

# 200 common Java interview questions, with reference answers

If you are going to a Java interview, preparing for Java programming language questions is necessary. These Java interview questions are compiled from Tutorialspoint and are accompanied by reference answers.

If you are going to a Java interview, preparing for Java programming language questions is necessary. These Java interview questions are compiled from Tutorialspoint and are accompanied by reference answers. Good employers often don't come up with questions to ask, they often start with basic concepts about the topic and then rely on your answers to come up with the next discussion. The answer is quite concise and compact, so if you want to impress an employer, you must have a strong knowledge base for Java to implement further when asked to explain it thoroughly. Don't worry, the most important factor in your interview is still confidence, believe in your ability, combined with this questionnaire, make sure you get a successful interview. .

## 1. What do you know about Java?

Java is a high-level programming language, first developed by Sun Microsystems and published in 1995. Java runs on diverse platforms, such as Windows, Mac OS, and diverse UNIX versions.

More reference: [What is Java? Why choose Java?](#)

## 2. Which platforms are supported by the Java programming language?

Java is a cross-platform programming language that can run on Windows, Mac OS, and various UNIX versions such as HP-Unix, Sun Solaris, Redhat Linux, Ubuntu, CentOS, .

## 3. List 5 any features of Java?

Some characteristics of Java are Object Oriented, Platform Independent, Interpreter, Multi-thread, robust.

## 4. Why is Java independent of structure?

Its compiler creates a structure-independent file format, making the compiled code executable on any processor, in the presence of the Java runtime system.

## 5. How high performance (High Performance) is enabled in Java?

Java uses the Just-In-Time compiler to enable high performance. This compiler turns Java Bytecode (a program containing instructions that need to be interpreted) into directives that can be sent directly to the processor.

## 6. Why is Java considered dynamic (Dynamic)?

It is designed to adapt to a growing environment. Java programs can carry a large amount of information at runtime that can be used to test and process access to objects at runtime.

### **7. What is Java Virtual Machine and what role does it play in Java's platform independence?**

When Java is compiled, it is not compiled into a specific machine platform, but is translated into platform-independent Bytecode. This bytecode is distributed through the Web and interpreted by the Java Virtual Machine (JVM) on any platform it is running.

### **8. List two JDE of Java?**

Netbeans, Eclipse, . (let's say more JDE or talk a little more detail about these 2 JDE to score with employers).

### **9. List some keywords in Java (unlike keywords in C, C ++)?**

Some keywords in Java are import, super, finally, .

### **10. What do you understand about the subject?**

The object is an entity at runtime, its state is stored in the fields and the behavior is expressed through methods. Methods operate on the intrinsic state of the object and act as the main mechanism for communicating between objects.

### **11. Class definition (class)?**

A class is a design (blueprint) from which individual objects are created. A class can contain fields and methods to describe the behavior of an object.

### **12. What types of variables can a class contain?**

A class can include local variables, instance variables, and class variables.

### **13. What is the local variable or local variable?**

Variables defined inside methods, constructors or blocks are called local variables. This variable will be declared and initialized inside the method and it will be destroyed when the method ends.

### **14. What is an instance variable?**

Instance variables are variables inside a class but outside any method. These variables are initialized when the class is loaded.

### **15. What is class variable?**

These are variables declared with a class, outside any method, with the static keyword.

### **16. What is the Singleton class?**

The Singleton class in Java controls object creation, limiting the number of objects to one, but it is also flexible to allow you to create more objects if the situation changes.

### **17. What do you understand about Constructor?**

Constructor is called when a new object is created. Each class has a Constructor. If we don't explicitly write constructors for the class, then the Java compiler will build a default constructor for that class.

### **18. List three steps to create an object for a class?**

First, an object is declared, then initialized and finally launched.

### **19. Default value of byte data type in Java?**

The default value of the byte data type is 0.

### **20. Default value of data type float and double in Java?**

The default value of data types float and double in Python is different from in C / C ++. The default of float is 0.0f and the double is 0.0d.

### **21. When is the byte data type used?**

This data type is used to store space in wide arrays, primarily to replace integers, because a byte is 4 times smaller than an int.

### **22. What is static variable (static variable)?**

Class variables are also known as static variables (static variables) that are declared with the static keyword in a class, but outside a method, constructor, or block.

### **23. What do you understand about Access Modifier?**

Java provides several Access Modifiers to set the access level for classes, variables, methods and constructors. A member has default access when no Access Modifier is specified.

### **24. What is Protected Access Modifier?**

Variables, methods and constructors, which are declared protected in a superclass, are only accessible by parent classes in another package or any class within that class of the protected class.

