

# Examen Analyste Programmeur (Banque) – 8 Axes – Senior (Questions + Réponses)

## UML

**Q1.** Héritage vs interface (réalisation) en UML ? (niveau senior)

**Réponse :** Héritage = généralisation (est-un). Interface = contrat de comportements sans implémentation (selon langage). (attendu : justification, trade-offs, exemples concrets).

**Q2.** Quand utiliser un diagramme d'activités ? (niveau senior)

**Réponse :** Pour modéliser un workflow/processus (décisions, parallélisme), ex. ouverture de compte, traitement de virement. (attendu : justification, trade-offs, exemples concrets).

**Q3.** Diagramme de composants : intérêt ? (niveau senior)

**Réponse :** Visualiser modules/services, dépendances, interfaces exposées, utile en architecture (microservices, APIs). (attendu : justification, trade-offs, exemples concrets).

**Q4.** Diagramme d'états : cas d'usage ? (niveau senior)

**Réponse :** Pour objets à cycle de vie riche (ex. Transaction: INITIÉE→AUTORISÉE→COMPTABILISÉE→ANNULÉE). (attendu : justification, trade-offs, exemples concrets).

**Q5.** Agrégation : à quoi sert-elle ? (niveau senior)

**Réponse :** Relation tout/partie faible : la partie peut exister sans le tout (ex. Banque agrège des Agences). (attendu : justification, trade-offs, exemples concrets).

**Q6.** Association vs composition ? (niveau senior)

**Réponse :** Association = lien logique (cycle de vie indépendant). Composition = relation tout/partie forte (la partie dépend du tout, cycle de vie lié). (attendu : justification, trade-offs, exemples concrets).

**Q7.** Qu'est-ce qu'un stéréotype UML (<>, <>...) ? (niveau senior)

**Réponse :** Extension sémantique pour préciser le rôle d'un élément selon un profil (ex. UML pour logiciel). (attendu : justification, trade-offs, exemples concrets).

## Networking

**Q8.** Load balancer L4 vs L7 ? (niveau senior)

**Réponse :** L4 route sur IP/port (TCP/UDP). L7 comprend HTTP (URL, headers), permet routage avancé, WAF, etc. (attendu : justification, trade-offs, exemples concrets).

**Q9.** NAT : pourquoi l'utiliser ? (niveau senior)

**Réponse :** Traduction d'adresses privées vers publique, économise IPv4 et masque topologie interne. (attendu : justification, trade-offs, exemples concrets).

**Q10.** CIDR : que signifie /24 ? (niveau senior)

**Réponse :** Préfixe réseau de 24 bits; en IPv4 cela correspond à 256 adresses (dont 254 utilisables en pratique). (attendu : justification, trade-offs, exemples concrets).

**Q11.** Que fait un pare-feu stateful ? (niveau senior)

**Réponse :** Suit l'état des connexions (table d'états) pour autoriser le trafic de retour et appliquer des règles contextuelles. (attendu : justification, trade-offs, exemples concrets).

**Q12.** Qu'est-ce qu'un reverse proxy ? (niveau senior)

**Réponse :** Proxy côté serveur, termine TLS, fait cache, routage, protection, masquant les services internes. (attendu : justification, trade-offs, exemples concrets).

**Q13.** TCP vs UDP : principales différences ? (niveau senior)

**Réponse :** TCP est orienté connexion, fiable (ACK, retransmission, ordre). UDP est sans connexion, plus léger, pas de garantie d'ordre/fiabilité. (attendu : justification, trade-offs, exemples concrets).

**Q14.** DNS : rôle et mécanisme haut niveau ? (niveau senior)

**Réponse :** Résolution nom→IP via hiérarchie (root/TLD/autoritatifs) et cache (TTL). (attendu : justification, trade-offs, exemples concrets).

## OOP

**Q15.** Qu'est-ce qu'une interface fonctionnelle ? (niveau senior)

**Réponse :** Interface avec une seule méthode abstraite (Java), compatible lambda. (attendu : justification, trade-offs, exemples concrets).

**Q16.** Héritage vs composition : règle pratique ? (niveau senior)

**Réponse :** Préférer la composition (plus flexible) sauf vraie relation 'est-un' stable. (attendu : justification, trade-offs, exemples concrets).

**Q17.** Polymorphisme : exemple ? (niveau senior)

**Réponse :** Une variable de type Interface peut référencer différentes implémentations; dispatch dynamique à l'exécution. (attendu : justification, trade-offs, exemples concrets).

**Q18.** Inversion of Control : idée ? (niveau senior)

**Réponse :** Le framework orchestre la création/injection (DI), l'application fournit des composants. (attendu : justification, trade-offs, exemples concrets).

