PW SKILLS

Lecture

Collection and Map API in Java

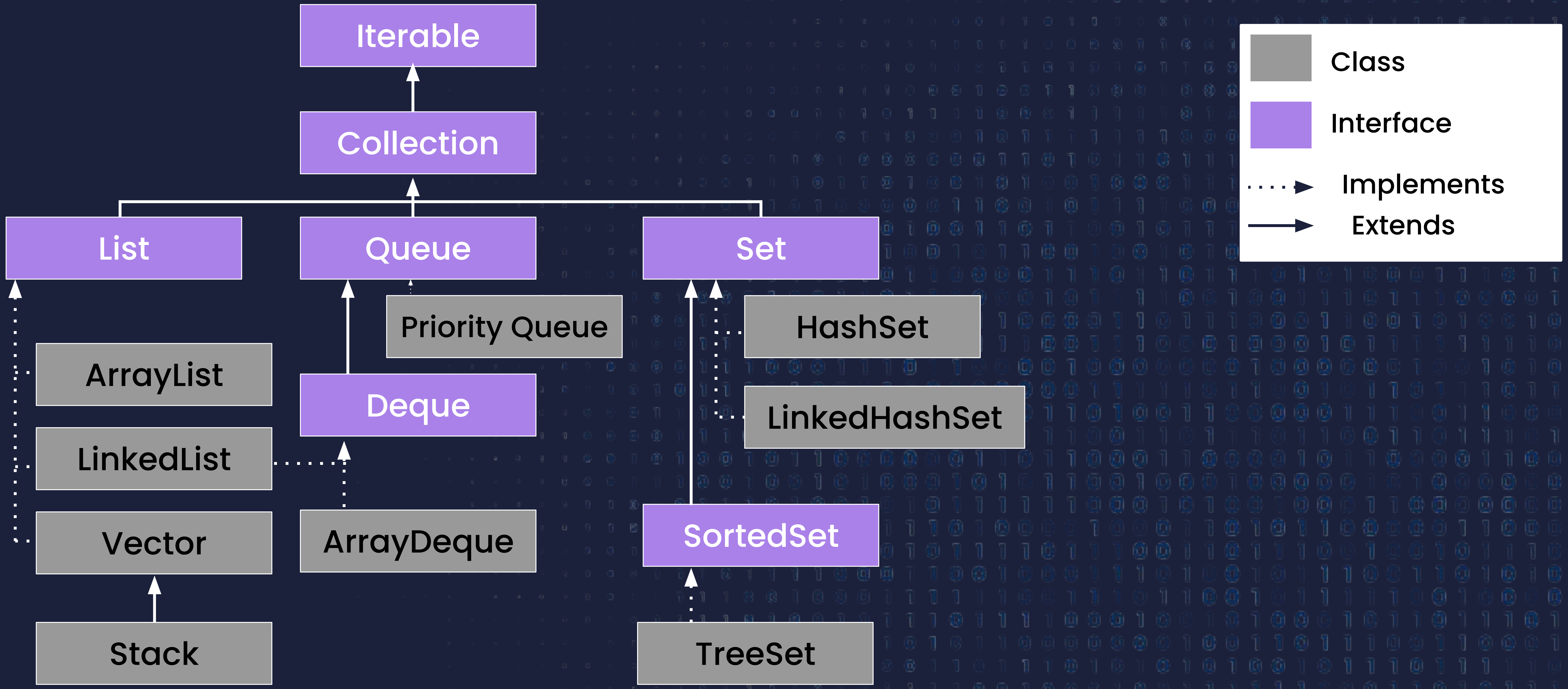
List of Concepts Involved:

- Why Collection?
- Collection Hierarchy
- ArrayList
- LinkedList
- ArrayDeque
- PriorityQueue
- TreeSet
- HashSet
- LinkedHashSet
- Iterator , List Iterator
- Introduction to Map in Java
- Map Hierarchy
- HashMap
- Other Inbuilt classes and Inbuilt methods under Map Hierarchy

Why Collection?

The Collection Framework(1.2V) is a set of interfaces, implementations, and algorithms that provide a standardized way to work with groups of objects. It was introduced to simplify the process of storing, manipulating, and processing collections of data in Java applications.