### **25. What is Synchronized Non Access Modifier?**

Java provides Synchronized Non Access Modifier to bring features other than Access Modifier, from Synchronized indicating that a method is only accessed by one thread at a time.

### **26. Under the precedence of operators in Java, which operator has the highest priority?**

Postfix operators like (), [], . is the highest priority.

### **27. What types of variables can be used in the switch command?**

The variables used in a switch command can only be one byte, string, enum, short, int or char.

**28. When can the parseInt () method be used?**

This method is used to retrieve the original data type of a given string.

**29. Why is String class considered immutable?**

String class is invariant or immutable, and when it has been created, a String object cannot be altered. Because String is immutable, it can be safely shared between multiple threads. This is a very important part for multi-threaded programming.

**30. Why is StringBuffer called variable?**

StringBuffer objects can be modified. If there is a situation where you need to make multiple modifications to strings, use StringBuffer.

**31. What is the difference between the StringBuffer and StringBuilder classes?**

Use StringBuilder whenever possible because it's faster than StringBuffer. But, if thread safety is needed, you should use StringBuffer objects.

**32. Which packages are used for pattern matching (Pattern Matching) with Regular Expression in Java?**

For this purpose, you use package java.util.regex.

**33. Which java.util.regex classes?**

java.util.regex consists of three classes: Pattern class, Matcher class and PatternSyntaxException class.

**34. What does finalize () do?**

You can define a method that will be called immediately before destroying the object by the Garbage Collector. This method is called finalize (), and it can be used to ensure that an object is completely finished.

**35. What is Exception?**

An Exception is an issue created during the execution of a program. The Exception captured by Handler is determined along with the Thread method call.

**36. What do you know about Checked Exception?**

The feature of this Exception is a user error or an unforeseen problem by the developer. For example, if a file has been opened, but the file cannot be found, an Exception will appear. These exceptions cannot be simply ignored at compile time.

**37. Explain Runtime Exception?**

It is an Exception that can be avoided by programmers. In contrast to Checked Exception, Runtime Exceptions are ignored at compile time.

**38. What are the two classes of Exception Class?**

The Exception class has two main subclasses: IOException class and RuntimeException class.

### **39. When will the keyword throws be used?**

If a method doesn't handle a Checked Exception, the method must be declared with the keyword throws. Keywords throws appear at the end of a method.

### **40. When does the keyword throw be used?**

An Exception can be thrown, either by initialization or an Exception you just captured, by using the keyword throw.

### **41. How is finally used under Exception Handling?**

The finally keyword is used to create a block of code that follows a try block. A finally block always executes, whether or not an Exception appears.

### **42. What should you keep in mind when creating your own Exception in Java?**

While creating my own Exception:

1. All Exception must be a Throwable child.
2. If you want to write a Checked Exception that is automatically followed by Handler or Declare Rule, you need to inherit the Exception class.
3. If you want to write a Runtime Exception, you need to inherit the RuntimeException class.

### **43. What is Inheritance?**

It is a process in which an object obtains attributes of another object. Using inheritance, you can more easily manage information created in a hierarchical structure.

### **44. When to use super keywords?**

If the method overrides one of the methods of the superclass, the overridden method can be recalled using the super keyword. It can also be used to reference a hidden field.

### **45. Definition of polymorphism?**

Polymorphism is the ability to help an object with many forms. In OOP, the most common use of polymorphism is when a parent class reference is used to refer to a subclass object.

### **46. ??What is abstraction?**

It involves the ability to create an abstract class (abstract class) in OOP. It helps minimize complexity and also improves system retention.

#### **47. What is Abstract class?**

These classes cannot be initialized and deployed either partially or not. This class contains one or more abstract methods, where the method declaration is simplified with no body.

#### **48. When is the abstract method used?**

If you want a class to contain a specific method but you want the actual implementation of that method to be decided by subclasses, you can declare the method in the superclass in abstract form.

#### **49. What is Encapsulation?**

It is a technique for creating fields in a private class and providing access to fields via public methods. If a field is declared private, it cannot be accessed by any method outside that class, from which to hide the fields inside the class. Therefore, closure is also referred to as Data hiding.

#### **50. Main benefits of closure?**

The main advantage of closure is the ability to modify your deployed code without destroying someone's code. It is like a code protection code and avoids your code and data from being accessed randomly from outside code. Close-up provides code retention, flexibility, and scalability.

#### **51. What is an interface?**

An interface is a collection of abstract methods. One class implements an Interface, which inherits the abstract methods of that Interface.

#### **52. Some characteristics of Interface?**

Include:

1. Interface cannot be initialized.
2. An interface does not contain any constructors.
3. All methods in an Interface are Abstract.

#### **53. In Java, what is Package?**

A Package can be defined as a group of related types (classes, interfaces, enumerations), providing access protection and namespace manager.