**Q19.** Qu'est-ce qu'un objet valeur (Value Object) ? (niveau senior)

**Réponse :** Objet identifié par ses valeurs (equals/hashCode), immutable; ex. IBAN, Money. (attendu : justification, trade-offs, exemples concrets).

**Q20.** Exception checked vs unchecked (Java) ? (niveau senior)

**Réponse :** Checked = déclarée/traitée (compile-time), unchecked = RuntimeException; choisir selon récupérabilité. (attendu : justification, trade-offs, exemples concrets).

**Q21.** Encapsulation : bénéfice concret ? (niveau senior)

**Réponse :** Masquer l'état interne, contrôler invariants via méthodes, réduire couplage. (attendu : justification, trade-offs, exemples concrets).

## DSA

**Q22.** Trie : pour quoi faire ? (niveau senior)

**Réponse :** Recherche préfixe/auto-complétion en  $O(L)$  (longueur du mot). (attendu : justification, trade-offs, exemples concrets).

**Q23.** Complexité Big-O : que mesure-t-elle ? (niveau senior)

**Réponse :** Croissance asymptotique du coût (temps/espace) selon  $n$ , en ignorant constantes. (attendu : justification, trade-offs, exemples concrets).

**Q24.** BFS vs DFS : quand choisir ? (niveau senior)

**Réponse :** BFS pour plus court chemin non pondéré; DFS pour détection cycles, topologie, backtracking. (attendu : justification, trade-offs, exemples concrets).

**Q25.** Tri rapide : moyenne vs pire ? (niveau senior)

**Réponse :** Moyenne  $O(n \log n)$ , pire  $O(n^2)$  si pivot mauvais; randomisation réduit risque. (attendu : justification, trade-offs, exemples concrets).

**Q26.** Deux pointeurs : exemple classique ? (niveau senior)

**Réponse :** Détecter paire somme cible dans tableau trié, ou retirer doublons in-place. (attendu : justification, trade-offs, exemples concrets).

**Q27.** HashMap : complexité moyenne insertion/recherche ? (niveau senior)

**Réponse :** Moyenne  $O(1)$ , pire  $O(n)$  (collisions/pathologiques), atténué par resizing/arbres (Java). (attendu : justification, trade-offs, exemples concrets).

**Q28.** Arbre équilibré (AVL/RB) : intérêt ? (niveau senior)

**Réponse :** Garantir hauteur  $O(\log n) \Rightarrow$  opérations  $O(\log n)$  stables. (attendu : justification, trade-offs, exemples concrets).

## Design Patterns

**Q29.** Adapter : quand ? (niveau senior)

**Réponse :** Intégrer API legacy/tiers avec interface différente (ex. switch de paiement). (attendu : justification, trade-offs, exemples concrets).

**Q30.** Factory Method vs Abstract Factory ? (niveau senior)

**Réponse :** Factory Method crée un produit via héritage. Abstract Factory fournit familles de produits compatibles. (attendu : justification, trade-offs, exemples concrets).

**Q31.** Observer : exemple ? (niveau senior)

**Réponse :** Publication d'événements (transaction postée) vers notifications, anti-fraude, reporting. (attendu : justification, trade-offs, exemples concrets).

**Q32.** CQRS : principe ? (niveau senior)

**Réponse :** Séparer modèles Command (écriture) et Query (lecture) pour scalabilité/perf. (attendu : justification, trade-offs, exemples concrets).

**Q33.** Decorator : bénéfice ? (niveau senior)

**Réponse :** Ajouter responsabilités dynamiquement sans modifier la classe (ex. logging, métriques). (attendu : justification, trade-offs, exemples concrets).

**Q34.** Strategy : cas d'usage bancaire ? (niveau senior)

**Réponse :** Sélection de calcul de frais/commission selon type client/produit. (attendu : justification, trade-offs, exemples concrets).

**Q35.** Circuit Breaker : pourquoi en microservices ? (niveau senior)

**Réponse :** Éviter surcharge/cascades; ouvrir circuit après erreurs pour donner temps de récupération. (attendu : justification, trade-offs, exemples concrets).

## Backend Patterns

**Q36.** AuthN/AuthZ : différence ? (niveau senior)

**Réponse :** AuthN = prouver identité; AuthZ = droits/permissions. (attendu : justification, trade-offs, exemples concrets).

**Q37.** Gestion erreurs API : bonnes pratiques ? (niveau senior)

**Réponse :** Codes HTTP, message clair, trace-id, pas de fuite sensible, mapping exceptions. (attendu : justification, trade-offs, exemples concrets).

**Q38.** Observabilité : 3 piliers ? (niveau senior)

**Réponse :** Logs, métriques, traces distribuées (OpenTelemetry). (attendu : justification, trade-offs, exemples concrets).

