|  |
| --- |
| Une image contenant dessin  Description générée automatiquement    RECRUTEMENT  QUESTIONNAIRE JAVA  TOUS NIVEAUX |

|  |  |
| --- | --- |
| Fichier | Questionnaire Java |
| Date | Février 2022 |
| Version/Édition | 2.0 |
| Contributeurs | Kouthair FRIDHI & Cyrille CARTOT |

TABLE DES MATIERES

[1 INTRODUCTION 2](#_Toc96692994)

[1.1 Les thèmes abordés 2](#_Toc96692995)

[1.2 Les niveaux 2](#_Toc96692996)

[2 QUESTIONNAIRE 1](#_Toc96692997)

[2.1 Connaissances théoriques & Object 1](#_Toc96692998)

[2.2 Connaissance du langage 2](#_Toc96692999)

[2.2.1 Focus sur la mémoire 5](#_Toc96693000)

[2.3 Multithreading 5](#_Toc96693001)

[2.4 Les tests 7](#_Toc96693002)

[2.5 Les outils / FW utiles 8](#_Toc96693003)

[2.5.1 Spring 8](#_Toc96693004)

[2.5.2 Maven 8](#_Toc96693005)

[2.5.3 Hibernate 9](#_Toc96693006)

[2.6 SQL 10](#_Toc96693007)

[2.7 Veille technologique 10](#_Toc96693008)

# INTRODUCTION

Ce questionnaire s’adresse à tous les développeurs Java afin de pouvoir les évaluer

Elargir le scope des questions.

## Les thèmes abordés

Les thèmes abordés sont classés en 5 catégories

* Connaissances théoriques & Objet
* Connaissances du langage
* Multithreading
* Les tests
* Les outils / FW utiles
* SQL
* Veille technologique

## Les niveaux

Ce questionnaire est à destination des développeurs quel que soit leur niveau.

Les questions posées sont classées par ordre de difficulté, ce qui permet d’évaluer le niveau du candidat dans chacun des thèmes.

Questions vertes : niveau junior

Questions jaunes : niveau senior

Questions oranges : niveau expert

# QUESTIONNAIRE

## Connaissances théoriques & Object

🡪 Citez au moins 3 design-Patterns :

* Construction : Singleton, Factory, Builder
* Adapter, Composite, Decorator, Facade, Proxy
* Comportement : Command, Interpréter, Iterator, Observer, State, Strategy

🡪 Connaissez-vous les principes Solid ?

S : Single Responsability

Une classe ne doit avoir qu’une seule responsabilité

O : Open Close Principle

Une classe doit être ouverte à l’extension, fermée à la modification

L : Liskov substitution Principle

Un classe mère peut remplacer une classe fille sans modifier le fonctionnement du code

I : Interface segregation

Si le code fait choses distinctes, il faut 2 interfaces

D : Dependency Injection

On code avec les Interfaces, pas les implémentations

🡪 Qu’est-ce le TDD ?

Test Driven Development.

🡪 API : Quelles sont les méthodes que l’on peut utiliser sur un service Restful ?

PUT, POST, DELETE, GET, PATCH

🡪 API : Quelle est la différence entre un POST et un PUT ?

POST créé un objet dans la BDD

PUT permet d’apporter des modifications à un objet déjà existant

🡪 Comment fonctionne le « Garbage Collector »

C’est un outil du Framework qui permet de nettoyer les objets du programme.

🡪 Expliquez le concept d’interface :

* Une interface est un ensemble de signatures de méthodes, qui sont par essence abstraites et publiques/constantes. Une interface est déclarée avec le mot clé « interface »**.**
* Les constantes d’une interface sont accessibles à toutes les classes implémentant l’interface.

🡪 Définir le polymorphisme, comment le met-on en place :

Déf : redéfinir une méthode dans une classe fille pour modifier le comportement

🡪 Différence entre polymorphisme et surcharge :

La surcharge : la méthode dans une même classe avec des paramètres différents

🡪 Expliquez une recherche dichotomique :

La recherche dichotomique : rechercher dans une liste triée. On va diviser par 2 la zone de recherche et on vérifie si on est au-dessus ou au-dessous. Puis on continue.

🡪 Citez les règles d’utilisation des classes abstraites :

* Une classe doit être abstraite si une de ses méthodes est abstraite
* Une classe abstraite n’est pas instanciable (on ne peut pas utiliser les constructeurs d’une classe abstraite et donc on ne peut pas créer d’objet ou d’instance de cette classe).
* Une classe qui hérite d’une classe abstraite ne devient concrète que si elle implémente toutes les méthodes abstraites de la classe dont elle hérite.
* Une méthode abstraite n’est pas implémentée, mais doit être implémentée dans les sous-classes non abstraites.
* Une classe abstraite peut contenir des méthodes non abstraites et des déclarations d’attributs ordinaires.