#### **54. Why is Package used?**

Package is used in Java to prevent naming conflicts, to control access, to find and locate, and to use classes, Interface, enumeration . more easily.

### **55. What do you understand about Multi-Thread?**

A multithreaded program consists of two or more parts that can run simultaneously. Each part of that program is called a Thread, and a Thread defines a different executable.

### **56. How can a thread be created in two ways?**

Thread can be created by: Runnable Interface deployment, inheriting Thread class.

### **57. What is an Applet?**

An Applet is a Java program that runs in a Web browser. An Applet can be a full-featured Java application because it has the entire Java API in its layout.

### **58. What class does an Applet inherit?**

An Applet inherits the java.applet.Applet class.

### **59. Explain Garbage Collector in Java?**

Java uses a garbage cleaner to free up memory. By deleting objects that are no longer used by any program.

### **60. Definition of objects cannot be changed?**

An immutable object cannot be changed since it was created.

### **61. Explain the use of this () with Constructors?**

It is used with variables or methods and is used to call the Constructor of the same class.

### **62. What do you know about Set Interface?**

It is an element set that cannot contain duplicate elements. Set Interface only includes methods that are inherited from Collection and add limits to prohibit the duplicate element from appearing.

### **63. Presenting TreeSet?**

It is a Set deployed when we want elements in a sorted order.

### **64. What is Comparable Interface?**

It is used to sort collections and object arrays using collection.sort () and java.util methods. Objects of class Comparable Interface implementation can be sorted.

### **65. What is the difference between throws and throw?**

Include:

1. Throw is used to trigger an Exception while throws are used in Exception's declaration.

2. Without throw, Checked Exception cannot be processed, while throws are used to denote Exception not handled by the function.

### **66. Explain the following code in Java language?**

```
public static void main (String args[ ])
```

Below is a detailed analysis:

1. public - it is Access Specifier.
2. static - it allows main () to be called without initializing a specific instance of a class.
3. void - it tells the compiler that no value is returned by main ().
4. main () - this method is called at the beginning of the Java program.
5. String args [] - args parameter is an instance array of String class.

### **67. What is JRE (Java Runtime Environment)?**

The JRE is a deployment of Java Virtual Machine that executes Java programs. It provides the minimum conditions for executing a Java application.

### **68. What is JAR file?**

JAR stands for Java Archive and it combines multiple files into one. It holds Java classes in a library. JAR file is built in ZIP file format and has a .jar extension.

### **69. WAR file is what?**

This is the Web Archive File and is used to store XML, Java classes, and JavaServer pages, which are used to distribute a set of JavaServer Page, Java Servlet, Java classes, XML files, static Webpage, and so on.

### **70. JIT Compiler definition?**

It improves the runtime performance of Bytecode-based computer programs.

### **71. What is the difference between object-oriented programming language and object-based programming language?**

Object-based programming languages ??have all the characteristics of OOP except Inheritance. JavaScript is an example of Object-based Programming Language.

## **72. The purpose of the default Constructor?**

Java Compiler creates a default Constructor only if there is no constructor in the class.

## **73. Can a Constructor be created as a final?**

No, this is impossible.

## **74. What is a static block?**

It is used to initialize static data members. It is executed before the main method at the time of loading that class.

## **75. Definition of Composition?**

Keeping references of other layers within several other layers is known as Composition.

## **76. What is Method Overloading?**

If a class has multiple functions with the same name but has different parameters, it is treated as Method Overloading or Function Overloading.

## **77. What is method override (Method Overriding)?**

If a subclass provides a specific implementation of a method that is provided by its parent class, then it is Override the method.

## **78. What is the difference between overwriting and overwriting (Overloading vs Overriding)?**

Overload method to increase readability of the program. Overriding the method provides a specific implementation of the method that is provided by its parent class. The parameter must be different from the overloaded type, and in the override, the parameters are the same type.

## **79. What is the final class?**

Final classes are created so that methods implemented by that class cannot be overwritten. It cannot be inherited.

## **80. What is NullPointerException?**

A NullPointerException is thrown when calling the instance method of a Null object, accessing or modifying the fields of a Null object, .

## **81. How can a Thread enter a waiting state?**

A Thread can enter the Waiting state by summoning its sleep () method, either by locking on IO, or failing to attempt to acquire the object's lock, or by summoning. the wait () method of the object. It can also enter a wait state by summoning its suspend () method (this method is old).

## **82. How does Multi-Thread happen on a computer with a single CPU?**

Scheduler of the operating system allocates execution time for Tasks. By quickly switching between executing tasks, it gives us the impression that these Tasks are executed simultaneously.

