100 Most Asked Java Interview QnA

Made By:

Yadneyesh (Curious Coder)
CodWithCurious.com

Java

1. What is Java?

Java is a high-level, object-oriented programming language that is widely used for developing a variety of applications, including web, desktop, and mobile applications.

2. What is the difference between Java and JavaScript?

Java and JavaScript are two different programming languages with different purposes. Java is used for building applications, while JavaScript is primarily used for adding interactivity to web pages.

3. What is the main principle of Java programming?

Java follows the principle of "write once, run anywhere" (WORA), which means that Java code can be compiled into bytecode and executed on any platform that has a Java Virtual Machine (JVM).

4. What are the main features of Java?

Some of the main features of Java include platform independence, object-oriented programming, automatic memory management (garbage collection), and strong type checking.

5. What is a class in Java?

In Java, a class is a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that objects of that class can have.

6. What is an object in Java?

An object in Java is an instance of a class. It represents a specific entity or item that can have its own set of attributes and behaviors defined by its class.

7. What is a method in Java?

A method in Java is a block of code that performs a specific task. It can be called or invoked to execute its defined functionality.

8. What is the difference between a class and an object?

A class is a blueprint or template, while an object is an instance of that class. A class defines the structure and behavior of objects, while objects represent specific instances of the class.

9. What is inheritance in Java?

Inheritance is a mechanism in Java where a class can inherit properties and behaviors from another class. It allows for code reuse and creating a hierarchical relationship between classes.

10. What are the types of inheritance in Java?

Java supports single inheritance, where a class can inherit from only one superclass, and multiple inheritance through interfaces, where a class can implement multiple interfaces.

11. What is polymorphism in Java?

Polymorphism is the ability of an object to take on many forms. In Java, it allows objects of different classes to be treated as objects of a common superclass, enabling code flexibility and reusability.

12. What are the access modifiers in Java?

Java provides four access modifiers: public, private, protected, and default (no modifier). They control the visibility and accessibility of classes, methods, and variables.

13. What is encapsulation in Java?

Encapsulation is the process of hiding internal details and providing a public interface to interact with an object. It helps in achieving data abstraction and protects data from

unauthorized access.

14. What is a constructor in Java?

A constructor in Java is a special method that is used to initialize objects of a class. It is called automatically when an object is created and has the same name as the class.

15. What is the difference between a constructor and a method?

A constructor is a special method used for object initialization and is called automatically when an object is created. A method, on the other hand, is a block of code that performs a specific task and needs to be called explicitly.

16. What is the Java Virtual Machine (JVM)?

The JVM is a crucial part of the Java platform. It is responsible for executing Java bytecode and provides a runtime environment in which Java programs can run on any hardware or operating system.

17. What is the Java Development Kit (JDK)?

The JDK is a software development kit provided by Oracle, which includes the necessary tools and libraries to develop, compile, and run Java programs. It consists of the JVM, compiler, and other utilities.

18. What is the difference between the JDK and the JRE?

The JDK (Java Development Kit) is a software development kit that includes the tools needed to develop Java applications, while the JRE (Java Runtime Environment) is a runtime environment required to run Java applications.

19. What is a package in Java?

A package in Java is a way of organizing related classes and interfaces. It provides a namespace and helps in avoiding naming conflicts.

20. What is the difference between an abstract class and an interface?

An abstract class can have both abstract and non-abstract methods and can be extended by other classes, while an interface only contains abstract method declarations and can be implemented by classes.

21. What is a static method in Java?

A static method in Java is a method that belongs to the class rather than an instance of the class. It can be called without creating an object of the class.

22. What is the keyword "final" used for in Java?

The "final" keyword in Java can be used to declare a variable, a method, or a class. A final variable cannot be changed, a final method cannot be overridden, and a final class cannot be inherited.

23. What is method overloading in Java?

Method overloading is the ability to define multiple methods with the same name but different parameters in the same class. The appropriate method is called based on the arguments passed.

24. What is method overriding in Java?

Method overriding is the ability to provide a different implementation of a method in a subclass that is already defined in its superclass. It allows for the execution of the overridden method instead of the superclass method.

