Day-19

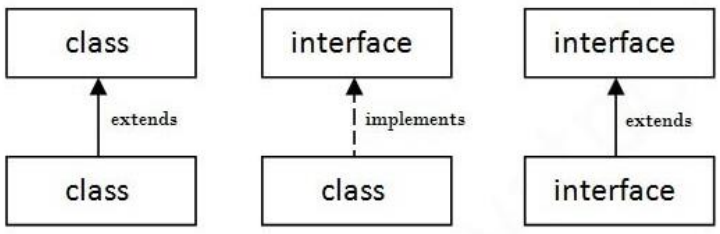
Interface

Interface in Java

1. An interface is a **blueprint** of a class.
2. It contains **final** and **static** variables.
3. It contains **abstract methods** (methods with only the signature). From Java 8 onwards, it can also have **default** and **static methods**.
4. An interface having exactly one abstract method, then it is called ad **Functional Interface**.
5. An abstract method only has a signature (no implementation).
6. All methods in an interface are implicitly **public.** From Java 9 onwards, **private** methods allowed.
7. Interfaces allow **multiple inheritance** (a class can implement multiple interfaces).
8. An interface is defined using the interface keyword.
9. A **class** extends another class, an **interface** extends another interface, and a **class implements** an interface.
10. You can create an **object reference** of an interface, but you cannot instantiate an interface directly.

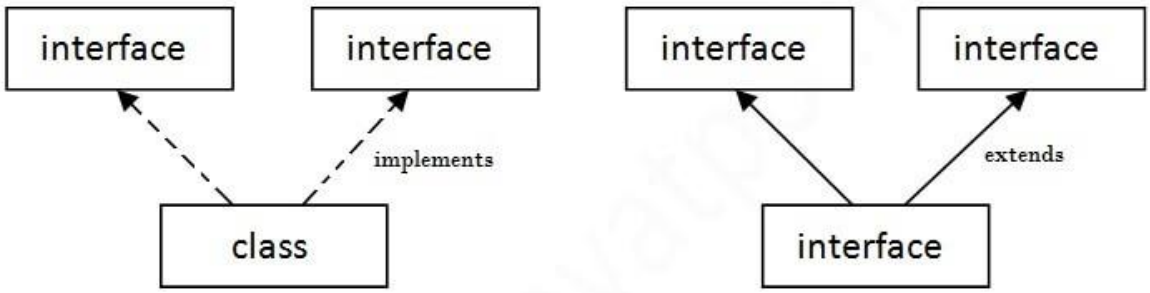
The relationship between classes and interfaces

- A class **extends** another class.
- An interface **extends** another interface.
- But a class **implements** an interface.



Multiple inheritance in Java by interface

If a class implements multiple interfaces or an interface extends multiple interfaces, it is known as multiple inheritance.



Multiple Inheritance in Java

Difference between abstract class and interface

- **Abstract class and interface both** are used to achieve **abstraction** where we can declare the abstract methods.
- **Abstract class** and **interface both can't be instantiated.**
- But there are many differences between abstract class and interface that are given below.

Abstract class	Interface
1) Abstract class can have abstract and non-abstract methods.	Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
2) Abstract class doesn't support multiple inheritance.	Interface supports multiple inheritance.
3) Abstract class can have final, non-final, static and non-static variables.	Interface has only static and final variables.
4) Abstract class can provide the implementation of interface.	Interface can't provide the implementation of abstract class.
5) The abstract keyword is used to declare abstract class.	The interface keyword is used to declare interface.
6) An abstract class can extend another Java class and implement multiple Java interfaces.	An interface can extend another Java interface only.
7) An abstract class can be extended using keyword "extends".	An interface can be implemented using keyword "implements".

8) A Java abstract class can have class members like private, protected, etc.	Members of a Java interface are public by default.

9)Example:
public abstract class Shape{
public abstract void draw();
}
	Example:
	public interface Drawable{
	void draw();
	}

Quiz: Abstract Class and Interface in Java

1. What is abstraction in Java?

A. Hiding the implementation details and showing only functionality
B. Showing all the details of a class
C. Encapsulation of data
D. Overriding methods in a subclass

Answer: A

2. Which of the following is a correct statement about abstract classes?

A. Abstract classes cannot have constructors
B. Abstract classes can be instantiated
C. Abstract classes can have abstract and non-abstract methods
D. Abstract classes must have all methods as abstract

Answer: C

3. What keyword is used to define an abstract class in Java?

A. abstract
B. interface
C. class
D. final

Answer: A

4. Can an abstract class have a constructor?

A. Yes
B. No

Answer: A

5. Which of the following statements is correct?

A. An abstract class can be instantiated directly
B. An interface can contain constructors
C. Abstract methods must be implemented in a subclass
D. An abstract class cannot have final methods

Answer: C

6. How do you declare an abstract method?

A. abstract void methodName() { }
B. void methodName();
C. abstract void methodName();
D. abstract void methodName() { //implementation }

Answer: C

7. What happens if a subclass does not implement all abstract methods of its superclass?

A. It will compile but throw an error at runtime
B. It must be declared abstract
C. The abstract class will automatically provide the implementation
D. Nothing happens

Answer: B

8. Which of the following statements about interfaces is true?

A. Interfaces can have non-static fields
B. Interfaces can have both abstract and non-abstract methods
C. Interfaces can be extended by more than one class
D. Interfaces cannot have static methods

