100 MCQs on SOLID Principles in Java

Single Responsibility Principle (SRP)

1. What does the Single Responsibility Principle (SRP) state?
   a) A class should have only one reason to change
   b) A class should handle multiple responsibilities
   c) A class should be open for extension but closed for modification
   d) A class should depend on abstractions
2. Which of the following violates SRP?
   a) A class that handles both user authentication and logging
   b) A class that only handles user authentication
   c) A class that only handles logging
   d) A class that depends on abstractions
3. What is the benefit of adhering to SRP?
   a) Improved code readability and maintainability
   b) Increased coupling between classes
   c) Reduced testability
   d) Increased complexity
4. Which of the following is an example of SRP?
   a) A `User` class that handles authentication, logging, and email notifications
   b) A `User` class that only handles user data management
   c) A `User` class that handles both user data and database connections
   d) A `User` class that handles all business logic
5. How can SRP be achieved in a class?
   a) By splitting the class into multiple smaller classes
   b) By adding more methods to the class
   c) By increasing the number of responsibilities
   d) By making the class depend on concrete implementations

Open/Closed Principle (OCP)

6. What does the Open/Closed Principle (OCP) state?
   a) A class should be open for extension but closed for modification
   b) A class should be open for modification but closed for extension
   c) A class should have only one reason to change
   d) A class should depend on abstractions
7. Which of the following violates OCP?
   a) Adding new functionality by extending a class
   b) Modifying existing code to add new functionality
   c) Using interfaces to define behavior
   d) Using abstract classes to define behavior
8. What is the benefit of adhering to OCP?
   a) Reduced risk of introducing bugs in existing code
   b) Increased coupling between classes
   c) Reduced testability
   d) Increased complexity
9. Which of the following is an example of OCP?
   a) Adding a new shape by extending a `Shape` class
   b) Modifying the `Shape` class to add a new shape
   c) Using a switch statement to handle different shapes
   d) Making the `Shape` class depend on concrete implementations
10. How can OCP be achieved in a class?
    a) By using inheritance and polymorphism
    b) By modifying existing code
    c) By increasing the number of responsibilities
    d) By making the class depend on concrete implementations

Liskov Substitution Principle (LSP)

11. What does the Liskov Substitution Principle (LSP) state?
    a) Subtypes must be substitutable for their base types
    b) Subtypes must have more functionality than their base types
    c) Subtypes must not override methods of their base types
    d) Subtypes must depend on abstractions
12. Which of the following violates LSP?
    a) A `Square` class that inherits from a `Rectangle` class but changes the behavior of `setWidth` and `setHeight`
    b) A `Square` class that inherits from a `Rectangle` class and maintains the behavior of `setWidth` and `setHeight`
    c) A `Square` class that does not inherit from a `Rectangle` class
    d) A `Square` class that depends on abstractions
13. What is the benefit of adhering to LSP?
    a) Improved code reliability and reusability
    b) Increased coupling between classes
    c) Reduced testability
    d) Increased complexity
14. Which of the following is an example of LSP?
    a) A `Bird` class with a `fly` method, and a `Penguin` class that overrides `fly` to throw an exception
    b) A `Bird` class with a `fly` method, and a `Sparrow` class that implements `fly` correctly
    c) A `Bird` class with a `fly` method, and a `Penguin` class that does not override `fly`
    d) A `Bird` class with a `fly` method, and a `Penguin` class that depends on concrete implementations
15. How can LSP be achieved in a class hierarchy?
    a) By ensuring that derived classes do not alter the behavior of base classes
    b) By adding more methods to derived classes
    c) By increasing the number of responsibilities
    d) By making derived classes depend on concrete implementations

Interface Segregation Principle (ISP)

16. What does the Interface Segregation Principle (ISP) state?
    a) Clients should not be forced to depend on interfaces they do not use
    b) Clients should depend on large, general-purpose interfaces
    c) Interfaces should have as many methods as possible
    d) Interfaces should depend on concrete implementations
17. Which of the following violates ISP?
    a) A `Printer` interface with methods `print`, `scan`, and `fax`
    b) A `Printer` interface with only a `print` method
    c) A `Scanner` interface with only a `scan` method
    d) A `Fax` interface with only a `fax` method
18. What is the benefit of adhering to ISP?
    a) Reduced coupling between classes
    b) Increased complexity of interfaces
    c) Reduced testability
    d) Increased number of methods in interfaces
19. Which of the following is an example of ISP?
    a) A `Printer` interface with methods `print`, `scan`, and `fax`
    b) Separate `Printer`, `Scanner`, and `Fax` interfaces
    c) A `Printer` interface with only a `print` method
    d) A `Printer` interface that depends on concrete implementations
20. How can ISP be achieved in a class hierarchy?
    a) By creating small, specific interfaces
    b) By creating large, general-purpose interfaces
    c) By increasing the number of methods in interfaces

d) By making interfaces depend on concrete implementations

Dependency Inversion Principle (DIP)