### **83. What is the run () method of a Thread?**

After a Thread is created, through its start () method in the Thread class, the JVM summons the Thread's run () method when this Thread starts executing.

### **84. Is there a problem with the order of the catch commands written for FileNotFoundException and IOException?**

Have. FileNotFoundException is inherited from IOException. The subclasses of Exception must be captured first.

### **85. What is the difference between yield and sleep?**

When a task summons its yield () method, it becomes ready. When a task summons its sleep () method, it switches to the wait state.

### **86. Why is Vector class used?**

The Vector class provides the ability to deploy an expandable array of objects. Vector is useful if you don't know the size of the array in advance, or if you only need an array that can be resized during a program's lifetime.

### **87. How many bits are used to represent Unicode, ASCII, UTF-16 and UTF-8 characters?**

Unicode requires 16 bits and ASCII requires 7 bits. Although ASCII character set only uses 7 bits, it is usually represented by 8 bits. UTF-8 represents characters using 8 bits and is 16 bits for UTF-16.

### **88. What are Wrapper classes?**

These are the classes that allow the original data types to be accessed as objects. For example: Integer, Character, Double, Boolean, .

### **89. What is the difference between a Window and a Frame?**

The Frame class inherits Window to define a main application window that may have a bar menu.

### **90. Which packages have lightweight components?**

That is javax.Swing. All components in Swing, except JApplet, JDialog, JFrame and JWindow, are lightweight components.

### **91. What is the difference between the paint () and repaint () methods?**

The paint () method supports drawing through a Graphics object. The repaint () method used to make the paint () method can be summoned by AWT Thread.

### **92. What is the purpose of the File Class?**

It is used to create objects that provide access to local file system files and directories.

### **93. Differences between Reader / Writer class structure and class structure InputStream / OutputStream.**

The Reader / Writer class structure is character-oriented, and the class structure of InputStream / OutputStream is byte-oriented (Byte-oriented).

**94. What class do you use to get design information about an object?**

Class Class is used to obtain information about the design of an object and the representation of the java.lang.Class class represents classes, Interface in a running Java application.

**95. What are the different static and non-static variables?**

Static variables (static variables) are bound to the entire class rather than an instance of a class. Non-static variables get unique values ??with an object representation.

**96. What is Serialization and Deserialization?**

Serialization is the process of writing the state of an object to a Byte Stream. Deserialization is the process of recovering these objects.

**97. What is a case?**

It is part of the process of analyzing a program and describes a situation that a program may encounter and which program should take action in that situation.

**98. Explain the use of subclasses in a Java program?**

Subclasses inherit all public and protected methods and deploy deployment. It also inherits all the default modifier methods and their implementers.

**99. How to add menushtcut to menu item?**

If there is an instance of the button called b1, you can add the menushcut by calling the b1.setMnemonic method ('F'), from which the user can use the shortcut Alt + F to click the button. there.

**100. Can you write a Java class that can be used as an Applet as well as an application?**

Yes, just add a main () method to the Applet.

**101. What is the difference between Swing and AWT components?**

The AWT components are heavy-weight, while the Swing components are lightweight. Heavy components depend on the Local Windowing Toolkit. For example, java.awt.Button is a heavy component, when it is running on the Java platform for the Unix platform, it maps to Motif Button really.

**102. What is the difference between Constructor and other methods?**

The constructor must have the same name as the class name and cannot return a value. They are only called while normal methods can be called multiple times.

**103. Are there any limitations when using Inheritance?**

Yes, when inheritance is inherited from both parent and interface classes, sometimes it can generate error-prone with dynamic overwriting and dynamic loading in some situations.

#### **104. When is ArrayStoreException thrown?**

When copying elements between different arrays, if the source parameter or the destination parameter is not arrays or their types are incompatible, then an ArrayStoreException will be thrown.

#### **105. Can you call this Constructor from Constructor other than a class with multiple Constructors?**

Yes, use this () syntax.

#### **106. What is the difference between sleep () and wait () methods?**

For example, sleep (2000); Make Thread wait 2 seconds correctly. While wait (2000); It can take up to 2 seconds for the Thread to wait. A Thread may stop waiting earlier if it receives a notify () or notifyAll () call. The wait () method is defined in the Object class and the sleep () method is defined in the Thread class.

#### **107. When was ArithmeticException thrown?**

ArithmeticException ???c ném khi chia s? nguyên cho s? 0 ho?c l?y ph?n d? c?a phép chia cho s? 0. Nó không bao gi? ???c ném trong các phép toán v? s? th?c.

#### **108. M?t bi?n là transient (t?m th?i) là gì?**

M?t bi?n transient là m?t bi?n mà không th? ???c x?p theo th? t? trong Serialization và nó ???c kh?i t?o b?i giá tr? m?c ???nh c?a nó trong Deserialization.