Answer: C

9. In Java, can a class implement multiple interfaces?

A. Yes
B. No

Answer: A

10. What is the default access modifier for interface methods in Java?

A. public
B. private
C. protected
D. default

Answer: A

11. Can an interface contain concrete methods?

A. Yes, starting from Java 8
B. No, interfaces can only have abstract methods
C. Yes, but only static methods
D. No, only abstract classes can have concrete methods

Answer: A

12. What keyword is used to implement an interface in Java?

A. extend
B. implement
C. import
D. inherit

Answer: B

13. What is true about a class that implements an interface?

A. It must implement all the methods of the interface
B. It must be declared as an abstract class
C. It can inherit from multiple classes at the same time
D. It cannot implement more than one interface

Answer: A

14. Can an abstract class implement an interface without providing implementation for all interface methods?

A. Yes
B. No

Answer: A

15. Which of the following is NOT true about an abstract class?

A. It can have instance variables
B. It can have static methods
C. It must have at least one abstract method
D. It cannot be instantiated

Answer: C

16. What is the default access modifier for fields in an interface?

A. public
B. private
C. protected
D. final

Answer: A

17. How are interface methods treated in Java 7?

A. Public and abstract by default
B. Private and abstract by default
C. Protected and abstract by default
D. Public and static by default

Answer: A

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

A. Abstract class can have instance methods, but interface cannot
B. Abstract class cannot have any methods, but an interface can
C. Abstract class supports multiple inheritance, but an interface does not
D. Interfaces are always public, but abstract classes can be private

Answer: A

19. Can an interface extend another interface?

A. Yes
B. No

Answer: A

20. What keyword is used to extend an abstract class in Java?

A. implement
B. extend
C. inherit
D. apply

Answer: B

21. Which of the following is a valid declaration of an interface?

A. abstract interface MyInterface {}
B. class MyInterface {}
C. interface MyInterface {}
D. interface MyInterface extends AnotherClass {}

Answer: C

22. Can an interface have private methods in Java 9 and later?

A. Yes
B. No

Answer: A

23. Which of the following is allowed in an abstract class?

A. Instantiating the abstract class
B. Declaring constructors
C. Defining private abstract methods
D. Declaring static abstract methods

Answer: B

24. What is a functional interface?

A. An interface with one or more abstract methods
B. An interface with exactly one abstract method
C. An interface that cannot be implemented
D. An interface with no methods

Answer: B

25. Can you declare an abstract method as final?

A. Yes
B. No

Answer: B

26. What happens if you attempt to instantiate an abstract class?

A. It throws an exception at runtime
B. It gives a compile-time error
C. The abstract methods are implemented automatically
D. The instantiation succeeds

Answer: B

27. In Java 8, what new feature was introduced for interfaces?

A. Support for default methods
B. Support for constructors
C. Support for protected fields
D. Support for multiple inheritance

Answer: A

28. Can an abstract class be final in Java?

A. Yes
B. No

Answer: B

29. What is the use of the default keyword in interfaces?

A. To define a method that can be overridden
B. To define a private method in the interface
C. To define a constructor
D. To define a final method

Answer: A

30. Can a class extend more than one abstract class in Java?

A. Yes
B. No

Answer: B

31. Can a class extend another class in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: A class can extend another class to inherit its properties and methods.

32. Can a class extend more than one class in Java (i.e., multiple inheritance)?

- a) Yes
- b) No

Answer: b) No
Explanation: Java does not support multiple inheritance for classes. A class can only extend one class.

33. Can a class implement an interface in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: A class can implement an interface, providing concrete implementations for its abstract methods.

34. Can a class implement multiple interfaces in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: A class can implement multiple interfaces in Java. This allows Java to achieve multiple inheritance through interfaces.

35. Can an abstract class extend a concrete class in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: An abstract class can extend a concrete class. The abstract class can inherit properties and methods, but it may still have abstract methods. A concrete class in Java is a class that has complete method implementations and can be instantiated to create objects.

36. Can an abstract class extend multiple classes in Java?

- a) Yes
- b) No

Answer: b) No
Explanation: Just like normal classes, abstract classes can only extend one class at a time.

37. Can an abstract class extend another abstract class in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: An abstract class can extend another abstract class. It may inherit abstract methods and provide implementations for them.

38. Can an abstract class extend more than one abstract class in Java?

- a) Yes
- b) No

Answer: b) No
Explanation: Java does not support multiple inheritance for classes, whether they are abstract or concrete.

39. Can an abstract class implement an interface in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: An abstract class can implement an interface, but it does not need to provide implementations for all methods, leaving some abstract.

40. Can an abstract class implement multiple interfaces in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: An abstract class can implement multiple interfaces, similar to how a concrete class can.

41. Can an interface extend another interface in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: An interface can extend another interface to inherit its abstract methods.

42. Can an interface extend multiple interfaces in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: Java allows interfaces to extend multiple interfaces, providing a form of multiple inheritance.

43. Can an interface implement a class in Java?

- a) Yes
- b) No

Answer: b) No
Explanation: An interface cannot implement a class. Only classes can implement interfaces.

44. Can an interface implement multiple classes in Java?

- a) Yes
- b) No

Answer: b) No
Explanation: Interfaces cannot implement classes. They can only extend other interfaces.

45. Can a class extend one class and implement multiple interfaces in Java?

- a) Yes
- b) No

Answer: a) Yes
Explanation: A class can extend one class and implement multiple interfaces, combining inheritance and interface implementation.