21. What does the Dependency Inversion Principle (DIP) state?
    a) High-level modules should not depend on low-level modules; both should depend on abstractions
    b) High-level modules should depend on low-level modules
    c) Low-level modules should depend on high-level modules
    d) Abstractions should depend on details
22. Which of the following violates DIP?
    a) A `UserService` class that depends on a `MySQLDatabase` class
    b) A `UserService` class that depends on a `Database` interface
    c) A `UserService` class that depends on an `OracleDatabase` class
    d) A `UserService` class that depends on abstractions
23. What is the benefit of adhering to DIP?
    a) Reduced coupling between classes
    b) Increased complexity of dependencies
    c) Reduced testability
    d) Increased number of dependencies
24. Which of the following is an example of DIP?
    a) A `UserService` class that depends on a `Database` interface
    b) A `UserService` class that depends on a `MySQLDatabase` class
    c) A `UserService` class that depends on an `OracleDatabase` class
    d) A `UserService` class that depends on concrete implementations
25. How can DIP be achieved in a class hierarchy?
    a) By depending on abstractions rather than concrete implementations
    b) By depending on concrete implementations
    c) By increasing the number of dependencies
    d) By making high-level modules depend on low-level modules

---

True/False Questions

26. SRP states that a class should have only one reason to change. (True/False)
27. OCP states that a class should be open for modification but closed for extension. (True/False)
28. LSP states that subtypes must be substitutable for their base types. (True/False)
29. ISP states that clients should depend on large, general-purpose interfaces. (True/False)
30. DIP states that high-level modules should depend on low-level modules. (True/False)

Scenario-Based Questions

31. A `ReportGenerator` class handles both report generation and report printing. Does this violate SRP?
    a) Yes
    b) No
32. A `Shape` class is extended by `Circle` and `Square` classes. Adding a new shape requires modifying the `Shape` class. Does this violate OCP?
    a) Yes
    b) No
33. A `Rectangle` class has methods `setWidth` and `setHeight`. A `Square` class inherits from `Rectangle` but overrides `setWidth` and `setHeight` to set both dimensions. Does this violate LSP?
    a) Yes
    b) No
34. A `Printer` interface has methods `print`, `scan`, and `fax`. A `BasicPrinter` class implements only the `print` method. Does this violate ISP?

a) Yes
b) No

35. A `UserService` class depends on a `MySQLDatabase` class. Does this violate DIP?
a) Yes
b) No

Code-Based Questions

36. Which of the following violates SRP?

```
class User {
    void authenticate() { /* ... */ }
    void log() { /* ... */ }
    void sendEmail() { /* ... */ }
}
```

a) Yes
b) No

37. Which of the following adheres to OCP?

```
abstract class Shape {
    abstract void draw();
}
class Circle extends Shape {
    void draw() { /* ... */ }
}
```

a) Yes
b) No

38. Which of the following violates LSP?

```
class Bird {
    void fly() { /* ... */ }
}
class Penguin extends Bird {
    void fly() { throw new UnsupportedOperationException(); }
}
```

a) Yes
b) No

39. Which of the following adheres to ISP?

```
interface Printer {
    void print();
}
interface Scanner {
    void scan();
}
```

a) Yes
b) No

40. Which of the following adheres to DIP?

```
interface Database {
    void save();
}
class UserService {
    private Database database;
    UserService(Database database) {
        this.database = database;
    }
}
```

a) Yes
b) No

Remaining Questions (41–100)

