

Examen Analyste Programmeur – Niveau Tech Lead

UML

1. Quelle est la différence entre un diagramme de classes et un diagramme de séquence ? (vision architecture / décision stratégique)
2. À quoi sert une relation d'agrégation en UML ? (vision architecture / décision stratégique)
3. Quand utiliser un diagramme d'activités ? (vision architecture / décision stratégique)
4. Expliquez la notion d'héritage en UML. (vision architecture / décision stratégique)
5. Quelle est l'utilité des interfaces en UML ? (vision architecture / décision stratégique)
6. Différence entre association et composition ? (vision architecture / décision stratégique)
7. Que représente une multiplicité ? (vision architecture / décision stratégique)
8. Diagramme UML le plus adapté pour modéliser un workflow ? (vision architecture / décision stratégique)

Networking

9. Expliquez le modèle OSI. (vision architecture / décision stratégique)
10. Différence entre TCP et UDP. (vision architecture / décision stratégique)
11. À quoi sert le DNS ? (vision architecture / décision stratégique)
12. Qu'est-ce qu'une adresse IP privée ? (vision architecture / décision stratégique)
13. Différence entre HTTP et HTTPS ? (vision architecture / décision stratégique)
14. Qu'est-ce qu'un pare-feu ? (vision architecture / décision stratégique)
15. Expliquez le rôle d'un load balancer. (vision architecture / décision stratégique)
16. Qu'est-ce que la latence réseau ? (vision architecture / décision stratégique)

OOP

17. Expliquez les quatre piliers de la POO. (vision architecture / décision stratégique)
18. Différence entre abstraction et encapsulation ? (vision architecture / décision stratégique)
19. Qu'est-ce que le polymorphisme ? (vision architecture / décision stratégique)
20. Quand utiliser une classe abstraite ? (vision architecture / décision stratégique)
21. Différence entre héritage et composition ? (vision architecture / décision stratégique)
22. Qu'est-ce qu'un constructeur ? (vision architecture / décision stratégique)
23. Avantages de la POO ? (vision architecture / décision stratégique)

24. Exemple de surcharge de méthode. (vision architecture / décision stratégique)

DSA

25. Différence entre tableau et liste chaînée. (vision architecture / décision stratégique)

26. Complexité temporelle de la recherche linéaire. (vision architecture / décision stratégique)

27. Qu'est-ce qu'une pile (stack) ? (vision architecture / décision stratégique)

28. Qu'est-ce qu'une file (queue) ? (vision architecture / décision stratégique)

29. Complexité moyenne du tri rapide. (vision architecture / décision stratégique)

30. À quoi sert une table de hachage ? (vision architecture / décision stratégique)

31. Différence entre BFS et DFS. (vision architecture / décision stratégique)

32. Qu'est-ce qu'un arbre binaire ? (vision architecture / décision stratégique)

Design Patterns

33. À quoi sert le pattern Singleton ? (vision architecture / décision stratégique)

34. Différence entre Factory et Abstract Factory. (vision architecture / décision stratégique)

35. Quand utiliser Observer ? (vision architecture / décision stratégique)

36. Avantages du pattern Strategy. (vision architecture / décision stratégique)

37. Pattern MVC : rôle du contrôleur ? (vision architecture / décision stratégique)

38. Pattern Adapter : cas d'usage. (vision architecture / décision stratégique)

39. Différence entre State et Strategy ? (vision architecture / décision stratégique)

40. Pourquoi éviter l'abus de Singleton ? (vision architecture / décision stratégique)

Backend Patterns

41. Qu'est-ce qu'une architecture REST ? (vision architecture / décision stratégique)

42. Différence entre monolithe et microservices. (vision architecture / décision stratégique)

43. À quoi sert un API Gateway ? (vision architecture / décision stratégique)

44. Qu'est-ce que la pagination côté backend ? (vision architecture / décision stratégique)

45. Gestion des erreurs dans une API ? (vision architecture / décision stratégique)

46. Qu'est-ce que l'idempotence ? (vision architecture / décision stratégique)

47. Rôle des middlewares. (vision architecture / décision stratégique)

48. Qu'est-ce qu'un service stateless ? (vision architecture / décision stratégique)

SQL & Base de données

- 49. Différence entre DELETE et TRUNCATE. (vision architecture / décision stratégique)
- 50. À quoi sert une clé primaire ? (vision architecture / décision stratégique)
- 51. Qu'est-ce qu'une clé étrangère ? (vision architecture / décision stratégique)
- 52. Différence entre INNER JOIN et LEFT JOIN. (vision architecture / décision stratégique)
- 53. Qu'est-ce qu'une transaction ? (vision architecture / décision stratégique)
- 54. Niveaux d'isolation des transactions. (vision architecture / décision stratégique)
- 55. Index : avantages et inconvénients. (vision architecture / décision stratégique)
- 56. Qu'est-ce que la normalisation ? (vision architecture / décision stratégique)

Frontend Patterns

- 57. Qu'est-ce que MVC côté frontend ? (vision architecture / décision stratégique)
- 58. Différence entre SPA et MPA. (vision architecture / décision stratégique)
- 59. Rôle d'un state manager. (vision architecture / décision stratégique)
- 60. Qu'est-ce que le data binding ? (vision architecture / décision stratégique)
- 61. Avantages des composants réutilisables. (vision architecture / décision stratégique)
- 62. Qu'est-ce que le lazy loading ? (vision architecture / décision stratégique)
- 63. Gestion des formulaires côté frontend. (vision architecture / décision stratégique)
- 64. Différence entre props et state. (vision architecture / décision stratégique)