Name: Shreemathi R N

Designation: Trainee

Domain: Testing

Java Task

1. Dictionary in Java

In Java, Dictionary is the list of key-value pairs. We can store, retrieve, remove, get, and put values in the dictionary by using the Java Dictionary class.

Java Dictionary class is an abstract class parent class of any class. It belongs to java.util package. Its direct known subclass is the Hashtable class. Like the Hashtable class, it also maps the keys to values. Note that every key and value is an object and any non-null object can be used as a key and as a value.

2.Difference between Hashset, Hashmap, TreeSet

Hashset

HashSet is set Implementation.

- A HashSet is a collection that contains unique elements and provides fast access to elements based on their hash values, without guaranteeing any order.
- HashSet is Unordered , the elements in a hashset are not stored in any particular order.
- It does not allow duplicate elements.
- When you need a collection with unique elements but do not care about the order.

Hashmap

- Java Hash Map is a hash table based implementation of Map interface.
- A HashMap is a collection that stores key-value pairs, allowing fast lookups, inserts, and removals of key-value mappings. Keys must be unique, but values can be duplicated.
- Hashmap is Unordered, The keys in a HashMap are not stored in any particular order.
- It allows only unique keys but allows duplicate values. The keys must be unique, but multiple keys can map to the same value.
- When you need to store key-value pairs, where each key maps to a single value, and you want fast access by key.

TreeSet

- HashSet is set Implementation.

- A TreeSet is a collection that contains unique elements, is sorted in natural order (or according to a custom comparator), and provides efficient operations on ordered data.
- TreeSet is Sorted order, elements are stored in their natural order (if they are Comparable) or according to a custom comparator.
- It does not allow duplicate elements.
- When you need a collection with unique elements in a sorted order.

3. Interfaces

- An interface in Java is a reference type, similar to a class, that can contain only constants, method signatures, default methods, static methods, and nested types. Interfaces cannot contain instance fields or constructors.
- They are used to specify a set of methods that a class must implement, providing a way to achieve abstraction and multiple inheritance in Java.
- It can be used to achieve loose coupling.

4. Why interface used as a Base class for collection?

In Java, interfaces are used as the base types (or base classes, conceptually) for the Collections Framework because they provide several key benefits that contribute to the design principles of abstraction, flexibility, reusability, and polymorphism.

5. Why interface ends with able?

In Java, the convention of naming interfaces with the suffix -able (e.g., Serializable, Comparable, Runnable) is a naming convention that indicates that the class implementing the interface is capable of or able to perform the action or behavior described by the interface. This naming style enhances readability, clarity, and understanding of the interface's purpose.

But,not all interfaces in Java end with -able. While many interfaces in Java, especially those related to capabilities or behaviors, use the -able suffix, it's far from a universal rule. The -able suffix is a naming convention used for interfaces that define a specific capability (like Serializable, Cloneable, Comparable, etc.), but it's not required or followed for all interfaces.

6. 15 Interface in Java

1. Collection
2. List
3. Set
4. Map
5. Queue

6. Deque
7. Autocloseable
8. Stream
9. Iterator
10. Iterable
11. Clonable
12. Comparable
13. Runnable
14. Serializable
15. Comparator

7. Throw and Throws

Throw

The throw keyword in Java is used to explicitly throw an exception from a method or any block of code. We can throw either checked or unchecked exception. The throw keyword is mainly used to throw custom exceptions

Throws

throws is a keyword in Java that is used in the signature of a method to indicate that this method might throw one of the listed type exceptions. The caller to these methods has to handle the exception using a try-catch block.

8. Why map is not inside the collections.

- Collection is intended for groups of individual elements, whereas Map is designed to handle key-value pairs (a set of mappings from keys to values).
- Map is not part of Collection because it works with key-value pairs rather than individual elements.
- This separation makes it easier to maintain clarity in the design of the Java Collections Framework.