Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Interview › Core Java Interview Q&A › Differences Between X & Y › Java Collection interview Q&A on differences between X and Y

# Java Collection interview Q&A on differences between X and Y

Posted on April 1, 2015 by Arulkumaran Kumaraswamipillai — No Comments ↓

This post is for quick brush-up. These Q&A are discussed in detailed elsewhere in posts like When to use which Java collection or data structure? and why? | Sorting objects in Java interview Q&A

**Q1.** What are differences between Enumeration and Iterator?
**A1.** Enumeration is old and it's there from JDK1.0 while iterator is newer. The key difference between Enumeration and iterator is that **"Iterator has a remove() method"** while Enumeration doesn't. Enumeration acts as Read-only interface, whilst an Iterator can manipulate the objects like adding and removing.

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

⊞ Ice Breaker Interview
⊟ Core Java Interview (
⊞ Java Overview (4)
⊞ Data types (6)
⊞ constructors-methc
⊞ Reserved Key Wor
⊞ Classes (3)
⊞ Objects (8)
⊞ OOP (10)
⊞ GC (2)
⊞ Generics (5)
⊞ FP (8)
⊞ IO (7)
⊞ Multithreading (12)
⊞ Algorithms (5)
⊞ Annotations (2)
⊞ Collection and Data
⊟ Differences Betwee
♥ Java Iterable \
♦ Multithreading
♦ Why do Proxy,

**Q2.** What are differences between fail-fast and fail-safe iterators?

**A2.** Iterators returned by most of pre JDK1.5 collection classes like Vector, ArrayList, HashSet, etc are **fail-fast iterators**. Iterators returned by JDK 1.5+ ConcurrentHashMap and CopyOnWriteArrayList classes are **fail-safe iterators**.

Use copy-on-write and concurrent maps from the java.util.concurrent package prevent **ConcurrentModificationException** being thrown while preserving thread safety. These classes provide **fail-safe iteration** as opposed to non-concurrent classes like ArrayList, HashSet, etc use **fail-fast iteration** leading to **ConcurrentModificationException** if you try to remove an element while iterating over a collection.

**Q3.** What are differences between ArrayList and LinkedList?

**A3.**

- Insertions and deletions are faster in LinkedList compared to an ArrayList as LinkedList uses links (i.e. before and next reference) as opposed to an ArrayList, which uses an array under the covers, and may need to resize the array if the array gets full. Adding to an ArrayList has a worst case scenario of O(n) whilst LinkedList has O(1).
- LinkedList has more memory footprint than ArrayList. An ArrayList only holds actual object whereas LinkedList holds both data and reference of next and previous node.
- Random access has the worst case scenario of O(n) in LinkedList as to access 6th element in a LinkedList with 8 elements, you need to traverse through 1 to 5th element before you can get to the 6th element, whereas in an ArrayList, you can get the 6th element with O(1) with list.get(5).

**Q4.** What are differences between HashSet,ArryList and CopyOnWriteArraySet and CopyOnWriteArrayList?

**A4.** HashSet and ArryList are not thread-safe and you need

Core Java Modif
Differences betw
Java Collection i
⊞ Event Driven Progr
⊞ Exceptions (2)
⊞ Java 7 (2)
⊞ Java 8 (24)
⊞ JVM (6)
⊞ Reactive Programr
⊞ Swing & AWT (2)
⊞ JEE Interview Q&A (3
⊞ Pressed for time? Jav
⊞ SQL, XML, UML, JSC
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & I
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶ Free Java Interviev

## 16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience Ir
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)

to provide your own synchronization, whereas CopyOnWriteArraySet and CopyOnWriteArrayList are not only thread-safe, but more efficient as they allow concurrent multiple reads and single write. This concurrent read and write behavior is accomplished by making a brand new copy of the list every time it is altered.

**Q5.** What are differences between HasMap and TreeMap?
**A5.** TreeMap is an implementation of a SortedMap, where the order of the keys can be sorted, and when iterating over the keys, you can expect that keys will be in order. HashMap on the other hand, makes no such guarantee on the order.

