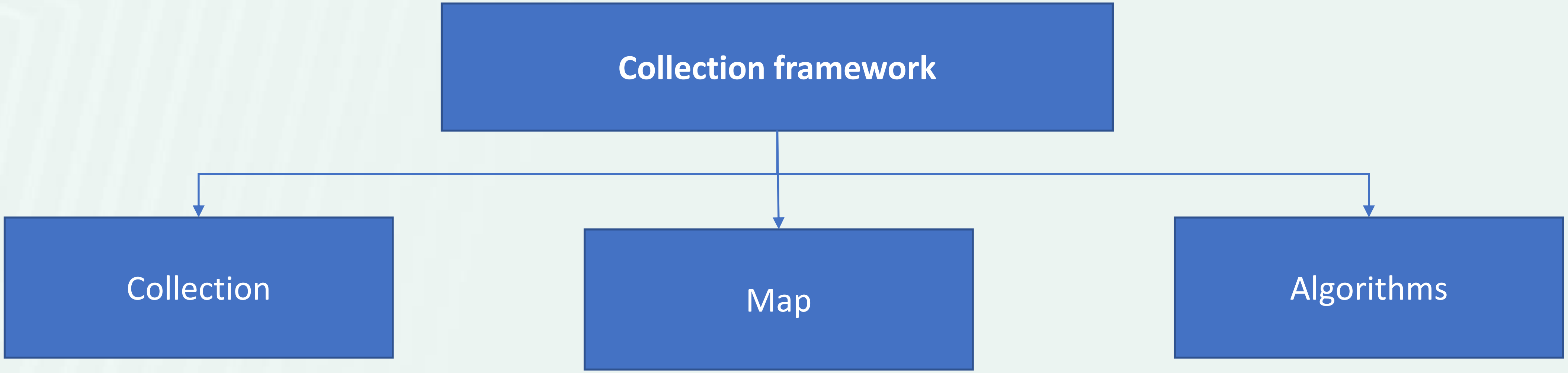
