

## JAVA

- Citez les différents types de visibilité en java (private, public, package, default = package level)
  - Qu'est-ce qui différencie ces visibilités ?
  - Si on ne met rien comme visibilité devant un attribut ou une méthode dans une classe, quelle est sa visibilité ? (package level)
  - Même question pour une interface ?
- Qu'est-ce qu'une interface ? A quoi cela sert-il ?
- Qu'est-ce qu'une classe abstraite ?
  - Une classe abstraite peut-elle avoir un constructeur ?
  - Quelle est la différence avec une interface ? Quel est l'intérêt de l'une par rapport à l'autre ?
- Comment accède-t-on aux méthodes / attributs d'une classe mère ?
  - Est-ce qu'on peut appeler le constructeur de la classe mère n'importe où dans son propre constructeur ?
- Comment fait-on pour qu'un attribut ne soit pas modifiable ?
  - Est-ce qu'on peut utiliser ce modificateur (final) à d'autres endroits que sur des attributs ? Qu'est-ce qu'il se passe alors ?
  - Est-ce que le paramètre d'une méthode (de type int par exemple) est modifié à la sortie de la méthode si la méthode le modifie ?
- Pouvez-vous nous citer les différents types de collections en java ? (List, Set, Map, Queue)
  - Donnez nous un exemple d'implémentation de List ? (ArrayList)
  - Y a-t-il d'autres implémentations ? (LinkedList)
  - Quelle est la différence entre les deux ? Quand utiliser l'une ou l'autre ?
- Qu'est-ce que la sérialisation ? A quoi ça sert ?
  - Comment sérialise-t-on un objet ? (implémenter Serializable)
  - Il y a une autre interface qui permet de sérialiser un objet, laquelle ?
  - A quoi sert le mot clé transient ?
- A quoi sert le mot clé volatile ?
- Comment met-on en place le multi threading ? (la classe hérite de Thread)
  - Il y a une autre façon de le faire, laquelle ? (implémenter Runnable)
  - Pourquoi cette solution est-elle moins utilisée ?

- Comment « sécuriser » une donnée partagée dans un environnement multi threadé ?
- A quel niveau est positionné le lock quand on utilise « synchronized » ?
- Si on met un « synchronized » sur une méthode statique, où est placé le lock ?
- Comment sont gérées les exceptions en java ?
  - Quelles sont les différentes méthodes pour lever une exception ?
  - Est-ce qu'on peut gérer les exceptions sans block try catch ?
  - Quand on ne gère pas un type d'exception et qu'elle est levée, qu'est-ce qu'il se passe ?
- Quels sont les 4 types d'encapsulation en Java ?
  - Privée : attribut ou méthode accessible uniquement à l'intérieur de la classe
  - Protégé : attribut ou méthode accessible au sein de la classe ainsi que dans les classes filles et également aux classes du package
  - Publique : attribut ou méthode accessible aussi bien à l'intérieur qu'à l'extérieur de la classe
  - Défaut : attribut ou méthode accessible par les classes présentes dans le même package
- Citez-moi des design pattern que vous connaissez
  - Pouvez-vous m'écrire une implémentation du pattern singleton ?
  - Singleton est-il thread safe ? [Le design pattern est thread safe si l'on ajoute le mot-clé synchronized à la méthode getInstance()]
  - Quels sont les 3 types principaux de design patterns ? C'est quoi le but de chaque groupe ? Donnez un exemple pour chacun.
- Qu'est-ce que Spring ?
  - Comment cela fonctionne-t-il ? A quoi ça sert ?
  - Quel design pattern est utilisé par spring ? => Définir Inversion of control et dependency injection (<http://martinfowler.com/articles/injection.html>) Pourquoi les utiliser ?
- Qu'est-ce qu'Hibernate ?
  - Comment cela fonctionne-t-il ? A quoi ça sert ?
  - Qu'est-ce que le lazy-loading ?
  - Quelles sont les différences entre les niveaux de cache ?
  - Quel design pattern est utilisé dans ce cas ?
- Avec quoi lisez-vous un fichier xml en java ? (SAX ou DOM)
  - Quelle est la différence entre ces parseurs ? Quel est le plus rapide

- Si j'ai un gros fichier xml (en taille et profondeur de tags), quel parseur je choisis pour le lire ? Pourquoi ? si je veux avoir la valeur d'une donnée quel parseur je l'utiliser ?
- Difference entre Hashtable (all methods synchronized, early collection, pretty slow), HashMap (nothing synchronized, fast), ConcurrentHashMap (synchronized for fast throughput)
- Principe de rangement des éléments dans une hashMap
  - Détails des fonctions hashCode et equals
  - Pourquoi en ré-implémenter un demande la ré-implémentation de l'autre ?
  - Comment est implémenté un hashmap? (Développement algorithmique)
  - Liste chaînée avec des listes des éléments pour chaque hashCode
  - Pour le HashMap (listes chaines pour les éléments du même hashCode, pointer vers le tête de liste pour chaque hashCode - quel est l'avantage de ça ?
- Différences entre les mots clefs final, finally et finalize
  - Final permet de signifier pour une variable que sa valeur ne pourra pas être modifiée
  - Finally est le bloc situé après un Try ou un Try/Catch dont les instructions seront toujours exécutées
  - Finalize correspond à la fonction qui est appelée par le GC avant la destruction d'une instance de la mémoire
- Quelles sont les nouveautés apportées par java 8 ?