25. What is the difference between method overloading and method overriding?

Method overloading involves defining multiple methods with the same name but different parameters in the same class, while method overriding involves providing a different implementation of a method in a subclass that is already defined in its superclass.

26. What is the "this" keyword in Java?

The "this" keyword in Java refers to the current instance of a class. It can be used to access instance variables, call instance methods, or invoke constructors.

27. What is a static variable in Java?

A static variable in Java is a variable that belongs to the class rather than an instance of the class. It is shared among all instances of the class.

28. What is the purpose of the "final" keyword in method parameters?

The "final" keyword in method parameters is used to make the parameter value unchangeable within the method. It ensures that the parameter cannot be reassigned or modified.

29. What is the purpose of the "static" keyword in Java?

The "static" keyword in Java is used to declare variables, methods, and nested classes that belong to the class itself, rather than instances of the class. It allows accessing them without creating an object of the class.

30. What is the difference between "==" and ".equals()" in Java?

The "==" operator in Java is used to compare the equality of object references, while the ".equals()" method is used to compare the equality of object values. The ".equals()" method can be overridden to provide custom equality comparison.

31. What is the purpose of the "super" keyword in Java?

The "super" keyword in Java is used to refer to the superclass of a class. It can be used to access superclass members, invoke superclass constructors, or differentiate between superclass and subclass members with the same name.

32. What is a thread in Java?

A thread in Java is a lightweight unit of execution within a program. It allows concurrent execution of multiple tasks or activities, enabling better utilization of system resources.

33. How do you create and start a thread in Java?

To create and start a thread in Java, you can either extend the "Thread" class and override the "run()" method, or implement the "Runnable" interface and pass it to a new "Thread" object. Then call the "start()" method on the thread object to begin execution.

34. What is synchronization in Java?

Synchronization in Java is a technique used to control the access and execution of multiple threads to ensure that only one thread can access a shared resource or code block at a time.

35. What is the difference between the "synchronized" block and the "synchronized" method?

A "synchronized" block in Java allows a specific block of code to be synchronized, ensuring that only one thread can execute it at a time. A "synchronized" method applies synchronization to the entire method, making it mutually exclusive for all threads.

36. What is the purpose of the "volatile" keyword in Java?

The "volatile" keyword in Java is used to indicate that a variable's value may be modified by multiple threads. It ensures that any read or write operation on the variable is directly performed on the main memory, rather than relying on CPU caches.

37. What is an exception in Java?

An exception in Java is an event that occurs during the execution of a program, which disrupts the normal flow of instructions. It represents an error condition or an exceptional circumstance.

38. What is the difference between checked and unchecked exceptions?

Checked exceptions are checked at compile-time, and the programmer is required to handle or declare them using the "throws" keyword. Unchecked exceptions, on the other hand, are not checked at compile-time, and the programmer is not obligated to handle or declare them.

39. How do you handle exceptions in Java?

Exceptions in Java can be handled using try-catch blocks. The code that may throw an exception is placed inside the try block, and if an exception occurs, it is caught and handled in the catch block.

40. What is the purpose of the "finally" block in exception handling?

The "finally" block in Java is used to define a block of code that will be executed regardless of whether an exception occurs or not. It is often used to release resources or perform cleanup operations.

41. What is the difference between the "throw" and "throws" keywords in Java?

The "throw" keyword in Java is used to manually throw an exception, while the "throws" keyword is used in method declarations to specify that the method may throw certain types of exceptions.

42. What is the difference between checked exceptions and runtime exceptions?

Checked exceptions are checked at compile-time and must be handled or declared, while runtime exceptions (unchecked exceptions) are not required to be handled or declared.

43. What is the Java API?

The Java API (Application Programming Interface) is a collection of classes, interfaces, and other resources provided by the Java Development Kit (JDK). It provides a set of predefined classes and methods for building Java applications.

44. What is the difference between an ArrayList and a LinkedList?

An ArrayList is implemented as a resizable array, allowing fast random access but slower insertion and removal of elements. A LinkedList is implemented as a doubly-linked list, allowing fast insertion and removal but slower random access.