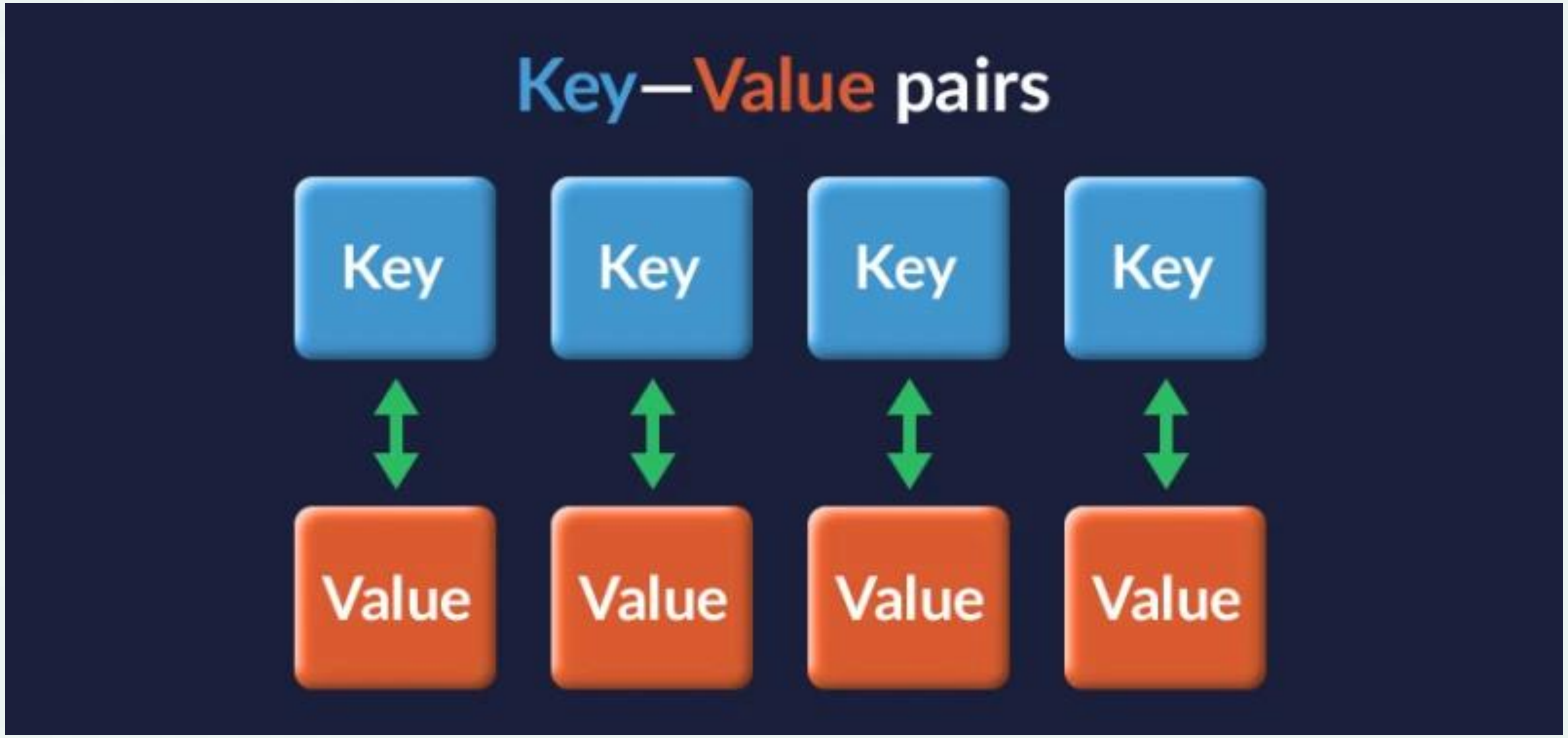
Set va Map. Map interface

