

57 Core Java interview questions to ask your applicants

Questions

1. What is the difference between JDK, JRE, and JVM?
2. Can you explain the concept of object-oriented programming and how it is implemented in Java?
3. What are the main principles of encapsulation, inheritance, and polymorphism in Java?
4. How does Java achieve platform independence?
5. What are the differences between '==' operator and 'equals()' method in Java?
6. Can you explain Exception Handling in Java and the difference between checked and unchecked exceptions?
7. What is the purpose of the 'final' keyword in Java?
8. How do you create a thread in Java? Can you describe the differences between using the Runnable interface and extending the Thread class?
9. What are Java Collections, and can you name some commonly used collection classes?
10. Can you explain the concept of garbage collection in Java and how it works?
11. Can you explain the concept of method overloading in Java?
12. What is the difference between an abstract class and an interface in Java?
13. How does Java handle memory management, and what is the role of the garbage collector?
14. Explain the concept of multithreading in Java and its benefits.
15. What are Java Annotations and how are they used?
16. Can you explain the role of the Java ClassLoader and its types?
17. How do you manage memory leaks in Java applications?
18. Describe the process and benefits of using Java Reflection.
19. What is the Java Memory Model, and why is it important in multithreading?
20. How would you implement a Singleton design pattern in Java?
21. Can you explain the difference between ConcurrentHashMap and HashMap?
22. What are the principles and benefits of the Stream API in Java 8?
23. Can you discuss the internal workings of the Java HashMap?
24. Explain the concept of lambda expressions and their uses in Java.
25. How does the Fork/Join framework in Java work?
26. What are the key differences between Java 8 and previous versions?
27. How would you optimize the performance of a Java application?
28. What is the role of the ExecutorService framework in Java?
29. Can you explain the difference between Future and CompletableFuture in Java?
30. Describe how you would handle security concerns in a Java application.
31. What is the difference between a class and an object in Java?
32. Can you explain the concept of inheritance in Java?
33. What is polymorphism in Java and how is it useful?
34. What do you understand by encapsulation in Java?
35. What is abstraction in Java and why is it important?
36. How do constructors work in Java?
37. What is the role of 'this' keyword in Java?
38. What is method overriding in Java and how does it differ from overloading?
39. Can you explain the significance of constructors in class inheritance?
40. What is the concept of composition in Java, and how does it differ from inheritance?
41. How does Java implement polymorphism at runtime?
42. What are the benefits of using interfaces over abstract classes?
43. Can you explain the concept of the 'super' keyword in Java?
44. What is the role of an interface in achieving multiple inheritance in Java?
45. How can you achieve abstraction using interfaces in Java?
46. Can you discuss the differences between public, protected, and private access modifiers in Java?
47. What is the importance of encapsulation and how can it be achieved in Java?
48. How would you approach troubleshooting a performance issue in a Java application?
49. Can you describe a situation where you had to refactor a piece of code? What was the outcome?
50. Imagine you are tasked with integrating a third-party library into an existing Java application. How would you ensure compatibility?
51. If a bug is reported in production, what steps would you take to diagnose and resolve it?
52. How do you handle version control and collaboration when working on a Java project with a team?
53. Describe a challenging problem you faced in a Java project and how you resolved it.
54. How would you implement logging in a Java application to ensure you can track issues effectively?
55. If you had to design a RESTful web service in Java, what considerations would you take into account for security?
56. How would you manage dependencies in a Java application to ensure smooth builds and deployments?
57. Can you explain how you would approach writing unit tests for a Java method?