Java Interview Question Bank

Fresher Academy

GENERAL

1. Giới thiệu bản thân (tiếng anh+ tiếng việt)
2. Nhận xét các bạn vừa mới phỏng vấn
3. Điều gì làm em khác biệt với những bạn khác

JAVA CORE

1. Các kiểu dữ liệu nguyên thủy:

   Kiểu số nguyên: Integer Type – byte (1), short (2), int (4), long (8)

   Kiểu só thập phân: Floating point – Float(4), Double(8)

   Kiểu ký tự: Char, String

   Kiểu logic: giá trị kiểu true false
2. Kiểu dữ liệu tham chiếu/đối tượng
3. Difference between int & INTEGER.
4. Autoboxing và Unboxing trong Java là gì?
5. Trong java có các loại Access Modifier nào? Phạm vi truy cập ra sao?
6. Could you describe about "Strong typed"?
   - Check variables at compile time
   - Weak typed: check variables at runtime (script languages such as: JavaScript, PHP…).
7. What does "static" keyword mean?
   - Class resources
   - Used for method, attributes, inner class.
   - Available for all objects.
8. Describe the principles of OOPs.
   - Abstraction
   - Encapsulation
   - Inheritance
   - Polymorphism.
9. Explain about Polymorphism.
   - One name many forms
   - Override, overload methods.
   - Increase flexibility.

<u>10.</u> Explain about Inheritance.

- Increase reusability
- Extends class, implements interface.
- Is – a relationship.

<u>11.</u> Explain about Encapsulation.

- Hiding information and data.
- Use access modifier(public, protected, private)
- Make the system more modularized.

<u>12.</u> Explain about the different forms of Polymorphism?

- Overriding
- Overloading
- Anonymous class.

<u>13.</u> What is the difference between method overloading and method overriding?

- Java method has five elements: modifiers, return types, names, parameters, exceptions.

a. Overloading method:

- Same names
- Others are flexible

b. Overriding:

- Same names
- Same parameters (number and type, order)
- Access modifier is less restrict
- Return type: same type or covariant type. (equal or narrower)
- Exception: Checked exception (equal or narrower); flexible runtime exceptions

<u>14.</u> What is dynamic binding?

- Binding: Association btw reference and actual object.
- Binding at runtime (Overriding method).
- Static binding: at compile time.

<u>15.</u> Explain about "Abstraction"?

- Increase extendability.
- Increase abstraction of layered architecture.
- Use interface or abstract class.

<u>16.</u> Could you explain "composition" and "inheritance" in JAVA?

- Composition: Has – a relationship. (Famous example: Object Adapter pattern)
- Inheritance: Is – a relationship. (Class adapter pattern).

<u>17.</u> Exception handling with composition and inheritance?

- Inheritance:
    - An overriding method can throw any uncheck exceptions

  - An overriding method can throw narrower or fewer exceptions than the overridden method.

- Composition:
  - Use try-catch block or throws exception when re-use method which throws exception

18. What are differences between abstract class and interface?

- Implementation
- Characteristics of method and attribute
- Purpose of using.

a. Abstract class:
  - Single inheritance with “extends” key word
  - Could have both abstract and concrete methods. Attributes are normal as
  - normal classes.
  - Use when we want to have common behaviors for subclasses.

b. Interface:
  - Support for multiple inheritance.
  - Have only abstract methods
  - Provide the contract.

19. How can we obtain an array from an ArrayList class?

- Use toArray() method.
- Arrays.asList(array). (Vice-versa. Note: cannot remove element from list)

20. What is garbage collection? Can we enforce garbage collection to perform?

- Java garbage collection is the process by which Java programs perform automatic memory management.
- JVM mechanism for collecting unused objects and removing them.
- Purpose: optimize and save memory.
- Your best option is to call *System.gc()* which simply is a hint to the garbage collector that you want it to do a collection. There is **no way** to force and immediate collection though as the garbage collector is non-deterministic.

21. What meaning of String immutable? Can you explain the concept?

- When modifying a String, a new String object is created in memory, stored in the String pool and the instance refers to the new object.

22. Describe the basic steps to reverse a string?

- Opt1: When modifying a String, a new String object is created in memory, stored in the String pool and the instance refers to the new object.
- Opt2: Use the built in reverse() method of the StringBuilder.
- Opt3: Convert the input string into the character array by using toCharArray() built in method. Then add the characters of the array into the LinkedList object. Use built in reverse() method for the Collections class.

23. What is Pass by Value and Pass by reference? Does Java support both of them?

- Pass by value:
    - Pass only the bit-pattern (copy) of value.
    - Method can’t change the variable value.
- Pass by reference:
    - Receive a pointer of variable.
    - Java only supports Pass by value

24. What is the difference between error and exception?

- Error:
    - Irrecoverable condition occurred at run-time
    - Can’t repair at run-time
    - Eg: OutOfMemory
- Exception:
    - Caused by bad input
    - Can handle
    - Eg: NullPointerException, IndexOutOfBoundException…

25. Kể các collection biết. Phân biệt list và linklist. Add dãy 10 số từ 0 -10 thì nên dùng cái nào?

26. What are differences between ArrayList and Vector?

- ArrayList:
    - No synchronization
    - Increase 50% capacity.
- Vector:
    - Synchronization
    - Double capacity when full size.