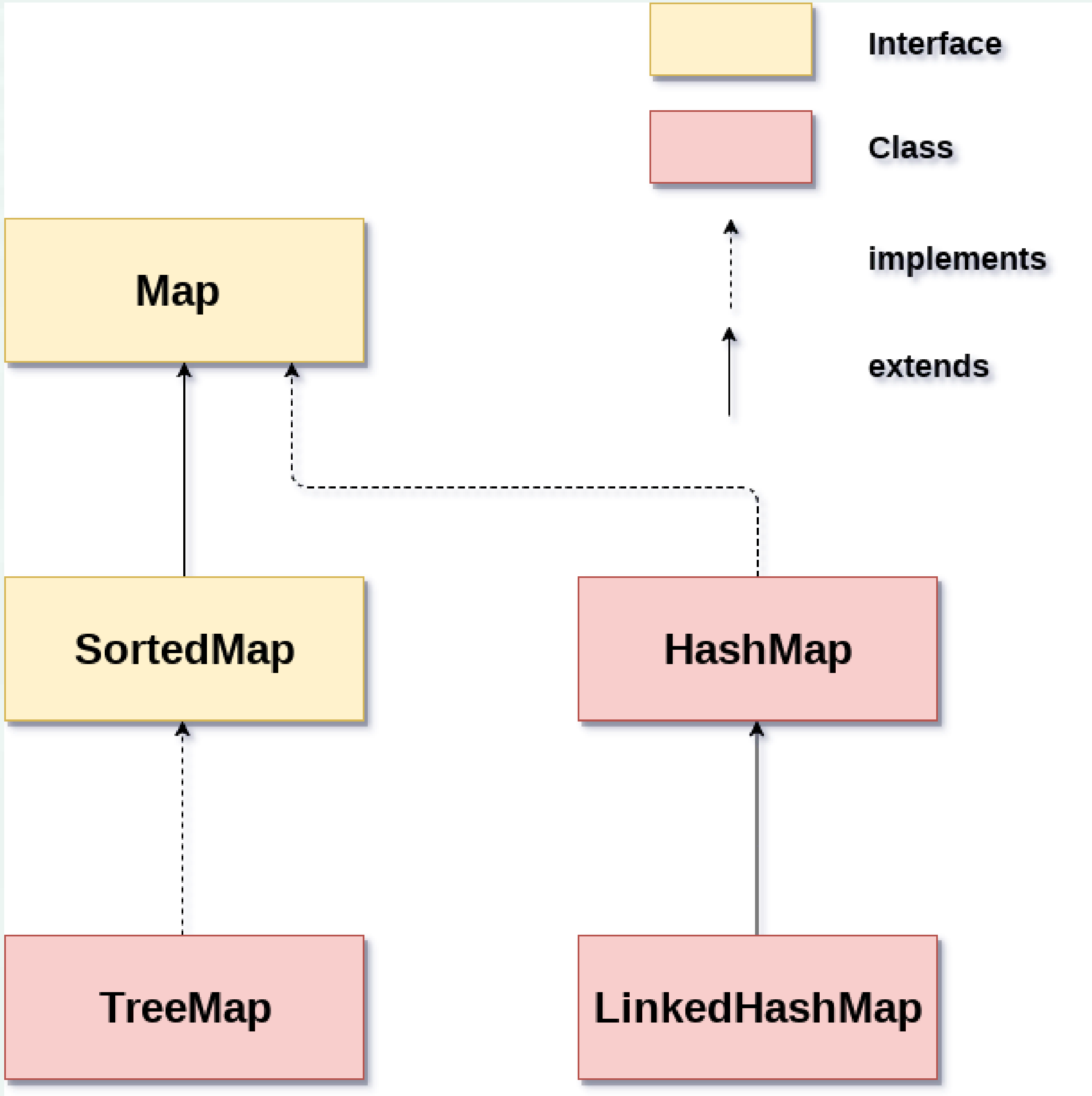
PERSONAL
DEVELOPMENT PROCESS
stay weird, stay different