#### **109. Synchronization (??ng b? hóa) là gì?**

Synchronization là kh? n?ng ?i?u khi?n truy c?p c?a nhi?u Thread t?i ngu?n ?ã chia s?. T? khóa synchronized trong Java cung c?p locking ?? ?m b?o s? truy c?p t??ng h? mang tính lo?i tr? c?a ngu?n ?ã chia s? và ng?n c?n Data Race (Tranh ?o?t d? li?u).

#### **110. Collection API là gì?**

Collection API là m?t t?p h?p các l?p và Interface mà h? tr? các ho?t ??ng trên các Collection c?a ??i t??ng.

#### **111. Trình d?n rác có b?o ??m r?ng m?t ch??ng trình s? không ch?y h?t b? nh??**

Trình d?n rác không b?o ??m r?ng m?t ch??ng trình s? không ch?y h?t b? nh?. Nó là có th? ?? cho các ch??ng trình s? d?ng h?t ngu?n b? nh? nhanh h?n vi?c chúng b? thu th?p b?i trình d?n rác. Các ch??ng trình c?ng là có th? t?o các ??i t??ng mà không ph? thu?c vào trình d?n rác.

#### **112. L?p cha g?n nh?t (tr?c ti?p) c?a l?p Applet?**

Panel là l?p cha g?n nh?t (tr?c ti?p). M?t panel cung c?p không gian trong ?ó m?t ?ng d?ng có th? ?ính kèm b?t c? thành ph?n nào, bao g?m các panel khác.

#### **113. Toán t? nào trong Java là Right-Associative?**

Toán tử = là Right-Associative. (Nếu bạn chưa hiểu right và left associative, bạn theo dõi: Toán tử = cho phép bạn thực hiện nhiều phép gán trong cùng một lệnh. Ví dụ: a=b=c=d=99;)

#### 114. Hai khác nhau giữa lệnh break và lệnh continue?

Lệnh break chấm dứt một lệnh mà nó áp dụng (switch, for, do, hoặc while). Một lệnh continue tiếp tục s? d?ng tiếp k? thúc vòng lặp tiếp theo và trở về? lại khi nào cho lệnh vòng lặp.

#### 115. Nếu một biến tiếp khai báo là private, thì nó có thể tiếp truy cập trong phạm vi nào?

Biến private chỉ có thể tiếp truy cập bên trong lớp mà nó tiếp khai báo.

#### 116. Mục đích của lớp System?

Mục đích của lớp System là cung cấp truy cập tới ngôn ngữ.

#### 117. Liệt kê các kiểu g?c trong Java?

Có 8 kiểu dữ liệu g?c trong Java là byte, char, short, int, long, float, double và Boolean.

#### 118. M?i quan hệ giữa Clipping và Repainting đối với AWT?

Khi một cửa sổ tiếp repaint bởi AWT Thread, nó thi?t lập các khu vực Clipping thành khu vực cửa cửa sổ mà cần Repainting.

#### 119. Lớp nào là lớp cha g?n nh?t (tr?c tiếp) của lớp Container?

Lớp Component là lớp cha g?n nh?t.

#### 120. Lớp Exception nào tiếp t?o bởi Java runtime?

Java runtime tạo các RuntimeException và Error.

#### 121. Đối với các kiểu kiến nào thì một ph?ng thức finalize() của tiếp tiếp?ng tiếp tri?u h?i bởi trình d?n rác (Garbage Collector)?

Trình d?n rác tri?u h?i một ph?ng thức finalize() của tiếp tiếp?ng khi nó phát hiện rằng tiếp tiếp?ng đã thi?t b?i.

#### 122. Một Thread đã ch?t có thể tiếp restart lại nh? thì nào?

Một Thread đã ch?t không thể restart lại tiếp.

#### 123. Các toán tử s? h?c nào có thể gây ra s? kiến ném một ArithmeticException?

Phép chia / và lấy phần dư % s? nguyên có thể gây ra s? kiến ném một ArithmeticException.

#### 124. Biến kiểu Boolean tiếp t? tiếp?ng khi t?o với giá trị?

Giá trị mặc định của kiểu Boolean là false.

#### 125. Các lệnh try có thể l?ng nhau không?

Have.

## 126. ClassLoader là gì?

ClassLoader là một lớp mà mọi ứng dụng và các thư viện khác đều sử dụng. Lớp ClassLoader là một lớp abstract.

