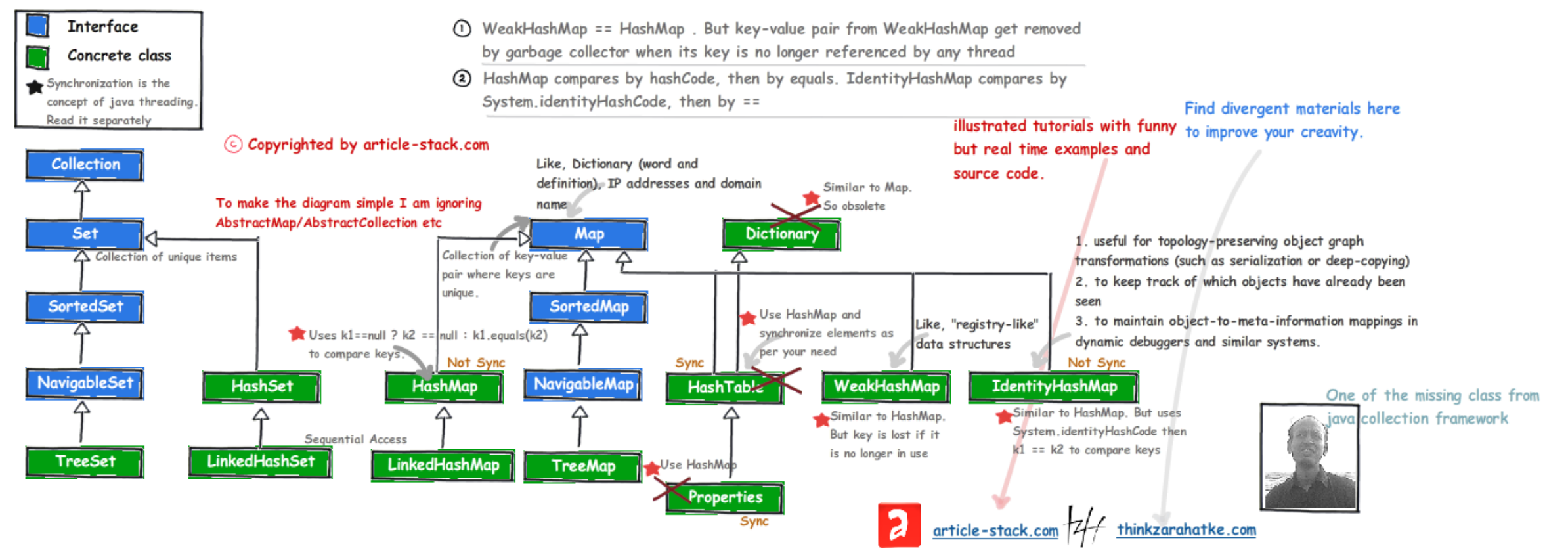

Interface
Concrete class
Synchronization is the concept of java threading. Read it separately
① WeakHashMap == HashMap . But key-value pair from WeakHashMap get removed by garbage collector when its key is no longer referenced by any thread
② HashMap compares by hashCode, then by equals. IdentityHashMap compares by System.identityHashCode, then by ==
© Copyrighted by article-stack.com
illustrated tutorials with funny but real time examples and source code.
Find divergent materials here to improve your creavity.
Collection
Set
Collection of unique items
SortedSet
NavigableSet
TreeSet
To make the diagram simple I am ignoring AbstractMap/AbstractCollection etc
HashSet
LinkedHashSet
Sequential Access
Like, Dictionary (word and definition), IP addresses and domain name
Map
Collection of key-value pair where keys are unique.
Uses k1==null ? k2 == null : k1.equals(k2) to compare keys.
HashMap
Not Sync
LinkedHashMap
SortedMap
NavigableMap
TreeMap
Similar to Map. So obsolete
Dictionary
Use HashMap and synchronize elements as per your need
Sync
HashTable
Use HashMap
Properties
Sync
Like, "registry-like" data structures
WeakHashMap
Similar to HashMap. But key is lost if it is no longer in use
1. useful for topology-preserving object graph transformations (such as serialization or deep-copying)
2. to keep track of which objects have already been seen
3. to maintain object-to-meta-information mappings in dynamic debuggers and similar systems.
IdentityHashMap
Not Sync
Similar to HashMap. But uses System.identityHashCode then k1 == k2 to compare keys
One of the missing class from java collection framework
article-stack.com
thinkzarahatke.com

Dont Know

Start

Values are known at compilation time?