Collection framework



Map





Interface

Class

implements

extends

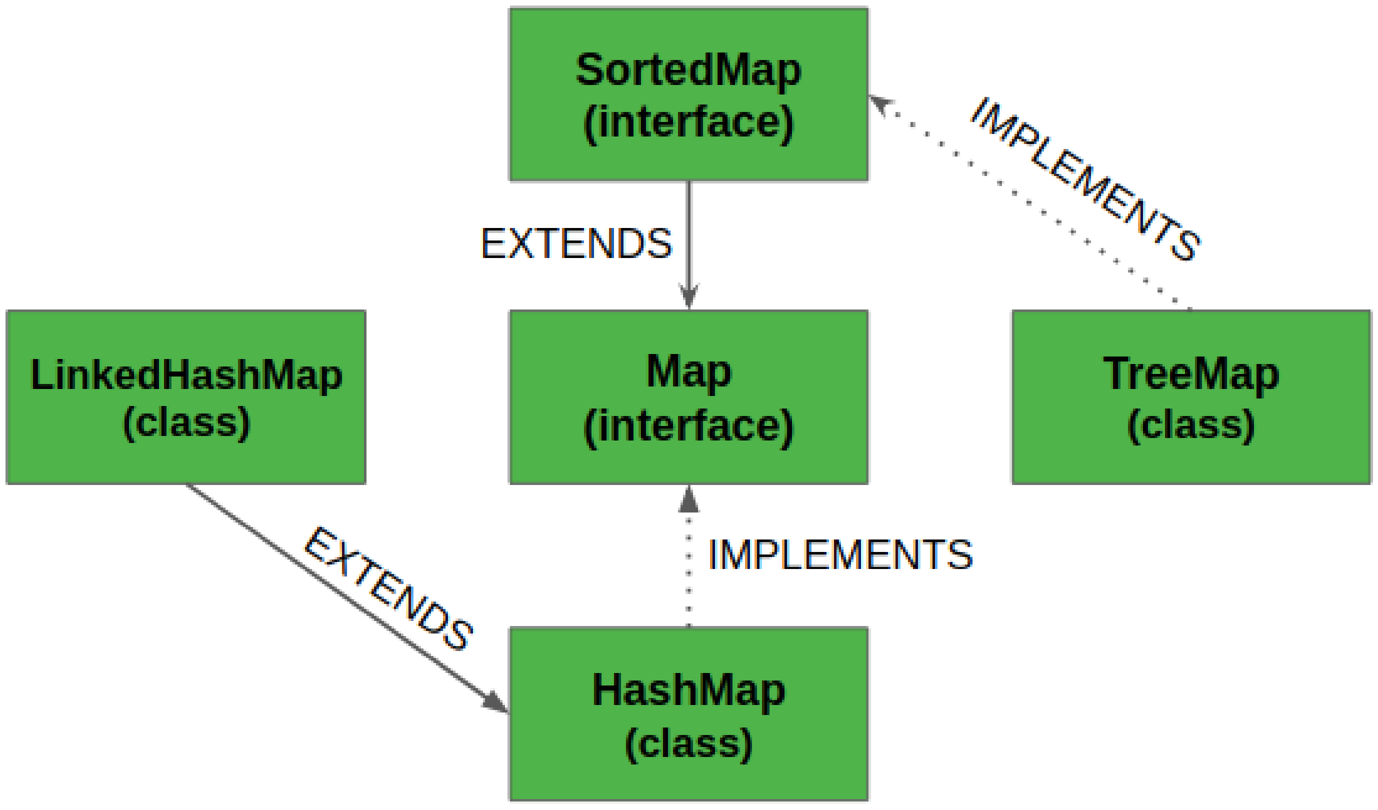
Map

SortedMap

HashMap

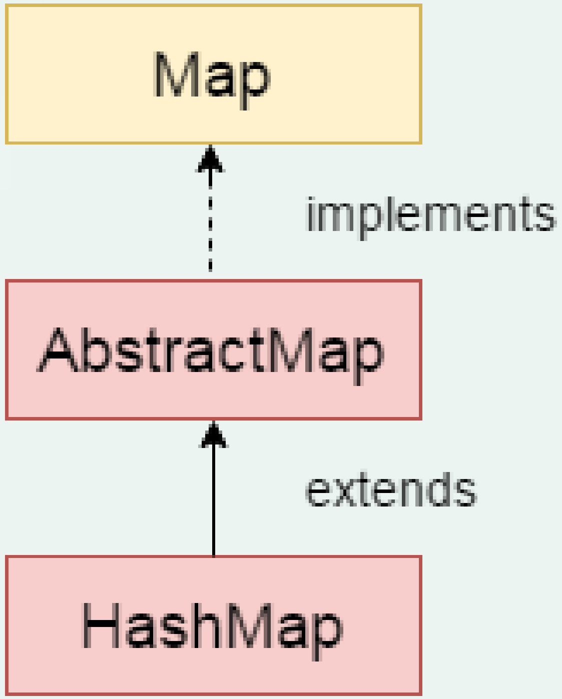
TreeMap

LinkedHashMap

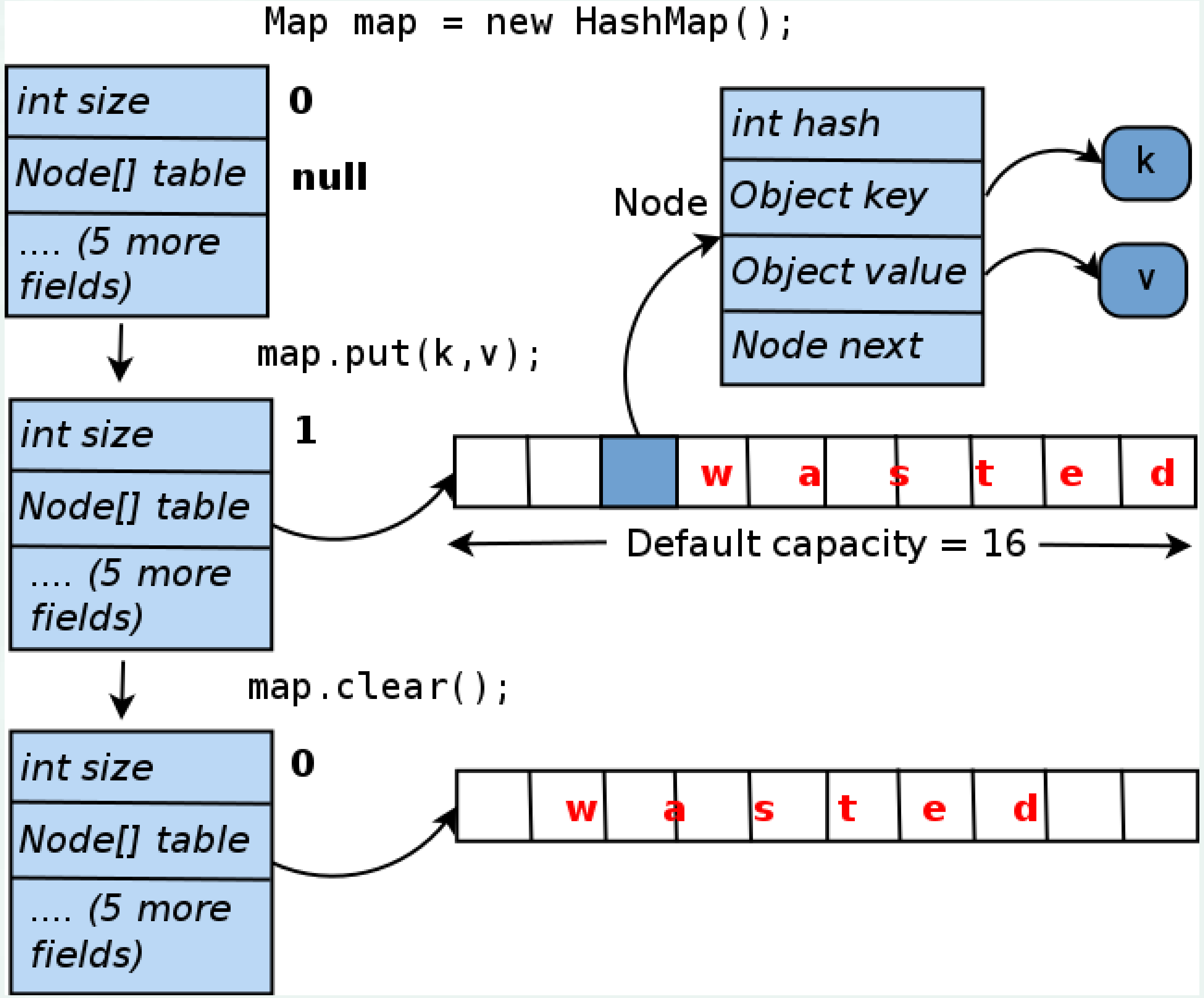


MAP Hierarchy in Java

HashMap



- ✓ key asosida saqlangan value lardan tashkil topadi.
- ✓ key lari unikal bo’ladi;
- ✓ faqat bir dona null key ga va bir nechta null value ga ega bo’lishi mumkin;
- ✓ Synchronized emas;
- ✓ Ordered emas;
- ✓ initial default capacity 16 ga load factor esa 0.75 ga teng.



```
Map map = new HashMap();
```

int size **0**

Node[] table **null**

.... (5 more fields)

```
map.put(k,v);
```

int size **1**

Node[] table

.... (5 more fields)

Node

int hash

Object key

Object value

Node next

k

v

w a s t e d

Default capacity = 16

```
map.clear();
```

int size **0**

Node[] table

.... (5 more fields)

