

# **Examen Analyste Programmeur – Niveau Intermédiaire**

## **UML**

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

## **Networking**

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

## **OOP**

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

24. Exemple de surcharge de méthode.

## DSA

- 25. Différence entre tableau et liste chaînée.
- 26. Complexité temporelle de la recherche linéaire.
- 27. Qu'est-ce qu'une pile (stack) ?
- 28. Qu'est-ce qu'une file (queue) ?
- 29. Complexité moyenne du tri rapide.
- 30. À quoi sert une table de hachage ?
- 31. Différence entre BFS et DFS.
- 32. Qu'est-ce qu'un arbre binaire ?

## Design Patterns

- 33. À quoi sert le pattern Singleton ?
- 34. Différence entre Factory et Abstract Factory.
- 35. Quand utiliser Observer ?
- 36. Avantages du pattern Strategy.
- 37. Pattern MVC : rôle du contrôleur ?
- 38. Pattern Adapter : cas d'usage.
- 39. Différence entre State et Strategy ?
- 40. Pourquoi éviter l'abus de Singleton ?

## Backend Patterns

- 41. Qu'est-ce qu'une architecture REST ?
- 42. Différence entre monolithe et microservices.
- 43. À quoi sert un API Gateway ?
- 44. Qu'est-ce que la pagination côté backend ?
- 45. Gestion des erreurs dans une API ?
- 46. Qu'est-ce que l'idempotence ?
- 47. Rôle des middlewares.
- 48. Qu'est-ce qu'un service stateless ?

## **SQL & Base de données**

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

## **Frontend Patterns**

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