45. What is the difference between a HashSet and a TreeSet?

A HashSet in Java stores elements in no particular order, using a hash table for fast access but does not maintain any specific order. A TreeSet stores elements in sorted order and allows for efficient retrieval of elements based on their natural ordering or a custom comparator.

46. What is the difference between the "equals()" method and the "hashCode()" method?

The "equals()" method is used to compare the equality of objects based on their values, while the "hashCode()" method is used to calculate a unique hash code value for an object, typically used for efficient retrieval in hash-based data structures like HashMaps.

47. What is the difference between a shallow copy and a deep copy?

A shallow copy creates a new object that shares the same references as the original object, while a deep copy creates a new object and recursively copies all the referenced objects as well, resulting in separate copies.

48. What is a lambda expression in Java?

A lambda expression in Java is an anonymous function that can be used to simplify the syntax of functional interfaces. It allows for more concise and readable code, especially when working with functional programming constructs.

49. What is functional programming in Java?

Functional programming in Java is a programming paradigm that emphasizes writing programs using pure functions and immutable data. It involves treating functions as first-class citizens and utilizing higher-order functions and lambda expressions.

50. What are the Java 8 features for functional programming?

Java 8 introduced several features to support functional programming, including lambda expressions, functional interfaces, the Stream API for working with collections, and default methods in interfaces.

51. What is the difference between an interface and an abstract class?

An interface in Java can only declare method signatures and constants but cannot provide implementations, while an abstract class can have both method declarations and concrete implementations. A class can implement multiple interfaces but can inherit from only one abstract class.

52. What is the purpose of the "default" keyword in interface methods?

The "default" keyword in Java interfaces is used to define a default implementation for a method. It allows adding new methods to existing interfaces without breaking the implementations of classes that implement those interfaces.

53. What is the difference between a BufferedReader and a Scanner?

A BufferedReader in Java reads text from a character stream with efficient buffering, while a Scanner can parse different types of data from various sources such as files, strings, or standard input.

54. What is the purpose of the "StringBuilder" class in Java?

The "StringBuilder" class in Java is used to create and manipulate mutable sequences of characters. It is more efficient than concatenating strings using the "+" operator, as it avoids unnecessary object creations.

55. What is the difference between the "Comparable" and "Comparator" interfaces?

The "Comparable" interface is used to define a natural ordering for a class by implementing the "compareTo()" method. The "Comparator" interface, on the other hand, provides a way to define custom ordering by implementing the "compare()" method and is independent of the class being compared.

56. What is the purpose of the "assert" keyword in Java?

The "assert" keyword in Java is used to perform assertions, which are checks placed in the code to verify specific conditions. It is primarily used during development and testing to catch potential bugs or invalid assumptions.

57. What is the difference between a local variable and an instance variable?

A local variable in Java is declared inside a method or a block and has a limited scope within that method or block. An instance variable, also known as a member variable, is declared within a class but outside any method and is accessible to all methods of the class.

58. What is the purpose of the "transient" keyword in Java?

The "transient" keyword in Java is used to indicate that a variable should not be serialized during object serialization. When an object is deserialized, transient variables are set to their default values.

59. What is the purpose of the "static" block in Java?

The "static" block in Java is used to initialize static variables or perform one-time initialization tasks for a class. It is executed when the class is loaded into memory, before any objects of that class are created.

60. What is the purpose of the "strictfp" keyword in Java?

The "strictfp" keyword in Java is used to ensure strict adherence to the IEEE 754 standard for floating-point calculations. It ensures consistent results across different platforms by disabling some optimizations that can affect precision.

61. What is the difference between a public class and a default (package-private) class?

A public class in Java can be accessed from any other class, regardless of the package they belong to. A default class, also known as a package-private class, is only accessible within the same package and cannot be accessed from outside the package.

62. What is the purpose of the "enum" keyword in Java?

The "enum" keyword in Java is used to define an enumeration, which is a special type that represents a fixed set of constants. It allows for more structured and type-safe representation of predefined values.

63. What is the purpose of the "break" and "continue" statements in Java?

The "break" statement in Java is used to terminate the execution of a loop or switch statement and resume execution after the loop or switch block. The "continue" statement is used to skip the current iteration of a loop and move to the next iteration.