Collection Hierarchy



List Interface

- List interface is the child interface of Collection interface. It inhibits a list type data structure in which we can store the ordered collection of objects. It can have duplicate values.
- List interface is implemented by the classes ArrayList, LinkedList, Vector, and Stack.

ArrayList

- The ArrayList class implements the List interface.
- It uses a dynamic array to store the duplicate element of different data types.
- The ArrayList class maintains the insertion order and is non-synchronized.
- The elements stored in the ArrayList class can be randomly accessed.

LinkedList

- LinkedList implements the List and Deque interface.
- It uses a doubly linked list internally to store the elements.
- It can store the duplicate elements.
- It maintains the insertion order and is not synchronized.
- In LinkedList, the manipulation is fast because no shifting is required.

ArrayDeque

- ArrayDeque class implements the Deque interface.
- It facilitates us to use the Deque. Unlike queue, we can add or delete the elements from both the ends.
- ArrayDeque is faster than ArrayList and Stack and has no capacity restrictions.

PriorityQueue

- The PriorityQueue class implements the Queue interface.
- It holds the elements or objects which are to be processed by their priorities.

PriorityQueue doesn't allow null values to be stored in the queue.

TreeSet

- TreeSet class implements the Set interface that uses a tree for storage.
- TreeSet also contains unique elements.
- However, the access and retrieval time of TreeSet is quite fast.
- The elements in TreeSet stored in ascending order.

HashSet

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 HashSet class are:

- HashSet stores the elements by using a mechanism called hashing.
- HashSet contains unique elements only.
- HashSet allows null value.
- HashSet class is non synchronized.
- The initial default capacity of HashSet is 16, and the load factor is 0.75.

LinkedHashSet

LinkedHashSet class is a Hashtable and Linked list implementation of the Set interface. It inherits the HashSet class and implements the Set interface.

The important points about the LinkedHashSet class are:

- LinkedHashSet class contains unique elements only like HashSet.
- provides all optional set operations and permits null elements.
- is non-synchronized.
- maintains insertion order.

Iterator , List Iterator

- An Iterator is an interface and we can traverse the elements of a list in a forward direction whereas a ListIterator is an interface that extends the Iterator interface and we can traverse the elements in both forward and backward directions.
- An Iterator can be used in these collection types like List, Set, and Queue whereas ListIterator can be used in List collection only.
- The important methods of Iterator interface are hasNext(), next() and remove() whereas important methods of ListIterator interface are add(), hasNext(), hasPrevious() and remove().

Legacy classes and Enumeration

Legacy classes

Legacy classes refers to the older classes that were included in the early versions of Java and have since been replaced by newer, more efficient classes. One such class is Enumeration, which is a legacy interface that was used to traverse collections before the introduction of the Iterator interface.

The Legacy classes are Dictionary, Hashtable, Properties, Stack, and Vector. The Legacy interface is the Enumeration interface.