27. What is Iterator? How to use it?

- A Java interface for traversing through collection.
- Methods: hasNext(), next(), remove();

28. When you use Iterator?"

- Traverse through a collection.
- Make a copy of collection data.
- No effects to the collection.

29. What equals() and hashCode() method respond for?

- equals() method:
    - Compare logically two objects.
- hashCode() method:
    - An integer number associcated with the objects using for storing and retriving in demands.

- Both methods are useful when we want to store objects in hash collection or set duplicate elements.

30. How and when override them?

- equals() method:
  - public boolean equals(Object obj){} → must pass Object type
  - Check null -> check instanceof -> compare properties.
- hashcode() method:
  - public int hashCode(){}
  - Based on attributes we implement an agorithm to generate distinct numbers.

31. What is the difference between equals() and "==" ?

- equals(): Compare logically (data).
- "==" : Compare address (reference).

32. What are differences between Comparator and Comparable?

- Comparable:
  - Override compareTo(Object obj)
- Comparator:
  - Act as the third party
  - Override compare(Object obj1, Object obj2)

33. Comparable interface? When to use them?

- Comparable: implement to compare an object itself with another.
- Use:
  - Avoid duplicate elements on Set
  - Sort collections or array by using Collections.sort(collection) and Arrays.sort(array)

34. Is it possible to use multiple comparator?

- Yes
- With each criterion, we have implement Comparator interface

35. What are differences between HashMap and HashTable?

- HashMap:
  - No synchronization
  - Allow one null key and many null values
- HashTable:
  - Synchronization
  - Don't allow null key and null values.

36. What are differences between HashMap and TreeMap?

- HashMap:
  - Don't guarantee the order of keys.

- TreeMap:
  - Implements SortedMap interface
  - Order of keys is sorted.

37. How to make a Hashmap thread-safe?

- Use ConcurrentHashMap
- HashMap is a non-threadsafe Map which should not be used by multiple threads.
- Hashtable is a thread-safe Map that allows only one thread to execute a read/update operation at a time.
- And ConcurrentHashMap is a thread-safe Map with greater flexibility and higher scalability as it uses a special locking mechanism that enables multiple threads to read/update the map concurrently.

38. What are differences between List and Set?

- List:
  - Support random access by index
  - Allow storing duplicate elements.
- Set:
  - Don’t support random access
  - No duplicate elements.

39. How to sort a list?

- Implements Comparable -> Use Collections.sort();
- Implements Comparator -> Use Collections.sort(list, comparator);

40. How to check duplicated elements in the Set?

- Override equals() and hashCode().
- Wrong implementation of equals() can lead to memory leak problem.

41. How to find common elements in two sets?

- Solution 1: Iterate two sets then check in loops one by one
- Solution 2: Move elements to two lists then sort lists -> check common element with an efficient algorithm.

42. How to find + remove duplicated elements in a list?

- Solution 1: Convert it to a set then set contains no duplicate objects.
- Solution 2: Sort the list then compare continuous objects faster.

43. Can you explain TreeSet? HashSet?

- TreeSet:
  - Implements SortedSet interface.
  - Use a tree for storage.
  - Elements are sorted.
- HashSet:
  - Extends AbstractSet interface.

  - Use hash table for storage.

44. What are differences between Arrays and ArrayList?

- Arrays:
  - Fixed size
  - Data type: primitive, objects.
  - Dimension: multi-dimension array.
- ArrayList:
  - Dynamic size.
  - Data type: only object.
  - Dimension: No.
  - Support Generics from Java 5.

45. Have you ever worked with MultiMap?

- MultiMap:
  - Component of Guava framework.
  - One key, multiple values.
  - get(key) return a list of values.

46. What's the LinkedList? When to use LinkedList?

- LinkedList:
  - Provide linked list data structure.
  - Use large memory (for references).
  - Efficient for inserting or deleting.
  - Not efficient for random access as a normal list.

47. What are differences among String, StringBuilder and StringBuffer?"

- Immutability:
  - String is immutable.
  - StringBuffer and StringBuilder are mutable.
- Synchronization:
  - StringBuilder is not synchronized.
  - StringBuffer is synchronized.

48. What are differences between Deep copy and Shallow copy?

- Deep copy:
  - Duplicate everything (Collection: structure + elements).
- Shallow copy:
  - Copy as little as possible. (Collection: only structure + shared elements).

49. How do we implement Shallow cloning?

- Implements Cloneable interface
- Override clone().

How do we implement Deep cloning? (2 ways)

- Solution 1: Implements Cloneable interface for all elements.
- Solution 2: Serialization. (Serialize and deserialize).

50. Define exceptions?

- Extends Exception class.