## 127. Chúng khác nhau gì? giữa Interface và một lớp Abstract?

Một lớp Abstract là một lớp có thể có các phương thức instance mà triển khai một hành vi cụ thể. Một Interface chỉ có thể khai báo các hàm và các phương thức instance, nhưng không thể triển khai hành vi cụ thể. Và tất cả phương thức là abstract ngoại trừ. Một Interface có tất cả thành viên public và không có trình triển khai.

## 128. Chúng ta gọi x/y ra như Static Modifier bằng gì? bằng gì? phương thức main?

Chúng ta trình ném lỗi NoSuchMethodError tại runtime.

## 129. Giá trị null của một tham chiếu null là gì? khai báo như là một biến instance?

Giá trị Null, trừ khi nó được khai báo tĩnh minh.

## 130. Lớp nào có thể là private hoặc protected không?

Không, một lớp nào không thể là private hoặc protected. Nó chỉ có thể là public hoặc không có modifier nào.

## 131. Tại sao chúng ta cần các lớp wrapper?

Chúng ta có thể truy cập chúng bằng các tham số phương thức khi một phương thức chỉ cần một đối tượng. Nó cung cấp các phương thức tiện ích.

## 132. Chúng khác nhau gì? giữa Error và Exception?

Một Error là một lỗi mà không thể xử lý tại runtime, ví dụ OutOfMemory error. Các Exception là các lỗi mà xử lý là do input không phù hợp, hoặc sai, ... ví dụ FileNotFoundException sẽ bắt ném như file đã cho không tồn tại.

## 133. Có cần thiết gì? giữa try/catch theo sau bởi một khối catch không?

Không cần thiết gì? giữa try/catch theo sau bởi một khối catch. Một khối try nên được theo sau bởi hoặc một khối catch hoặc một khối finally.

## 134. Khi một Thread được tạo và bắt đầu, trạng thái ban đầu (initial state) của nó là gì?

Một Thread sau khi được tạo và bắt đầu, nó trong trạng thái sẵn sàng (ready state).

## 135. Lớp Locale là gì?

Lớp Locale được sử dụng để thiết lập đầu ra output của chương trình theo các quy tắc của một khu vực địa lý, chính trị, hoặc văn hóa cụ thể.

### 136. Ph??ng th?c synchronized và l?nh synchronized là gì?

Các ph??ng th?c synchronized là các ph??ng th?c ???c s? d?ng ?? ?i?u khi?n truy c?p t?i m?t ??i t??ng. M?t l?nh synchronized có th? ch? ???c th?c thi sau khi m?t Thread ?ã thu ???c lock cho ??i t??ng ho?c l?p ???c tham chi?u trong l?nh synchronized ?ó.

### 137. G?i ph??ng th?c ??ng ho?c ?a hình t?i runtime là gì?

G?i ph??ng th?c ??ng ho?c ?a hình t?i runtime là m?t ti?n trình trong ?ó m?t l?i g?i t?i m?t ph??ng th?c b? ghi ?è ???c gi?i quy?t t?i runtime thay vì t?i compile time. Trong ti?n trình này, m?t ph??ng th?c b? ghi ?è ???c gi?i thông qua bi?n tham chi?u c?a m?t l?p cha.

### 138. Dynamic Binding (Late Biding) là gì?

Binding là nói t?i vi?c g?n k?t l?i g?i m?t t?i code ?? ???c th?c thi ?? ph?n h?i l?i l?i g?i ?ó. Dynamic Binding ngh?a là code ???c liên k?t v?i l?i g?i m?t th? t?c ?ã cho không ???c bi?t cho t?i th?i ?i?m c?a l?i g?i ?ó t?i runtime.

### 139. Constructor có th? ???c k? th?a không?

Không, Constructor không th? b? k? th?a.

### 140. L?i th? c?a ArrayList so v?i các m?ng?

ArrayList có th? t? ??ng t?ng kích c? và cung c?p m?t k? thu?t chèn và tìm ki?m m?nh m? h?n khi so sánh v?i m?ng thông th??ng.

### 141. Ho?t ??ng xóa trong LinkedList là nhanh h?n trong ArrayList, t?i sao?

Ho?t ??ng xóa trong LinkedList là nhanh h?n b?i vì nó ch? bao g?m vi?c c?p nh?t con tr? k? ti?p trong node tr?? c node b? xóa và c?p nh?t con tr? ??ng tr??c trong node sau node b? xóa.

### 142. B?n quy?t ??nh khi nào s? d?ng ArrayList và LinkedList?

N?u b?n c?n th??ng xuyên thêm và xóa các ph?n t? t? gi?a danh sách và ch? truy c?p các ph?n t? theo dãy, thì LinkedList nên ???c s? d?ng. N?u b?n c?n h? tr? truy c?p ng?u nhiên, mà không chèn ho?c xóa các ph?n t? t? b?t k? v? trí nào khác ngoài v? trí cu?i, thì nên s? d?ng ArrayList.

