

42 SOLID interview questions to ask your candidates

Questions

1. Can you explain what SOLID stands for and give a brief overview of each principle?
2. How does the Single Responsibility Principle improve code maintainability?
3. Provide an example of how you've applied the Open-Closed Principle in a recent project.
4. What's the main difference between the Liskov Substitution Principle and the Interface Segregation Principle?
5. How does the Dependency Inversion Principle relate to dependency injection?
6. Can you describe a situation where violating the Single Responsibility Principle led to issues in a project?
7. How would you refactor a large class to better adhere to the Single Responsibility Principle?
8. Explain how the Open-Closed Principle can help in reducing the risk of bugs when adding new features.
9. What are some common pitfalls when trying to implement the Liskov Substitution Principle?
10. How does the Interface Segregation Principle help in creating more flexible and maintainable systems?
11. Can you give an example of how you've used the Dependency Inversion Principle to improve testability?
12. How do SOLID principles contribute to creating more scalable software architectures?
13. What challenges have you faced when trying to apply SOLID principles in a legacy codebase?
14. How do you balance applying SOLID principles with meeting project deadlines?
15. Can you describe how SOLID principles relate to other design patterns or architectural concepts you've used?
16. Can you explain a real-world scenario where applying the Single Responsibility Principle improved a system's design?
17. How would you explain the Open-Closed Principle to a junior developer who's never heard of SOLID?
18. Can you describe a situation where strictly adhering to the Liskov Substitution Principle might be challenging?
19. How would you refactor a 'God Object' to better adhere to SOLID principles?
20. How does the Interface Segregation Principle relate to microservices architecture?
21. Can you give an example of how violating the Dependency Inversion Principle could lead to problems in a large-scale application?
22. How would you balance applying SOLID principles with the need for rapid development in a startup environment?
23. How do SOLID principles contribute to creating more testable code?
24. Can you describe a scenario where applying the Open-Closed Principle resulted in improved code quality?
25. What strategies do you use to ensure compliance with the Liskov Substitution Principle in your code?
26. How do you approach testing when implementing the Dependency Inversion Principle?
27. Can you share an experience where following the Interface Segregation Principle enhanced team collaboration?
28. What is the significance of the Dependency Inversion Principle in microservices architecture?
29. How would you explain the importance of SOLID principles to a non-technical stakeholder?
30. In your opinion, how do SOLID principles influence code readability and team productivity?
31. Can you give an example of how you've adapted SOLID principles in a rapidly changing project environment?
32. How do you determine which SOLID principle to prioritize when facing design challenges?
33. What tools or practices do you find helpful for enforcing SOLID principles in your codebase?
34. Can you discuss a time when applying the Single Responsibility Principle led to a more efficient design?
35. How do you ensure that your code remains flexible yet adheres to the Open-Closed Principle?
36. How does adhering to the SOLID principles benefit software development projects?
37. Can you explain a scenario where the Interface Segregation Principle was particularly useful?
38. What are the challenges you might face when applying the Open-Closed Principle in an existing codebase?
39. How can the Dependency Inversion Principle improve software testability?