🡪 Qu’est-ce que le BDD ?

Behavior Driven Development.

Développement (test) par le comportement.

🡪 Quel est le langage utilisé en BDD ?

Gherkin (principalement utilisé dans Cucumber pour java)

## Connaissance du langage

🡪 Citez quelques méthodes de la classe Object ?

* toString(), getClass(), equals(), hashcode(),
* clone(), wait(), wait(time), notify(), notifyAll(), finalize()

🡪 Quel est le lien entre hashcode et equals ?

Deux objets égaux doivent avoir le même hashcode, deux objets avec le même

hashcode ne sont pas forcément égaux

🡪 Qu’est-ce que l’on doit utiliser lorsque l’on veut avoir le plus de précisions possibles à la place d’un double :

BigDecimal (raison : les types primitifs sont limités en nombre de bytes)

🡪 Citez les différents types de collection :

List, Set, Map

🡪 Comment terminer un Stream pour qu’il retourne une collection ?

* collect(Collectors.tomap(cle,valeur))
* toList() à partir de V16

🡪 Comment terminer un Stream pour qu’il retourne une map ?

collect(Collectors.toMap()) ou collect(Collectors.groupingBy(cle,Collectors.counting))

🡪 Sur quelle liste est basée l’implémentation de la hashmap ?

LinkedList

🡪 La méthode get(int) est plus rapide sur un arrayList, sur une linkedList ou c’est la même chose ? Pourquoi ?

ArrayList. Car les linkedList ont des références sur les index précédents et suivants. Donc c’est plus gourmand pour une recherche.

🡪 La méthode add(int, Object) est plus rapide sur un arrayList, sur une linkedList ou c’est la même chose ? Pourquoi ?

LinkedList. Car il connait la référence de l’élément suivant et précédent.

🡪 Qu’est-ce qu’une expression Lambda ?

C’est une instance de classe anonyme qui implémente une interface fonctionnelle

🡪 Qu’est-ce qu’un Stream ?

Un ensemble d’opérations appliqué à une source de données que l’on peut classer en 2 catégories

Intermediate : opérations qui ne s’exécutent pas sans l’opération terminale (instructions)

Terminal : opération qui lance le Stream (exécution)

🡪 Quelle est la différence entre Checked/unchecked exception ?

La même diff qu’entre Exception et RuntimeException.

Exception : Doit être déclaré dans le code soit avec un throw ou un try catch

Runtime pas besoin

🡪 Quelle est la différence Error/Exception ?

Exception dans le code

Error dans la JVM

🡪 Qu’est-ce qu’une interface fonctionnelle ?

C’est une interface avec une seule méthode.

🡪 A quoi sert l'opération flatMap du Stream, quelle est la différence avec Map ?

flatMap sert a mettre tout au même niveau.

Combiner une liste de List en une List.

Exemple Stream<List<String>> en Stream<String>

🡪 Que signifie sealed Interface / Class ? (V15)

On ne peut implémenter une interface Sealed que par des classes identifiées (permits)

Ex : public sealed interface Service permits Car, Truck

On ne peut implémenter une classe abstraite Sealed que par des classes identifiées (permits)

Ex : public abstract sealed class Vehicle permits Car, Truck

### Focus sur la mémoire

🡪 Citez les 2 types de mémoire en Java :

Stack, Heap

🡪 Dans quel type de mémoire passe le « Garbage Collector »

Dans la Heap

🡪 Comment analyser une fuite mémoire :

heapDump, memoryDump ou outil de profiling JvisuelVM/java mission control

et d’autres…

## Multithreading

🡪 Définition et différences entre un processus et un thread.

Un ou plusieurs threads sont contenus dans un processus.

🡪 Comment créer un thread ?

class extends Thread ou implement runnable

🡪 Comment démarrer un thread ?

Thread.Start()

🡪 A quoi sert le mot clef LOCK ?

Permet d’obtenir le verrou sur une variable.

Quand on veut mettre un bout de code thread Safe, il ne pourra pas être accédé par plusieurs threads à la fois. Tous les autres threads attendent.

🡪 Comment arrêter un thread ?

On utilise la méthode inerrupt. Pas stop ( deprecated because unsafe )

🡪 Comment créer un threadpool ?

Avec Executor pour un pool de thread avec n thread

Executors.newFixedThreadPoolExecutor(X)

🡪 Deux grands problèmes rencontrés quand on fait du multithreading ?

* DeadLocks
* Complexité du code.

🡪 Comment fait-on pour attendre la fin d’un thread ?

Join()

🡪 Différence entre mutex et sémaphore :

Un Mutex contrôle l'accès à une seule ressource partagée. Il fournit des opérations pour acquérir () l'accès à cette ressource et le libérer () une fois terminé. Un sémaphore contrôle l'accès à un pool de ressources partagé.