### 143. M?t Values Collection View là gì?

Nó là m?t Collection ???c tr? v? b?i ph??ng th?c values() c?a Map Interface. Nó bao g?m t?t c? ??i t??ng hi?n di?n nh? là các value trong Map ?ó.

### 144. Dot Operator (Toán t? .) là gì?

Dot Operator ???c s? d?ng ?? truy c?p các bi?n và ph??ng th?c instance c?a các ??i t??ng l?p. Nó c?ng ???c s? d?ng ?? truy c?p các l?p, các package con t? m?t Package.

### 145. B?n có th? s? d?ng Private Constructor ? ?âu và nh? th? nào?

Constructor dùng private ???c s? dùng n?u b?n không mu?n các l?p khác kh?i t?o ??i t??ng và ?? ng?n c?n vi?c xu?t hi?n các l?p con.

#### 146. Type Casting (Ép ki?u) là gì?

Type Casting (Ép ki?u) ngh?a là xem m?t bi?n ? m?t ki?u này nh? th? nó là ki?u khác.

#### 147. Miêu t? vòng ??i (Life Cycle) c?a Thread?

M?t Thread là m?t s? th?c thi trong m?t ch??ng trình. Vòng ??i c?a Thread bao g?m:

1. Tr?ng thái Newborn
2. Tr?ng thái Runnable
3. Tr?ng thái Running
4. Tr?ng thái Blocked
5. Tr?ng thái Dead

#### 148. ??m khác nhau gi?a hai toán t? >> và >>> ?

Toán t? >> mang sign bit khi d?ch chuy?n sang ph?i. Toán t? >>> ??i?n bit 0 khi ?ã ???c d?ch chuy?n.

#### 149. Ph??ng th?c nào c?a l?p Component ???c s? dùng ?? thi?t l?p v? trí và kích c? c?a m?t component?

S? dùng ph??ng th?c setBounds() cho m?c ích này.

#### 150. Dãy giá tr? c?a ki?u short?

Dãy giá tr? c?a ki?u short là t?  $-(2^{15})$  t?i  $2^{15} - 1$ .

#### 151. L?p cha g?n nh?t c?a Menu?

Là l?p JMenuItem.

#### 152. Java có cho phép các tham s? m?c ??nh không?

Java không cho phép các tham s? m?c ??nh (Default Argument).

#### 153. Trong Java, h? c? s? nào ???c bi?u th? v?i s? 0 b?t ??u?

H? c? s? 8, ví d?: 06

#### 154. Trong Java, h? c? s? nào ???c bi?u th? v?i ph?n b?t ??u là 0x ho?c 0X?

H? th?p l?c phân, ví d?: 0XF.

#### 155. L?nh break có th? ???c s? dùng nh? là các nhãn (label) trong Java?

Có, ví d? nh? *break one*;

#### 156. L?nh import ???c s? dùng ? ?âu trong m?t ch??ng trình Java?

Lệnh import ???c s? d?ng ? ph?n ??u ch??ng tr?nh sau l?nh package.

### 157. Gi?i thích ph??ng th?c suspend() d??i l?p Thread?

Nó ???c s? d?ng ?? t?m d?ng s? th?c thi c?a Thread.

### 158. Gi?i thích ph??ng th?c isAlive() d??i l?p Thread?

Nó ???c s? d?ng ?? tìm xem có hay không m?t Thread là v?n ?ang ch?y.

### 159. B?n hi?u gì v? ph??ng th?c currentThread()?

?ây là m?t ph??ng th?c public và static ?? thu nh?n m?t tham chi?u t?i Thread hi?n t?i.

### 160. Gi?i thích main thread d??i s? th?c thi c?a l?p Thread?

Main thread ???c t?o t? ??ng và nó b?t ??u th?c thi ngay khi m?t ch??ng tr?nh b?t ??u. Nó là m?t thread mà t? ?ó hình thành t?t c? thread con khác.

### 161. Vòng ??i c?a m?t Applet?

Vòng ??i c?a m?t Applet g?m:

1. Kh?i t?o
2. Begin
3. Stop
4. Cancel
5. Painting

### 162. Vai trò c?a ph??ng th?c init() d??i Applet?

Nó kh?i t?o Applet và là ph??ng th?c ??u tiên ???c g?i.

### 163. Ph??ng th?c nào ???c g?i b?i l?p Applet ?? t?i m?t hình ?nh?

S? d?ng ph??ng th?c getImage(URL c?a ??i t??ng, tên file).

### 164. ??nh ngh?a ph?n code ???c s? d?ng nh? là m?t thu?c tính c?a Applet?

Nó ???c s? d?ng ?? xác ??nh tên c?a l?p Applet.