**Q6.** What are differences between HashMap and ConcurrentHashMap?
**A6.** HashMap is not thread-safe and you need to provide your own synchronization with Collections.synchornizedMap(hashMap), which will return a collection which is almost equivalent to the legacy Hashtable, where every modification operation on Map is locked. As the name implies, ConcurrentHashMap provides thread-safety by dividing the whole map into different partitions based upon concurrency level and locking only particular portion instead of locking the whole map. ConcurrentHashMap does not allow NULL key values, whereas HashMap there can only be one null key.

**Q7.** What are differences between HashMap and LinkedHashMap?
**A7.** LinkedHashMap will iterate in the order in which the entries were put into the map. HashMap does not provide any grantees about the iteration order.

**Q8.** What are differences between Queue and BlockingQueue?
**A8.** BlockingQueue is a Queue that supports additional operations that wait for the queue to become non-empty when retrieving an element, and wait for space to become available in the queue when storing an element. The main advantage is that a BlockingQueue is that it provides a correct, thread-safe implementation with throttling.

Transaction Managen

## 80+ step by step Java Tutorials

open all | close all
Setting up Tutorial (6)
Tutorial - Diagnosis (2
Akka Tutorial (9)
Core Java Tutorials (2
Hadoop & Spark Tuto
JEE Tutorials (19)
Scala Tutorials (1)
Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

## 100+ Java pre-interview coding tests

open all | close all
Can you write code? (
♦ Complete the given
Converting from A to I
Designing your classe
Java Data Structures
Passing the unit tests
What is wrong with th
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

## How good are your .....?

- The producers are throttled to add elements if the consumers get too far behind.
- If the Queue capacity is limited, the memory consumption will be limited as well.

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

**Q9.** What are differences between Array and List?
**A9.**

- Array is a fixed length data structure whilst a List is a variable length Collection class. List allows you to add and subtract elements even it is an O(n) operation in worst case scenario.
- An array can use primitive data types or objects, but the List classes can only use objects.
- Arrays are inflexible and do not have the expressive power of generic types.
- List gives you the data abstraction as you can swap ArrayList, LinkedList, CopyOnWriteArrayList, etc depending on the requirements.

**Q10.** What is the difference between Comparable and Comparator interface
**A10.** The Comparable interface and provides a compareTo(..) method to be called while sorting **naturally**(i.e.by default). You can define your own ordering (i.e. **custom**) logic through the compare(…) method by implementing the Comparator interface.

**Q11.** What is the difference between ArrayList and Vector?
**A11.** Vector, Stack, and Hashtable are legacy data structures and must not to be used. All methods in these classes are synchronized (i.e. coarse grained lock), hence not efficient. Favor the concurrent data structures for concurrent read and single write.

## Java 8

**Q12.** What is the difference between with and without lambdas to collections?
**A12.** Lambdas introduced in Java 8 would be worthless if we didn't have any means for applying lambdas to collections.

So, “deafult methods” were introduced to Java interfaces, which has the benefit that default methods don’t break the implementations. In other words, interfaces in Java 8 onwards can now implement methods. So, filetr, map, reduce, forEach, etc default methods are now added to the “java.util.stream.Stream” interface.

**Q13.** What is the difference between having the package java.util.stream and not having java.util.stream?
**A13.** The new java.util.stream package has been added to Java 8 to allow us to perform filter, map, and reduce operations with the help of lambda expressions on the collection classes. For example,

```
List<Person> persons = constructPersons(...);
Stream<Person> personsOver16 = persons.stream().f

```

## Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

765 views

18 Java scenarios based interview Questions and Answers

399 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

**279 views**

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

**239 views**

001B: ♦ Java architecture & design concepts interview questions & answers

**201 views**

Bio | **Latest Posts**

## Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

**About** Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ ♦ Part 4: Badly designed classes & interfaces

♦ Finding the perfect number ›

**Posted in** Differences Between X & Y, member-paid

# Leave a Reply

Logged in as geethika. Log out?

**Comment**

Post Comment

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

### Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

## Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