PW SKILLS

Lecture

Map and Generics

List of Concepts Involved:

- Introduction to Map in Java
- Map Hierarchy
- HashMap
- Other Inbuilt classes and Inbuilt methods under Map Hierarchy
- Need of Generics and Basics of Generics
- More on Generics in Java
- Collections class and it's inbuilt methods in Java
- Comparator vs Comparable Interface

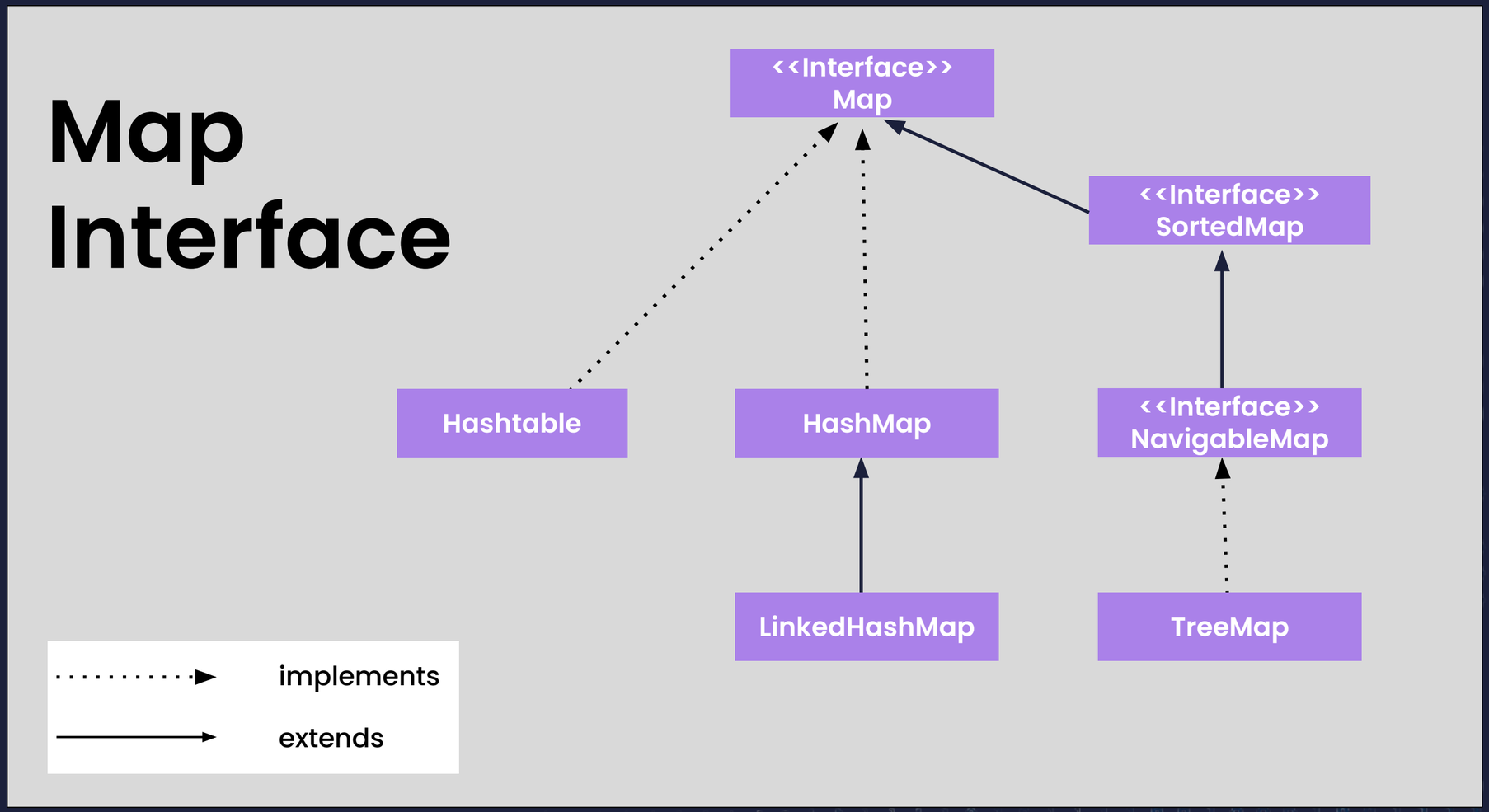
Topics covered in Previous Session:

- Collection Framework

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)

Generics and Basics of Generics

- It was introduced from JDK 5. Generics is a feature in Java that allows for the creation of classes, interfaces, and methods that can operate on a variety of data types.
- It provides compile-time type safety by enabling the specification of the data type of objects that a class or method can work with.
- Generics are implemented using type parameters, which are specified inside angle brackets <>.

Collections class and it's inbuilt methods in Java

- The Collections class in Java is a utility class that provides a set of static methods for working with collections.
- The class provides methods for manipulating and searching collections, sorting lists, finding the minimum and maximum values in a collection, and more.

Here are some of the inbuilt methods provided by the Collections class:

- sort(List<T> list)
- reverse(List<T> list)
- shuffle(List<?> list)
- binarySearch(List<? extends Comparable<? super T>> list, T key)
- max(Collection<? extends T> coll)
- frequency(Collection<?> c, Object o) etc...

Comparator vs Comparable Interface

Java provides two interfaces to sort objects using data members of the class:

- **Comparable:**
    - The Comparable interface is used to define the natural ordering of a class.
    - The compareTo() method is defined in the Comparable interface

- **Comparator:**
    - The Comparator interface is used to define a custom ordering of objects
    - The compare() method is defined in the Comparator interface.

Next Lecture

- Important APIs and Annotation

PW SKILLS

THANK YOU