### 165. Canvas là gì?

Nó là một bộ máy tính mà chúng ta sử dụng và các hình ảnh hoặc các hình ảnh khác.

### 166. Những nhà Lập trình mạng?

Nó nói về việc viết các chương trình mà thực thi qua nhiều thiết bị (máy tính), trong đó các thiết bị khác kết nối với nhau bằng sử dụng mạng.

### 167. Socket là gì?

Socket cung cấp kết nối giao tiếp giữa hai máy tính bằng sử dụng TCP. Một chương trình Client tạo một Socket trên máy giao tiếp của nó và kết nối với Socket ở một Server.

### 168. Lợi ích của Java Socket?

Lập trình dựa trên Socket giúp có thể dễ dàng triển khai cho truy cập thông chung. Các Socket là linh hoạt và nó làm cho việc truy cập qua mạng ít hơn.

### 169. Hạn chế của Java Socket?

Truy cập thông dựa trên Socket chỉ cho phép gửi các gói dữ liệu thô giữa các ứng dụng. Có Client-Side và Server-Side phải cung cấp các kết nối để làm cho dữ liệu có thể dùng được trong bất cứ cách thức nào.

### 170. Lập trình nào sử dụng bằng các ứng dụng Server để thu nhận một cổng (port) và các yêu cầu từ Client?

Lập trình java.net.ServerSocket sử dụng cho mục đích này.

### 171. Lập trình nào biểu diễn Socket mà cả Server và Client sử dụng để giao tiếp với nhau?

Đó là lập trình java.net.Socket.

### 172. Tại sao Generic sử dụng trong Java?

Generic cung cấp tính an toàn kiểu tại compile-time mà cho phép lập trình viên viết các kiểu không hợp lệ tại thời gian biên dịch. Các phương thức Generic và các lớp Generic trong Java cho phép lập trình viên xác định, viết một khai báo phương thức, một tập hợp các phương thức liên quan, hoặc viết một khai báo lớp định nghĩa là một tập hợp các kiểu liên quan.

### 173. Bạn cần thiết lập các biến môi trường nào trên thiết bị để có thể chạy các chương trình Java?

Hai biến môi trường CLASSPATH và PATH.

### 174. Có cần thiết phải nhập java.lang package không?

Is not. Nó được tải theo mặc định bởi JVM.

### 175. Lập trình cao nhất sử dụng là gì?

Nếu một lớp được khai báo bên trong một lớp và xác định Static Modifier, biến biên dịch xem lớp đó giống như một lớp cao nhất nào khác. Lập trình cao nhất bằng lập trình Inner.

### 176. Trình bày Externalizable Interface?

Externalizable là một Interface chứa hai phương thức readExternal và writeExternal. Hai phương thức này cung cấp cho bạn một cơ chế khi cần thông qua kỹ thuật Serialization.

### 177. Khi nào finally sẽ và khi nào thì không có System.exit(0); và khi nào try?

Trong trường hợp này, khi nào finally sẽ không thực thi, bởi vì khi bạn cung cấp System.exit(0); thì cơ chế thoát khi nào cũng thực thi, và vì thế khi nào finally này sẽ không bao giờ thực thi.

### 178. Bạn hiểu gì về Daemon Thread?

Daemon Thread là một Thread có quyền ưu tiên thấp, chúng không liên tục trong Background thực hiện hoạt động của trình diễn rác cho Java Runtime System.

### 179. Phương thức nào để tạo Daemon Thread?

Phương thức setDaemon để tạo một Daemon Thread.

### 180. Tên của Thread khi nào khai phương thức nào?

Tên của tác vụ khi nào khai phương thức run().

### 181. Lớp GregorianCalendar là gì?

Lớp GregorianCalendar cung cấp sự hỗ trợ Western Calendar truyền thống.

### 182. Bạn hiểu gì về lớp SimpleTimeZone?

Lớp SimpleTimeZone cung cấp sự hỗ trợ cho một Gregorian Calendar.

### 183. Sự khác nhau giữa hai tham số size và capacity của một Vector?

Tham số size là số phần tử thực sự có trong Vector, trong khi capacity là số phần tử tối đa nó có thể lưu trữ.

### 184. Một Vector có thể chứa các đối tượng không?

Có, một Vector có thể chứa các đối tượng. Bởi vì một Vector lưu trữ mọi thứ theo khái niệm đối tượng.

### 185. Trình bày Enumeration?

Một Enumeration là một Interface chứa các phương thức để truy cập cấu trúc dữ liệu liệt kê mà không có Enumeration

You finished reading the article "**200 common Java interview questions, with reference answers**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