51. Can you explain in short how JAVA exception handlings work?

- Use try-catch block, finally, “throws”, “throw” keywords to handle exceptions.
- Code in finally block always execute, use for cleaning code.

52. Can you explain different exception types?

- Checked exception
  - Invalid condition out of program’s control
  - Check at compile-time
- Unchecked exeption
  - Check at run-time
  - Defects (bugs) in programs

53. What is serialization?

- Proccess to convert object to byte-stream for transferring through network or writing to disk.

54. How do we implement serialization actually?

- Implement Serializable interface.
- Use writeObject() and readObject() to serialize and deserialize

55. What's the use of Externalizable interface?

- It is externalilizing your serialization.
- Purpose: to increase performance in some specific situations.
- Use readExternal() and writeExternal() to read from stream and write object into stream.

56. What's difference between thread and process?

- Thread:
  - Path of execution run on CPU, light weighted process
  - Related threads share same data memory
  - Have their own individual stacks
- Process:
  - Collection of threads shared the same virtual memory
  - Every process has its own data memory location

57. What is thread safety and synchronization?

- Thread safe:
  - A method that can run safely in multithread environment without any resource confliction.

- Synchronization:
    - Assure resources (variable, object, method…) are not accessed by multiple threads at the same time

<u>58.</u> What is semaphore, mutex?

- Java multi threads example to show you how to use Semaphore and Mutex to limit the number of threads to access resources.
    - <u>Semaphores</u> – Restrict the number of threads that can access a resource. Example, limit max 10 connections to access a file simultaneously.
    - <u>Mutex</u> – Only one thread to access a resource at once. Example, when a client is accessing a file, no one else should have access the same file at the same time.

<u>59.</u> What is deadlock? How do you detect them? Do you handle them? And finally, how do you prevent them from occurring?

- Lock: multiple processes access same resource at the same time
- Deadlock: two thread waiting another in a cycle

<u>60.</u> How do we create threads? (2 ways)

- Extends Thread class
- Implements Runable interface

<u>61.</u> What's difference between in using Runnable and Thread?

- Thread:
    - A class
    - Use when a class not extending another class
    - A thread has unique object instance associated with
- Runnable:
    - An interface
    - Use when a class already extending another class
    - Many threads share the same object instance

<u>62.</u> How to implement thread safety? (2 ways)

- Use “synchronized” with a block of code
- Use “synchronized” with the method

<u>63.</u> Let say we have 2 threads: A and B. Is there any way to be sure that thread A will execute before the thread B?"

- setPriority() in Thread class. Not guarantee A go first
- suggest: use join() method.

<u>64.</u> Can you explain the wait() and notify() and notifyAll() method?

- wait()
    - A thread gives up its hold on the lock, goes to sleep
- notify()
    - A thread wakes up and tries to acquire the lock again

65. How to monitor/manage threads? How to monitor JVM performance? JVM tuning?/ What tools do you use to check memory?

- Use JConsole and VisualVM
- VisualVM:
    - Display real-time, high-level data

66. What is Stack and Heap Memory?

- Heap:
    - Stores class instance + arrays
    - Shared memory
- Non-heap:
    - ‘method area’
- Stack memory:
    - Allocate automatic variable in function

67. How could you solve the memory leak?

- Use good Java best practices
- Consider static resources, set empty collections...
- Minimize the variable scopes
- Use tools to check before release applications

68. Difference between Iterator and Enumeration

- Iterator:
    - can get remove() from Iterator to remove an element.
    - can also add and remove the objects.
- Enumeration:
    - Enumeration is for read-only access.
    - Enumeration you can only traverse and fetch the objects.

69. Write a java program to count number of unique words separated by comma (,) and their occurrence from text file.

70. Difference between sleep and wait. Which is better?

- wait() method:
    - The wait() is used for multi threaded **synchronization**, where single resource is shared among multiple thread.
    - Class belongs: java.lang.Object
    - Releases the lock
    - Awake by call to notify() or notifyAll() method
- sleep() method:
    - the thread actually needs a delay in the background.
    - Class belongs: java.lang.Thread.
    - Does not release the lock for specified time or until interrupt.

- Wake up Condition: when time expires or due to interruption.

71. What is final modifier?

- final variable:
  - cannot change the value;
- final method:
  - cannot override it.
- final class:
  - cannot extend it.

72. How to swap two numbers without using temporary variable?

73. Write a program for Bubble Sort in java.

74. Write a program for Insertion Sort in java.

75. Write a program to find the second largest number.

76. How to find middle element of the linked list?

- In order to find middle element of linked list, you need to maintain two-pointer
  - one increment at **each node** while other increments after **two nodes** at a time.
  - when first pointer reaches end, second pointer will point to middle element of linked list.

77. How to find if linked list has a loop ?

- Can use two pointer approach to solve this problem.
  - Increment 'one node' pointer before the processing 'two nodes' pointer;
  - We are likely to find a situation where both the pointers will be pointing to same node. This will only happen if linked list has loop.