yes → EnumSet/EnumMap

No

Like Set of language keywords, friends, records returned from db queries, open connection to database server or a client etc

Elements are unique?

yes

Natural ordering is required?

yes

Size of Data

Large → TreeMap/TreeSet

When insertion is rare, less or one time

To cache property files or database table

Elements get inserted in natural order

TreeMap/TreeSet implements NavigableXXX interface from JDK 1.6 onwards, which allow traversing in either ascending or descending order. It also has additiopnal methods to select closest elements to an element

No

Small

No

When insertion,deletion is frequent

LinkedList

No

Random access required?

yes

can't hold more than Integer.MAX_VALUE

ArrayList

When insertion is rare, less or one time

Iteration order should be insertion order?

yes

LinkedHashMap/LinkedHashSet

slower than HashMap/HashSet

No

Java Heap Memory matters?

yes

WeakedHashMap

No

Keys are class object?

yes

Faster than HashMap

IdentityHashMap

No

Faster than LinkedHashMap /LinkedHashSet

HashMap/HashSet

① LinkedHashMap/Set is a type of HashMap/Set gives additional functionality of sequential access

① NavigableMap/Set is a type of SortedMap/Set gives additional functionality of closest matching selection without manual iteration.

If rB is type of B then rB will always have some additional functionalities over B. Thus rB will give lower performance than B. So if you can compromise with additional functionalities but not with performance then go for B.

◆Insertion order
Items are stored in same order they were entered.

◆Natural order
Items are sorted at the time of insertion. Sorting is done on the basis of Comparable Interface or Comparator Object

Interface

Concrete class

Synchronization is the concept of java threading. Read it separately

Insertion order — Items are stored in same order they were entered.

Natural order — Items are sorted at the time of insertion. Sorting is done on the basis of Comparable Interface or Comparator Object

Data Structure	HashXXX	LinkedXXX	TreeXXX	Properties
Balanced Binary Tree	☐	☐	☑	☐
Hash Table	☑	☐	☐	☑
Linked List	☐	☑	☐	☐

Property	List	Set	Map
Duplicates	☑		

	ArrayList	LinkedXXX	SortedXXX
iteration order	I	I	N

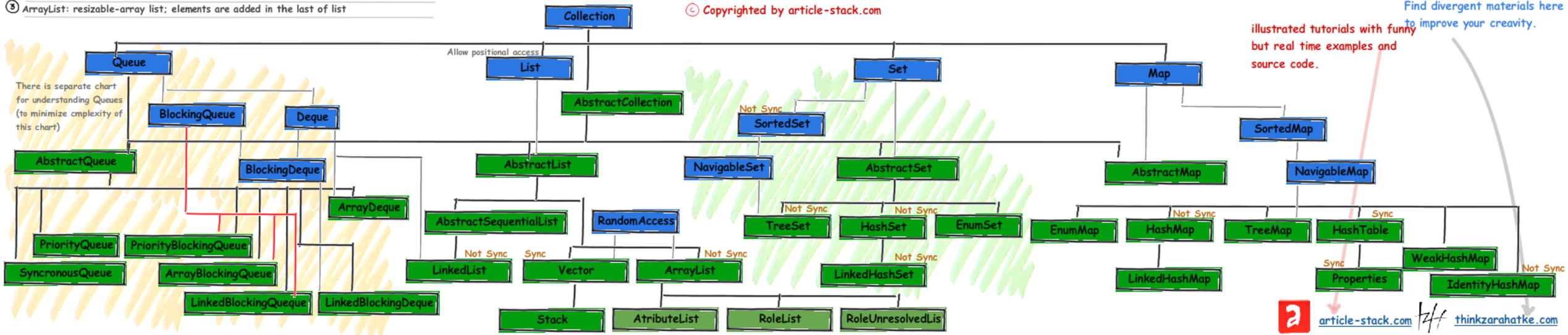
*I -Insertion
N -Natural

The AbstractCollection, AbstractSet, AbstractList, AbstractSequentialList and AbstractMap classes provide skeletal implementations of the core collection interfaces, to minimize the effort required to implement them.

In any data structure, if elements are naturally sorted then travering/searching would be faster. Hence, the update/deletion will be faster. But insertion would be s slower. These data structures are suitable to handle large data.

One of the missing class from java collection framework

① TreeSet & HashTable don't allow Null
② HashMap uses Chain hash table. While IdentityHashMap uses linear-probe hash table.
③ ArrayList: resizable-array list; elements are added in the last of list



Find divergent materials here to improve your creavity.

illustrated tutorials with funny but real time examples and source code.

article-stack.com thinkzarahatke.com