
JDK, JVM, JRE Architecture

1. What is the difference between JDK, JRE, and JVM?
2. What are the main components of JVM?
3. What is the role of the ClassLoader in JVM?
4. What is the difference between JIT compiler and interpreter in JVM?
5. What is bytecode and why is Java platform-independent?
6. Explain the memory areas inside JVM (Heap, Stack, Method Area, etc.)
7. What is Garbage Collection and how does it work in JVM?
8. What is the difference between Stack memory and Heap memory?

Principles of Java

9. What are the four main principles of OOP in Java?
10. What does "Write Once, Run Anywhere" mean in Java?
11. What is the difference between procedural and object-oriented programming?

Access Specifiers

12. What are the four access specifiers in Java?
13. What is the difference between private and protected?
14. What is the default access specifier and where is it accessible?
15. Can a top-level class be declared private in Java? Why or why not?

Class, Member Variables, Member Functions, Object

16. What is the difference between a class and an object?
17. What is the difference between instance variables and local variables?
18. How is an object created in memory in Java?

19. What is the difference between a member function and a static function?
20. What happens when you create an object using the new keyword?

Static and Non-Static Blocks

21. What is a static block and when is it executed?
22. What is an instance initializer block and when is it executed?
23. What is the order of execution of static block, instance block, and constructor?
24. Can a static block throw a checked exception?
25. Can you have multiple static blocks in a class?

---

main() Method

26. Why is the main method public in Java?
27. Why is the main method static in Java?
28. Why is the return type of main void?
29. Why does main accept only String[] as arguments?
30. Can you overload the main method in Java?
31. Can you run a Java program without a main method?

System.out.println()

32. What is System in System.out.println()?
33. What is out in System.out.println()?
34. What is println and how is it different from print?
35. What class does out belong to in Java?
36. Can you replace System.out with a custom PrintStream?

Constructors

37. What is a constructor and how is it different from a method?
38. What is constructor overloading?
39. What is constructor chaining and how is it achieved?
40. What is the default constructor and when does the compiler provide it?
41. Can a constructor be private? What is the use case?
42. Can a constructor be final, static, or abstract?

---

this and this() Calling Statement

43. What is the use of the this keyword in Java?
44. What is the difference between this and this()?
45. What are the rules for using this() in a constructor?
46. Can this be used inside a static method?
47. How does this help in resolving variable shadowing?

Type Casting and Type Conversion

48. What is the difference between implicit and explicit type conversion?
49. What is widening and narrowing conversion in primitives?
50. Can you cast a double to an int? What happens to the value?
51. What is the difference between type casting and type conversion?
52. What happens when you add an int and a double in Java?

Object Class

53. What methods does the Object class provide in Java?
54. What is the contract between equals() and hashCode()?
55. What does the toString() method do by default?
56. What is the clone() method and what interface must be implemented to use it?

57. What is the purpose of the finalize() method?

Upcasting and Downcasting

58. What is upcasting in Java and is it implicit or explicit?
59. What is downcasting and when can it throw an exception?
60. What is a ClassCastException and when does it occur?
61. How does instanceof help before downcasting?

Method Overloading / Early Binding

62. What is method overloading and what are its rules?
63. What is early binding (static binding)?
64. Can you overload methods by changing only the return type?
65. How does the compiler resolve overloaded methods?

Inheritance

66. What is inheritance and what are its types in Java?
67. Why does Java not support multiple inheritance with classes?
68. What is the difference between extends and implements?
69. What is the IS-A relationship in Java?
70. Can a constructor be inherited in Java?

Method Overriding / Late Binding

71. What are the rules for method overriding in Java?
72. What is late binding (dynamic dispatch)?
73. Can you override a static method in Java?
74. Can you override a private or final method?

75. What is the difference between overloading and overriding?

---

super and super() Calling Statement

76. What is the use of the super keyword in Java?
77. What is super() and when is it called implicitly?
78. Can you use both this() and super() in the same constructor?
79. How do you call an overridden method of the parent class using super?

Polymorphism

80. What is runtime polymorphism and how is it achieved?
81. What is the difference between compile-time and runtime polymorphism?
82. Can polymorphism be achieved with fields (variables)?

Abstract Class and Interface

83. What is an abstract class and when should you use it?
84. Can an abstract class have a constructor?
85. What is the difference between an abstract class and an interface?
86. Can an interface have concrete methods? (default and static methods)
87. What is a functional interface?
88. Can a class implement multiple interfaces?

Wrapper Classes

89. What are wrapper classes and why are they needed?
90. What is autoboxing and unboxing?
91. What is the difference between Integer.parseInt() and Integer.valueOf()?
92. Why does Integer.valueOf(127) == Integer.valueOf(127) return true but not for 128?

Generics

93. What are generics in Java and why are they used?
94. What is the difference between List<?> and List<Object>?
95. What is type erasure in generics?

Exception Handling

96. What is the difference between checked and unchecked exceptions?
97. What is the difference between throw and throws?
98. What is the purpose of the finally block?
99. Can you have a try block without a catch block?

Encapsulation

101. What is encapsulation and why is it important?
102. How do you achieve encapsulation in Java?
103. What is the difference between encapsulation and data hiding?
104. Why are getter and setter methods used instead of making fields public?

instanceof Keyword

105. What is the instanceof keyword and what does it return?
106. Can instanceof return false for a subclass object?
107. What happens when you use instanceof with a null reference?
108. How does instanceof relate to downcasting safety?

Abstraction

109. What is abstraction and how is it different from encapsulation?
110. How is abstraction achieved in Java — abstract class vs interface?

111. Can you instantiate an abstract class directly?

112. What is the real-world benefit of abstraction in software design?

Exception Handling (Deeper)

113 What is the difference between Error and Exception in Java?
114 What is a custom exception and how do you create one?

115 What is exception propagation in Java?

116 What is a multi-catch block (Java 7+)?

117 What is the try-with-resources statement and what interface must the resource implement?

118 Can finally block override a return statement in try?

Packages

119. What is a package in Java and why is it used?
120. What is the difference between import and static import?

121. How do you create a user-defined package in Java?

122. What is the default package in Java?

123. What is the difference between packages and access specifiers?

Final Keyword

124. What is the difference between final, finally, and finalize?

125. What happens when you declare a variable as final?

126. Can you override a final method?

127. What is a final class and can it be inherited? Give an example.

128. Can a final variable be declared without initialization?