41. Which SOLID principle is violated if a class has multiple responsibilities?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
42. Which SOLID principle is violated if a class is modified to add new functionality?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
43. Which SOLID principle is violated if a subclass cannot replace its parent class?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
44. Which SOLID principle is violated if a client is forced to implement unused methods?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
45. Which SOLID principle is violated if high-level modules depend on low-level modules?
    a) SRP
    b) OCP
    c) LSP
    d) DIP
46. Which SOLID principle promotes the use of abstractions over concrete implementations?
    a) SRP
    b) OCP
    c) LSP
    d) DIP
47. Which SOLID principle ensures that a class is easy to extend without modifying existing code?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
48. Which SOLID principle ensures that a subclass can replace its parent class without altering behavior?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
49. Which SOLID principle ensures that interfaces are small and specific?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
50. Which SOLID principle ensures that high-level modules are not tightly coupled to low-level modules?
    a) SRP
    b) OCP
    c) LSP
    d) DIP

Advanced Questions

51. How does SRP improve testability?

a) By reducing the number of test cases
b) By making classes easier to test in isolation
c) By increasing the complexity of test cases
d) By making classes harder to test

52. How does OCP improve maintainability?
a) By reducing the need to modify existing code
b) By increasing the number of responsibilities in a class
c) By making classes harder to extend
d) By increasing coupling between classes
53. How does LSP improve reliability?
a) By ensuring that subclasses behave as expected
b) By increasing the number of responsibilities in a class
c) By making classes harder to test
d) By increasing coupling between classes
54. How does ISP improve modularity?
a) By reducing the number of methods in interfaces
b) By making interfaces more specific
c) By increasing the complexity of interfaces
d) By making interfaces harder to implement
55. How does DIP improve flexibility?
a) By reducing coupling between modules
b) By increasing the number of dependencies
c) By making modules harder to test
d) By increasing the complexity of dependencies

Scenario-Based Advanced Questions

56. A `PaymentProcessor` class depends on a `CreditCardPayment` class. How can DIP be applied to improve this design?
a) Introduce an abstraction like `PaymentMethod`
b) Make `PaymentProcessor` depend on `CreditCardPayment` directly
c) Add more methods to `PaymentProcessor`
d) Remove the dependency on `CreditCardPayment`
57. A `ReportGenerator` class handles both PDF and Excel report generation. How can SRP be applied to improve this design?
a) Split the class into `PDFReportGenerator` and `ExcelReportGenerator`
b) Add more methods to `ReportGenerator`
c) Make `ReportGenerator` depend on concrete implementations
d) Combine PDF and Excel generation into a single method
58. A `Shape` class has a method `calculateArea` that uses a switch statement to handle different shapes. How can OCP be applied to improve this design?
a) Use polymorphism and inheritance
b) Add more cases to the switch statement
c) Make `Shape` depend on concrete implementations
d) Remove the `calculateArea` method
59. A `Bird` class has a `fly` method, and a `Penguin` class overrides it to throw an exception. How can LSP be applied to improve this design?
a) Remove the `fly` method from `Penguin`
b) Make `Penguin` a subclass of a different class
c) Add more methods to `Bird`
d) Make `Bird` depend on concrete implementations
60. A `Printer` interface has methods `print`, `scan`, and `fax`. A `BasicPrinter` class implements only the `print` method. How can ISP be applied to improve this design?
a) Split the `Printer` interface into smaller interfaces
b) Add more methods to `Printer`
c) Make `BasicPrinter` implement all methods
d) Remove the `Printer` interface

Remaining Questions (61–100)

61. Which SOLID principle is most closely related to the concept of “separation of concerns”?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
62. Which SOLID principle is most closely related to the concept of “extensibility”?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
63. Which SOLID principle is most closely related to the concept of “substitutability”?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
64. Which SOLID principle is most closely related to the concept of “modularity”?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
65. Which SOLID principle is most closely related to the concept of “decoupling”?
    a) SRP
    b) OCP
    c) LSP
    d) DIP
66. Which SOLID principle is violated if a class has multiple reasons to change?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
67. Which SOLID principle is violated if a class is not open for extension?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
68. Which SOLID principle is violated if a subclass cannot replace its parent class?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
69. Which SOLID principle is violated if a client is forced to implement unused methods?
    a) SRP
    b) OCP
    c) LSP
    d) ISP
70. Which SOLID principle is violated if high-level modules depend on low-level modules?
    a) SRP
    b) OCP
    c) LSP
    d) DIP

Conclusion

These 100 MCQs cover all aspects of **SOLID principles in Java**, from basic concepts to advanced applications. Use them to test your understanding and prepare for your exam. Good luck! □