In Java, the Map interface is a fundamental part of the Collections Framework, used to store data in key-value pairs.

Key Characteristics:

* **Key-Value Association:**

Each entry in a Map consists of a unique key mapped to a specific value.

* **Unique Keys:**

A Map cannot contain duplicate keys; each key can be associated with at most one value. Values, however, can be duplicated.

* Not a Subtype of Collection:

Unlike List or Set, Map does not extend the Collection interface, meaning it has its own distinct set of methods and behaviors.

* **Implementations:**

Common implementations include HashMap (fast, unordered), TreeMap (sorted by key), and LinkedHashMap (maintains insertion order).

* **Null Handling:**

Some Map implementations, like HashMap and LinkedHashMap, allow one null key and multiple null values, while others, like TreeMap, do not allow null keys.

Common Methods:

* put(K key, V value): Adds or updates a key-value pair.
* get(Object key): Returns the value associated with the given key.
* remove(Object key): Removes the mapping for the specified key.
* containsKey(Object key): Checks if the map contains the specified key.
* keySet(): Returns a Set view of the keys contained in the map.
* values(): Returns a Collection view of the values contained in the map.
* entrySet(): Returns a Set view of the key-value mappings contained in the map.

import java.util.HashMap;  
import java.util.Map;  
  
public class MapExample {  
 public static void main(String[] args) {  
 Map<String, Integer> ages = new HashMap<>();  
  
 *// Adding elements*  
 ages.put("Alice", 30);  
 ages.put("Bob", 25);  
 ages.put("Charlie", 35);  
  
 *// Retrieving a value*  
 System.out.println("Alice's age: " + ages.get("Alice"));  
  
 *// Checking for a key*  
 System.out.println("Contains Bob? " + ages.containsKey("Bob"));  
  
 *// Iterating through entries*  
 for (Map.Entry<String, Integer> entry : ages.entrySet()) {  
 System.out.println(entry.getKey() + " is " + entry.getValue() + " years old.");  
 }  
 }  
}