses?

You can access the elements from the specified index from List subclasses by using the get(int index) method.

Example:

```
List.get(5); // access the element at index 5
```

Q36) What is reason of ConcurrentModificationException?

"Modifying a collection (adding or removing elements) while iterating over it with an Iterator, Enumerator, or foreach loop."

Q37) What is Set in Java?

Set is an interface available in java.util package and it is extending java.util.Collection interface.

Q38) Can I add duplicate elements in Set subclasses?

No, you cannot add duplicate elements in Set subclasses

Q39) What are the concrete subclasses of Set interface?

Set interface has following concrete sub classes:

o HashSet

o LinkedHashSet

o TreeSet

Q40) What are the extra methods provided in Set interface?

Set interface does not have any extra methods.

Q41) How hashCode() and equals() method is used in Set subclasses?

In Set subclasses

- hashCode() is used to determine the index at which an object should be stored.

- equals() is used to check if an object is already present in the Set, to avoid duplicates.

If two objects have the same hashCode() and equals() returns true, the Set will consider them as duplicates and only store one of them.

Q42) What is the use of compareTo() method in TreeSet?

compareTo() method is used to identify the object uniquely and to manage the order of elements in TreeSet.

Q43) What is the use of compare() method in TreeSet?

The compare() method in TreeSet is used to compare two objects and determine their order in the set, allowing for custom sorting and duplicate prevention.

Q44) How to add custom objects in TreeSet?

To add custom objects in TreeSet ,you need to:

1. Implement the Comparable interface in your custom class.

2. Override the compareTo() method to define the sorting order.

Q45) How to add StringBuilder objects in TreeSet?

You can't directly add StringBuilder objects to a TreeSet. But you can use a wrapper class that implements Comparable and holds a StringBuilder object.

Q46) What is NavigableSet?

- NavigableSet is an interface added in Java 6.
- It is subtype of SortedSet.

Q47) What are the differences between List and Set?

List	Set
List can hold duplicate elements.	List can not hold duplicate elements.
Both Iterator and ListIterator can be used to access the elements of List.	Only Iterator can be used to access the elements of Set.
You can add the elements to the List at the required location.	You can not add the elements to the Set at the required location.
You can remove the elements from the List from the required location.	You can not remove the elements from the Set from the required location.
You can access the elements of List using index.	You can not access the elements of Set using index.
List does not have any subclasses to store the elements in sorted order.	Set has a subclass called TreeSet to store the elements in sorted order.
hashcode() or equals() methods are not used while storing the elements internally.	hashcode() or equals() methods are used while storing the elements internally to avoid duplicacy.

Q48) What are the differences between HashSet and TreeSet?

HashSet	TreeSet
Elements in the HashSet are unordered (Order of elements in the HashSet is based on hashcode values.)	Elements in the TreeSet are in sorted order. (Order of elements in the TreeSet is based on result of compareTo() method)
HashSet can hold different types of elements.	TreeSet can hold only same types of elements. (Because elements must be Comparable.)
Null value can be stored in HashSet.	Null value can not be stored in TreeSet.

Q49) What are the differences between HashSet and LinkedHashSet?

HashSet	LinkedHashSet
Elements in the HashSet are unordered (Order of elements in the HashSet is based on hashcode values.)	Elements in the LinkedHashSet are in the order added by user.

Q50) What is the use of Queue interface?

The Queue interface is used to implement a First-In-First-Out (FIFO) data structure, allowing you to:

- Add elements to the end (enqueue)
- Remove elements from the front (dequeue)
- Check the next element (peek)

Q51) Which version of Java has introduced Queue concept?

Queue is the newly added interface in Java 5.

Q52) What are important subclasses of Queue?

Java Collections has 3 classes for Queues use-cases.

a) ArrayDeque

b) LinkedList

c) PriorityQueue

Q53) Can I add null value in PriorityQueue?

No, you cannot add a null value to a PriorityQueue.

Q74) How can I store custom objects in PriorityQueue?

To store custom objects in a PriorityQueue , you need to:

1. Implement the Comparable interface in your custom class.
2. Override the compareTo() method to define the ordering.

Q65) What are the important methods of PriorityQueue?

Here are the important methods of PriorityQueue in Java:

1. add(element) - adds an element to the queue
2. offer(element) - adds an element to the queue (similar to add)
3. poll() - removes and returns the highest priority element
4. peek() - returns the highest priority element without removing it
5. size() - returns the number of elements in the queue
6. isEmpty() - checks if the queue is empty
7. remove(element) - removes a specific element from the queue
8. contains(element) - checks if the queue contains a specific element

Q66) What is ArrayDeque?

- ArrayDeque is a latest class from java.util package.
- ArrayDeque is implemented with Arrays.

Q67) What is LinkedList?

- LinkedList is the class from java.util package.
- LinkedList is implemented with Nodes.