w a s t e d

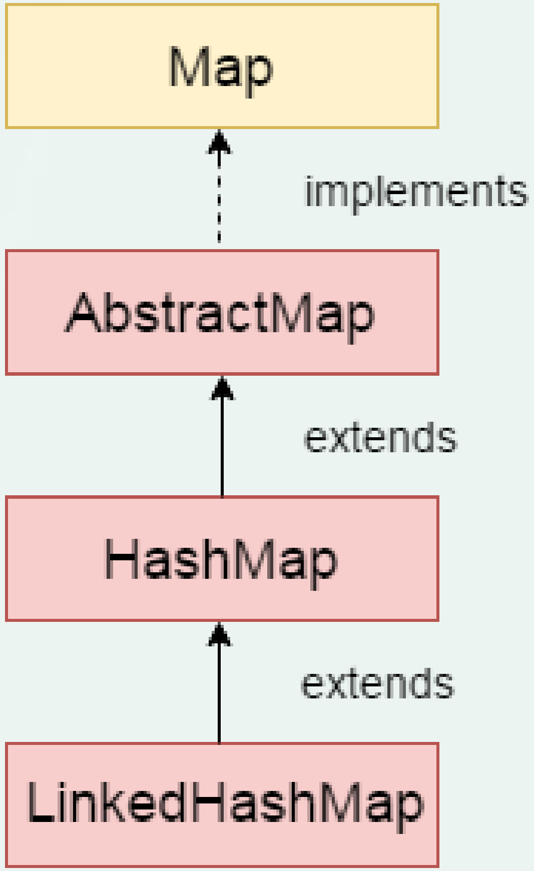
Contructor

Constructor	Description
HashMap()	It is used to construct a default HashMap.
HashMap(Map<? extends K,? extends V> m)	It is used to initialize the hash map by using the elements of the given Map object m.
HashMap(int capacity)	It is used to initializes the capacity of the hash map to the given integer value, capacity.
HashMap(int capacity, float loadFactor)	It is used to initialize both the capacity and load factor of the hash map by using its arguments.

Methodlari

V put(Object key, Object value)	It is used to insert an entry in the map.
void putAll(Map map)	It is used to insert the specified map in the map.
V putIfAbsent(K key, V value)	It inserts the specified value with the specified key in the map only if it is not already specified.
V get(Object key)	This method returns the object that contains the value associated with the key.
V getOrDefault(Object key, V defaultValue)	It returns the value to which the specified key is mapped, or defaultValue if the map contains no mapping for the key.
V remove(Object key)	It is used to delete an entry for the specified key.
boolean remove(Object key, Object value)	It removes the specified values with the associated specified keys from the map.
boolean containsValue(Object value)	This method returns true if some value equal to the value exists within the map, else return false.
boolean containsKey(Object key)	This method returns true if some key equal to the key exists within the map, else return false.
boolean isEmpty()	This method returns true if the map is empty; returns false if it contains at least one key.
void forEach(BiConsumer<? super K,? super V> action)	It performs the given action for each entry in the map until all entries have been processed or the action throws an exception.
V replace(K key, V value)	It replaces the specified value for a specified key.
void replaceAll(BiFunction<? super K,? super V,? extends V> function)	It replaces each entry's value with the result of invoking the given function on that entry until all entries have been processed or the function throws an exception.
Set keySet()	It is used to return a set view of the keys contained in this map.
Collection<V> values()	It returns a collection view of the values contained in the map.
Set entrySet()	It is used to return a collection view of the mappings contained in this map.