64. What is the purpose of the "try-with-resources" statement in Java?

The "try-with-resources" statement in Java is used to automatically close resources that implement the "AutoCloseable" interface. It ensures that resources, such as file streams or database connections, are properly closed, even if an exception occurs.

65. What is the purpose of the "instanceof" operator in Java?

The "instanceof" operator in Java is used to check whether an object is an instance of a specific class or implements a specific interface. It returns a boolean value indicating the result of the check.

66. What is the difference between the pre-increment and post-increment operators?

The pre-increment operator (++i) in Java increments the value of a variable and returns the incremented value, while the post-increment operator (i++) increments the value of a variable but returns the original value before the increment.

67. What is the difference between the pre-decrement and post-decrement operators?

The pre-decrement operator (--i) in Java decrements the value of a variable and returns the decremented value, while the post-decrement operator (i--) decrements the value of a variable but returns the original value before the decrement.

68. What is the purpose of the "Math" class in Java?

The "Math" class in Java provides various methods for performing common mathematical operations, such as square roots, trigonometric functions, exponential calculations, rounding, and more.

69. What is the purpose of the "StringBuffer" class in Java?

The "StringBuffer" class in Java is used to create and manipulate mutable sequences of characters, similar to the "StringBuilder" class. However, "StringBuffer" is synchronized and thread-safe, making it suitable for multi-threaded environments.

70. What is the purpose of the "Math.random()" method in Java?

The "Math.random()" method in Java returns a random double value between 0.0 (inclusive) and 1.0 (exclusive). It is often used to generate random numbers or simulate random behavior.

71. What is the purpose of the "Character" class in Java?

The "Character" class in Java provides methods for working with individual characters, such as checking for character types (letters, digits, whitespace), converting case, and performing character-based operations.

72. What is the purpose of the "Integer" class in Java?

The "Integer" class in Java is a wrapper class that provides methods for working with integer values, such as converting strings to integers, performing arithmetic operations, and converting integers to different representations (binary, hexadecimal).

73. What is the purpose of the "Double" class in Java?

The "Double" class in Java is a wrapper class that provides methods for working with double-precision floating-point values. It offers functionality for parsing strings, performing arithmetic operations, and converting doubles to different representations (binary, hexadecimal).

74. What is the purpose of the "System" class in Java?

The "System" class in Java provides access to system resources and allows interaction with the system environment. It contains methods for standard input/output, error output, current time, copying arrays, and more.

75. What is the purpose of the "File" class in Java?

The "File" class in Java is used to represent and manipulate file and directory paths. It provides methods for creating, deleting, renaming, and querying file properties such as size, last modified date, and permissions.

76. What is the purpose of the "FileNotFoundException" in Java?

The "FileNotFoundException" in Java is an exception that is thrown when an attempt to access a file that does not exist or cannot be found is made. It is typically caught and handled to handle file-related errors.

77. What is the purpose of the "NullPointerException" in Java?

The "NullPointerException" in Java is an exception that is thrown when a null reference is accessed and used where an object reference is expected. It indicates a programming error and should be handled or prevented to avoid unexpected crashes.

78. What is the purpose of the "ArrayIndexOutOfBoundsException" in Java?

The "ArrayIndexOutOfBoundsException" in Java is an exception that is thrown when an invalid index is used to access an array. It indicates that the index is either negative or exceeds the array's bounds.

79. What is the purpose of the "ArithmeticException" in Java?

The "ArithmeticException" in Java is an exception that is thrown when an arithmetic operation produces an illegal or undefined result. It typically occurs when dividing by zero or performing unsupported mathematical operations.

80. What is the purpose of the "NumberFormatException" in Java?

The "NumberFormatException" in Java is an exception that is thrown when a string cannot be parsed into a numeric value of the expected format. It occurs when attempting to convert a string to an integer, float, or double, but the string does not represent a valid number.

81. What is the purpose of the "StringBuilder" class in Java?

The "StringBuilder" class in Java is used to create and manipulate mutable sequences of characters. It provides methods for appending, inserting, deleting, and modifying

character sequences efficiently.