Enumeration Interface

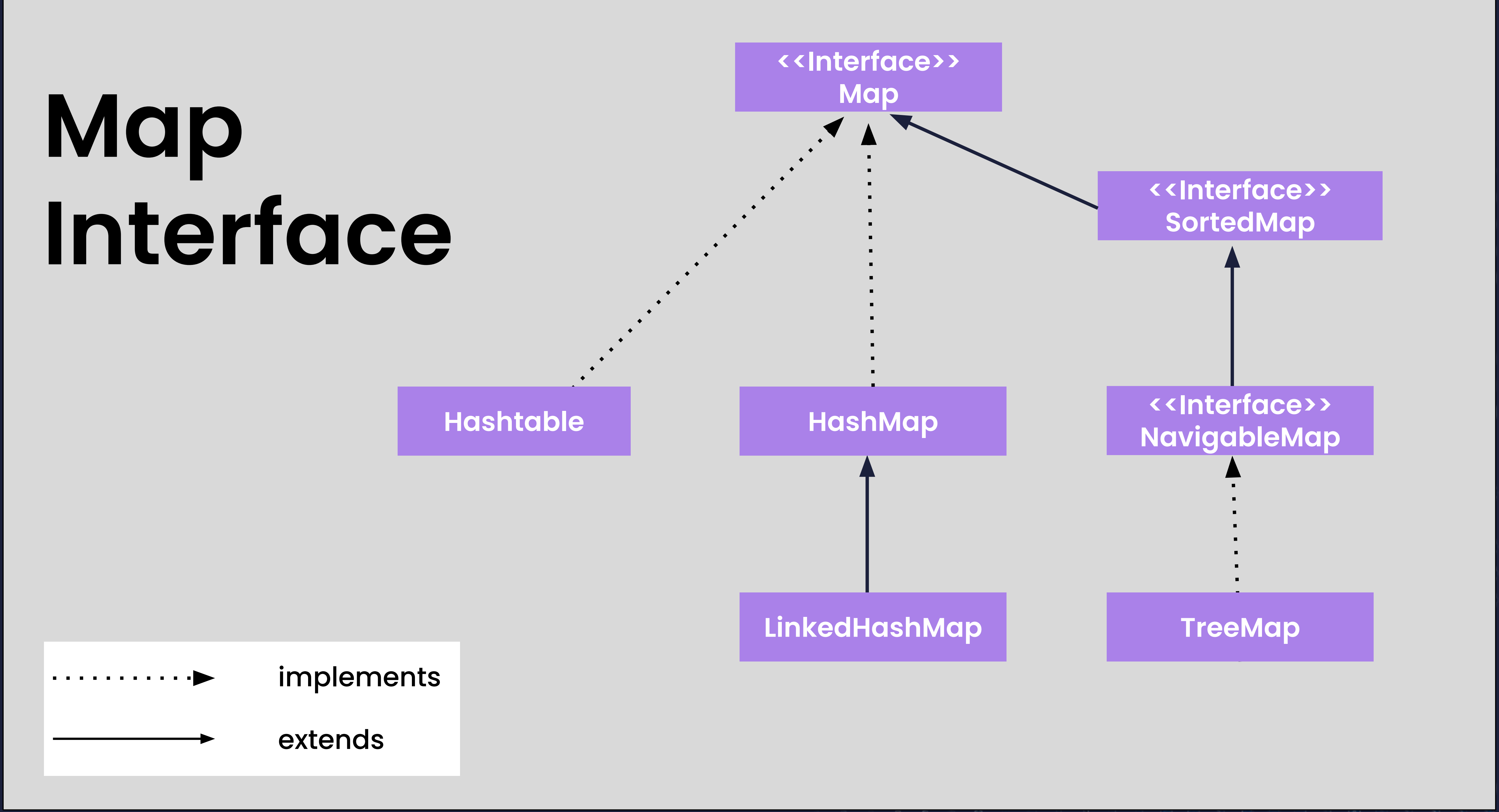
The Enumeration interface is the only legacy interface. It defines methods, which help us to enumerate the elements in a collection of objects. This interface has been suspended by Iterator. It contains only 2 methods as shown here:

- **boolean hasMoreElements():** Checks if the enumeration contains more elements. If it contains more elements, then it returns true, else it returns false.
- **Object nextElement():** Returns the next element of the enumeration. If there are no more elements to retrieve then it throws NoSuchElementException.

Introduction to Map in Java

- A Map is an interface that represents a collection of key-value pairs, where 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.

Map Hierarchy



HashMap

- Java HashMap allows null key and null values.
- HashMap is not an ordered collection. You can iterate over HashMap entries through keys set but they are not guaranteed to be in the order of their addition to the HashMap.
- HashMap is almost similar to Hashtable except that it's non-synchronized and allows null key and values.

In-built classes in Map Hierarchy

The three primary classes that implement map in Java are:

- HashMap
- LinkedHashMap
- TreeMap

Inbuilt methods under Map Hierarchy

Some of the Inbuilt Methods are:

- clear()
- containsKey(Object)
- containsValue(Object)
- entrySet()
- equals(Object)
- get(Object)
- hashCode()
- isEmpty()
- keySet()
- put(Object, Object)

SKILLS

THANK YOU