**Q39.** Idempotence : exemple en banque ? (niveau senior)

**Réponse :** POST virement avec idempotency-key : répéter la requête ne double pas l'opération. (attendu : justification, trade-offs, exemples concrets).

**Q40.** Saga : pourquoi ? (niveau senior)

**Réponse :** Gérer transactions distribuées via compensations (ex. réservation + débit + confirmation). (attendu : justification, trade-offs, exemples concrets).

**Q41.** Caching : où et quoi ? (niveau senior)

**Réponse :** CDN/reverse proxy/app/DB; cache des lectures idempotentes, invalidation/TTL. (attendu : justification, trade-offs, exemples concrets).

**Q42.** Rate limiting : comment ? (niveau senior)

**Réponse :** Token bucket/leaky bucket; protéger API contre abus/DDoS, contrôler coûts. (attendu : justification, trade-offs, exemples concrets).

## SQL & Bases de données

**Q43.** EXPLAIN : pourquoi ? (niveau senior)

**Réponse :** Voir plan d'exécution (index scan, seq scan), diagnostiquer lenteurs. (attendu : justification, trade-offs, exemples concrets).

**Q44.** Partitionnement : quand ? (niveau senior)

**Réponse :** Très grandes tables (transactions), améliore maintenance/perf par pruning. (attendu : justification, trade-offs, exemples concrets).

**Q45.** Deadlock : c'est quoi ? (niveau senior)

**Réponse :** Cycle d'attente entre transactions; SGBD choisit une victime et rollback. (attendu : justification, trade-offs, exemples concrets).

**Q46.** Niveaux isolation : exemple d'anomalie ? (niveau senior)

**Réponse :** READ COMMITTED évite dirty reads mais pas non-repeatable/phantom reads. (attendu : justification, trade-offs, exemples concrets).

**Q47.** Clé primaire vs unique ? (niveau senior)

**Réponse :** PK identifie ligne (non NULL, 1 par table). UNIQUE impose unicité (peut être multiple, NULL selon SGBD). (attendu : justification, trade-offs, exemples concrets).

**Q48.** Transactions : ACID ? (niveau senior)

**Réponse :** Atomicité, Cohérence, Isolation, Durabilité. (attendu : justification, trade-offs, exemples concrets).

**Q49.** Normalisation : objectif ? (niveau senior)

**Réponse :** Réduire redondance et anomalies (update/insert/delete), via formes normales. (attendu : justification, trade-offs, exemples concrets).

## Frontend Patterns

**Q50.** Sécurité front : XSS ? (niveau senior)

**Réponse :** Injection de script; éviter innerHTML, échapper/sanitizer, CSP. (attendu : justification, trade-offs, exemples concrets).

**Q51.** Lazy loading : bénéfice ? (niveau senior)

**Réponse :** Réduire bundle initial, améliorer performance (LCP) en chargeant à la demande. (attendu : justification, trade-offs, exemples concrets).

**Q52.** Props vs state ? (niveau senior)

**Réponse :** Props = données reçues (immutables côté enfant). State = état interne composant. (attendu : justification, trade-offs, exemples concrets).

**Q53.** Gestion formulaires : bonnes pratiques ? (niveau senior)

**Réponse :** Validation côté client + serveur, messages accessibles, prévention CSRF, masquage données sensibles. (attendu : justification, trade-offs, exemples concrets).

**Q54.** Performance : re-render inutile ? (niveau senior)

**Réponse :** Memoization, keys stables, découper composants, éviter state global abusif. (attendu : justification, trade-offs, exemples concrets).

**Q55.** SPA vs MPA ? (niveau senior)

**Réponse :** SPA charge app JS et navigue côté client; MPA recharge pages côté serveur, plus simple SEO par défaut. (attendu : justification, trade-offs, exemples concrets).

**Q56.** CSRF : mitigation ? (niveau senior)

**Réponse :** SameSite cookies, tokens anti-CSRF, double submit, vérifier origin. (attendu : justification, trade-offs, exemples concrets).

## Questions bonus

**Q57.** [Bonus] INNER JOIN vs LEFT JOIN ? (Senior)

**Réponse :** INNER: intersection. LEFT: toutes lignes de gauche + NULL si pas de match à droite.

**Q58.** [Bonus] Saga : pourquoi ? (Senior)

**Réponse :** Gérer transactions distribuées via compensations (ex. réservation + débit + confirmation).

**Q59.** [Bonus] TLS/HTTPS : que protège-t-on ? (Senior)

**Réponse :** Confidentialité (chiffrement), intégrité (MAC), authentification (certificats).

**Q60.** [Bonus] Template Method : idée ? (Senior)

**Réponse :** Squelette d'algorithme dans une classe mère, étapes personnalisées dans sous-classes.