82. What is the purpose of the "HashSet" class in Java?

The "HashSet" class in Java is an implementation of the Set interface that stores unique elements in no particular order. It provides constant-time performance for basic operations like adding, removing, and checking for the presence of elements.

83. What is the purpose of the "HashMap" class in Java?

The "HashMap" class in Java is an implementation of the Map interface that stores key-value pairs. It provides fast retrieval and insertion of elements based on their keys and allows for efficient mapping and lookup operations.

84. What is the purpose of the "LinkedList" class in Java?

The "LinkedList" class in Java is an implementation of the List interface that uses a doubly-linked list to store elements. It provides efficient insertion and removal of elements at both ends of the list but slower random access.

85. What is the purpose of the "Comparator" interface in Java?

The "Comparator" interface in Java is used to define custom ordering of objects. It provides a way to compare objects based on specific criteria other than their natural ordering defined by the "Comparable" interface.

86. What is the purpose of the "Comparable" interface in Java?

The "Comparable" interface in Java is used to define the natural ordering of objects of a class. It provides a method, "compareTo()", that allows objects to be compared and sorted based on their natural order.

87. What is the purpose of the "super" keyword in Java?

The "super" keyword in Java is used to refer to the superclass of a class or to call the superclass's constructor, methods, or variables. It is primarily used to differentiate between superclass and subclass members with the same name.

88. What is the purpose of the "this" keyword in Java?

The "this" keyword in Java is used to refer to the current instance of a class. It is primarily used to differentiate between instance variables and parameters or to invoke other constructors within a class.

89. What is the purpose of the "final" keyword in Java?

The "final" keyword in Java is used to define constants, make variables unchangeable, or prevent method overriding or class inheritance. It ensures that the value of a variable or the implementation of a method or class cannot be modified.

90. What is the purpose of the "static" keyword in Java?

The "static" keyword in Java is used to define class-level variables and methods that are shared among all instances of a class. It allows accessing variables or methods without creating an instance of the class.

91. What is the purpose of the "abstract" keyword in Java?

The "abstract" keyword in Java is used to define abstract classes or methods. An abstract class cannot be instantiated and serves as a base class for subclasses. An abstract method does not have an implementation and must be overridden in a subclass.

92. What is the purpose of the "interface" keyword in Java?

The "interface" keyword in Java is used to define interfaces, which declare methods that implementing classes must provide. It allows for multiple inheritance by implementing multiple interfaces and enables the concept of polymorphism.

93. What is the purpose of the "package" keyword in Java?

The "package" keyword in Java is used to define a package, which is a way to organize related classes and interfaces. It provides a hierarchical structure and helps prevent naming conflicts between classes.

94. What is the purpose of the "import" keyword in Java?

The "import" keyword in Java is used to import classes, interfaces, or packages into a source file. It allows using classes from other packages without specifying their fully qualified names.

95. What is the purpose of the "throw" keyword in Java?

The "throw" keyword in Java is used to manually throw an exception. It is typically used when a program encounters an error or exceptional situation that cannot be handled, and the control should be transferred to an exception handler.

96. What is the purpose of the "throws" keyword in Java?

The "throws" keyword in Java is used in method declarations to specify that a method may throw certain types of exceptions. It allows the caller of the method to handle the exception or propagate it further.

97. What is the purpose of the "try-catch-finally" block in Java?

The "try-catch-finally" block in Java is used to handle exceptions. The "try" block contains the code that may throw an exception, the "catch" block catches and handles the exception, and the "finally" block contains cleanup code that is executed regardless of whether an exception occurs or not.

98. What is the purpose of the "instanceof" operator in Java?

The "instanceof" operator in Java is used to check the type of an object at runtime. It returns a boolean value indicating whether an object is an instance of a particular class or implements a specific interface.

99. What is the purpose of the "break" statement in Java?

The "break" statement in Java is used to terminate the execution of a loop or switch statement. It allows exiting a loop prematurely or skipping the remaining cases in a switch statement.

100. What is the purpose of the "continue" statement in Java?

The "continue" statement in Java is used to skip the current iteration of a loop and continue with the next iteration. It allows skipping certain iterations based on specific conditions without exiting the loop entirely.