LinkedHashMap

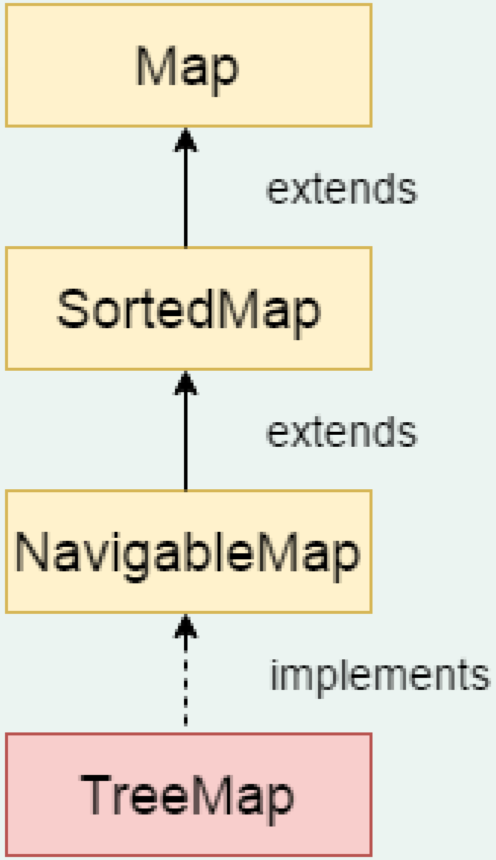


- ✓ key asosida saqlangan value lardan tashkil topadi.
- ✓ key lari unikal bo’ladi;
- ✓ faqat bir dona null key ga va bir nechta null value ga ega bo’lishi mumkin;
- ✓ Synchronized emas;
- ✓ Insertion ordered;
- ✓ initial default capacity 16 ga load factor esa 0.75 ga teng.

Contructor

Constructor	Description
LinkedHashMap()	It is used to construct a default LinkedHashMap.
LinkedHashMap(int capacity)	It is used to initialize a LinkedHashMap with the given capacity.
LinkedHashMap(int capacity, float loadFactor)	It is used to initialize both the capacity and the load factor.
LinkedHashMap(int capacity, float loadFactor, boolean accessOrder)	It is used to initialize both the capacity and the load factor with specified ordering mode.
LinkedHashMap(Map<? extends K,? extends V> m)	It is used to initialize the LinkedHashMap with the elements from the given Map class m.

TreeMapHashMap



PERSONAL
DEVELOPMENT PROCESS
stay weird, stay different

- ✓ key asosida saqlangan value lardan tashkil topadi, NavigableMap, va SartedMap interfacelarini implement qiladi.
- ✓ key lari unikal bo’ladi;
- ✓ key null ga teng bo’lmaydi lekin value null ga ega bo’lishi mumkin;
- ✓ Synchronized emas;
- ✓ ascending ordered;
- ✓ initial default capacity 16 ga load factor esa 0.75 ga teng.

Contructor

Constructor	Description
TreeMap()	It is used to construct an empty tree map that will be sorted using the natural order of its key.
TreeMap(Comparator<? super K> comparator)	It is used to construct an empty tree-based map that will be sorted using the comparator comp.
TreeMap(Map<? extends K,? extends V> m)	It is used to initialize a treemap with the entries from **m**, which will be sorted using the natural order of the keys.
TreeMap(SortedMap<K,? extends V> m)	It is used to initialize a treemap with the entries from the SortedMap **sm**, which will be sorted in the same order as **sm.**

Methodlari

Map.Entry<K,V> ceilingEntry(K key)	It returns the key-value pair having the least key, greater than or equal to the specified key, or null if there is no such key.
K ceilingKey(K key)	It returns the least key, greater than the specified key or null if there is no such key.
NavigableSet<K> descendingKeySet()	It returns a reverse order NavigableSet view of the keys contained in the map.
NavigableMap<K,V> descendingMap()	It returns the specified key-value pairs in descending order.
Map.Entry firstEntry()	It returns the key-value pair having the least key.
SortedMap<K,V> headMap(K toKey)	It returns the key-value pairs whose keys are strictly less than toKey.
Map.Entry<K,V> higherEntry(K key)	It returns the least key strictly greater than the given key, or null if there is no such key.
K higherKey(K key)	It is used to return true if this map contains a mapping for the specified key.
Map.Entry<K,V> lastEntry()	It returns the key-value pair having the greatest key, or null if there is no such key.
Map.Entry<K,V> lowerEntry(K key)	It returns a key-value mapping associated with the greatest key strictly less than the given key, or null if there is no such key.
K lowerKey(K key)	It returns the greatest key strictly less than the given key, or null if there is no such key.
SortedMap<K,V> subMap(K fromKey, K toKey)	It returns key-value pairs whose keys range from fromKey, inclusive, to toKey, exclusive.
SortedMap<K,V> tailMap(K fromKey)	It returns key-value pairs whose keys are greater than or equal to fromKey.
K firstKey()	It is used to return the first (lowest) key currently in this sorted map.
K lastKey()	It is used to return the last (highest) key currently in the sorted map.

E’TIBORINGIZ UCHUN RAXMAT