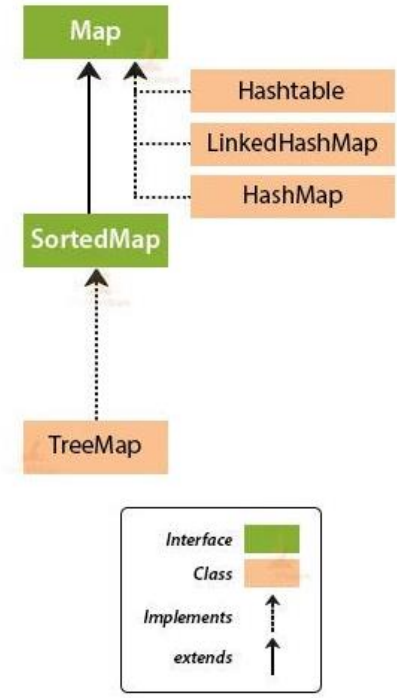
Day 23

Collections – HashMap

HashMap

- HashMap class implements the **Map** interface which allows us to store **key** and **value** pair.
- HashMap contains values based on the **key**.
- HashMap contains only **unique keys**.
- HashMap may have **one null key** and **multiple null values**.
- HashMap maintains **no order**.



Terminology:

Key

- A key is an object that you use to retrieve a value from the HashMap.
- Each key must be unique, meaning you cannot have duplicate keys in a HashMap.
- When you want to store or retrieve a value from the HashMap, you use its associated key.

Value

- A **value** is the data or object associated with a specific key in the HashMap.
- While keys must be unique, values can be duplicated.

Pair

- In a HashMap, each **key** is associated with a **value**; this is known as a **key-value pair**.
- Each entry in a HashMap is a pair consisting of a key and its corresponding value.

keySet

- The keySet() method returns a set view of all the keys in the HashMap.
- You can iterate over this set to access all keys without directly modifying the HashMap.

Entry

- An **entry** is a single key-value pair in the HashMap.
- You can access each entry using the entrySet() method, which returns a set view of the map's entries, allowing you to iterate over both keys and values together.

Comparison (ArrayList Vs HashSet Vs HashMap):

Feature	ArrayList	HashSet	HashMap
Definition	A resizable array that allows duplicates.	A collection that does not allow duplicates.	A collection that stores key-value pairs.
Data Structure Type	List	Set	Map
Duplicates	Allows duplicates.	Does not allow duplicates.	Keys must be unique; values can be duplicated.
Order of Elements	Insertion order preserved.	Insertion order not preserved.	Insertion order not preserved.
Null Values	Allows multiple null elements.	Allows a single null element.	Allows one null key and multiple null values.
Index-based Access	Yes, elements can be accessed by index.	No, elements cannot be accessed by index.	No, access is via keys only.
Common Operations	add(), get(), remove(), size().	add(), contains(), remove(), size().	put(), get(), remove(), size().
Internal Implementation	Backed by a dynamically resizing array.	Backed by a hash table.	Backed by a hash table for key-value mapping.
Best Used For	When you need ordered collection with possible duplicates.	When you need fast lookup with unique elements.	When you need to associate values with keys for fast retrieval.

Summary:

- **ArrayList** is ideal for ordered collections and allows duplicates.
- **HashSet** is used when uniqueness is required and order is not important.
- **HashMap** is best for storing key-value pairs where quick lookup by key is needed.

Quiz: HashMap

1. **What does a HashMap store?**
    - A. Only keys
    - B. Only values
    - C. Key-value pairs
    - D. Only objects

    Answer: C. Key-value pairs
2. **What happens if you try to insert a duplicate key in a HashMap?**
    - A. The new key is added.
    - B. The old value associated with the key is overwritten by the new value.
    - C. An error is thrown.
    - D. The HashMap does not allow duplicate keys and rejects the entry.

    Answer: B. The old value associated with the key is overwritten by the new value.
3. **Which method is used to retrieve a value from a HashMap using a key?**
    - A. getValue()
    - B. fetch()
    - C. get()
    - D. find()

    Answer: C. get()
4. **Can a HashMap have null keys and null values?**
    - A. No, neither null keys nor null values are allowed.
    - B. Yes, but only one null key is allowed and multiple null values.
    - C. Yes, multiple null keys and values are allowed.
    - D. Only null values are allowed but not null keys.

    Answer: B. Yes, but only one null key is allowed and multiple null values.
5. **Which method is used to remove a key-value pair from a HashMap?**
    - A. delete()
    - B. remove()
    - C. clear()
    - D. discard()

    Answer: B. remove()

More Videos on Collections:

https://t.ly/q9om_