🡪 Différence entre Task & Thread

Une tache contient un thread

🡪 Quelle est la différence entre parallèle et concurrent ?

Parallèle : une tache découpée en multitâche. Exécutent en même temps

Concurrent : c'est gérer bcp de taches (pas de même type généralement) indépendantes avec nos moyens de calcul.

🡪 Comment combiner/enchainer deux appels asynchrone en java (un qui dépend du résultat de l'autre).

CompletableFuture || Worker

🡪 Expliquez la différence entre runAsync et supplyAsync

runAsync permet de lancer une méthode asynchrone sans attendre résultat

supplyAsync permet de lancer une méthode asynchrone en attendant une valeur retournée.

🡪 Quel est le type de retour d’une méthode Asynchrone

CompletableFuture (depuis Java 8)

🡪 Qu’est-ce qu’un TLAB

Thread Local Allocation Buffer

🡪 A quoi cela sert :

Zone mémoire alloué à chaque thread pour stocker ces objects.

## Les tests

🡪 Citez des outils de test :

Junit, Cucumber, Mockito

🡪 Citez les différents types de test :

* Test unitaire
* Test d'intégration
* Test de non-régression

🡪 Qu’est-ce qu’un test unitaire ?

Le test unitaire consiste à isoler une partie du code et à vérifier qu’il fonctionne parfaitement.

🡪 Quel est le workflow à suivre lorsque l’on fait du TDD ?

* Ecrire le test
* Le test passe en rouge
* Ecrire le code à tester
* Le test passe en vert
* Refactoring

🡪 Pourquoi utiliser des Mock ?

Simuler des composants externes aux tests.

🡪 Comment tester les exceptions avec Junit 4 ?

En utilisant des @Rule expected exception ou bien @test(ExcpectedException)

🡪 Comment tester les exceptions avec Junit 5 ?

En utilisant assertThrow(Exception,traitement)

🡪 Donner quelques exemples d'annotations Junit 4 et leur équivalent en Juint5 ?

@Before/@After en jUnit4 --> @BeforeEach/AfterEach junit5

@BeforeClass/@AfterClass en jUnit4 --> @BeforeAll/@AfterAll junit5

🡪 Comment tester un endpoint ?

mockmvc (spring test) / wiremock

🡪 Est-ce que je peux changer l'accessibilité d'une méthode pour la faire tester ?

Oui, en passant de « private » à « package private » (référence livre effective java)

## Les outils / FW utiles

### Spring

🡪 Quelle est la portée (scope) par défaut d’un bean en Spring ?

Singleton

🡪 A quoi sert Spring Boot :

C'est un framework qui a pour but de faciliter la configuration d’un projet Spring

🡪 Quels sont les avantages de SpringBoot ?

Il fournit une nouvelle façon de créer une application prête pour être déployée en environnement de production.

Permet de se concentrer sur la partie Métier.

Configuration plus facile d’un projet.

🡪 Quelle est la différence entre @service @component :

Il n’y en pas. C’est juste une facilité de lecture.

🡪 Sur quel FW se base Spring ?

IOC (inversion of control)

DI (Dependency Injection)

🡪 Qu’est-ce que l’AOP :

La programmation orienté aspect permet d’intercepter les entrées ou sortie d’une méthode (par exemple pour rajouter de la sécurité)

### Maven

🡪 Qu’est-ce que Maven ?

Un outil de gestion et d'automatisation de production des projets logiciels (comme la gestion des dépendances)

🡪 Donnez une autre alternative à Maven

Gradle

🡪 Comment sauter les tests unitaires au moment du build

DskipTests. Changer dans Goal

🡪 Quelle est la différence entre dependencies et dependencyManagement

Les artifact dans dependacy seront toujours inclus dans les modules fils

Les artifact dans dependacyManagement seront inclus dans les modules fils si on les spécifie dans leur POm (avantage pas besoin de rajouter version et scope)

🡪 Comment faire pour utiliser deux dépendances sur le même module une pour le test et l’autre pour le run (exemple base H2 pour le test et une base postgres pour le run)

Utiliser le profil ou le scope

### Hibernate

🡪 Qu’est-ce que Hibernate

ORM : Object Relational Mapping.

Permet donc de représenter une base de données en objets Java et vice versa.

🡪 Donner quelques annotations utilisées au niveau des entité

@Id,@column,@ManyToOne,@ManyToMany

🡪 Combien y a-t-il de niveau de cache ?

2

🡪 Quelle est la différence entre les deux niveaux ?

Le premier lié à la session et est arrêté avec la fin de la session, le 2eme est partagé par toutes les sessions

🡪 Avec Hibernate et pour une association @OneToMany, mieux vaut utiliser une Set ou List ?

Si pas de doublons une Set pour des meilleurs performances sinon une liste

## SQL

🡪 Différence entre jointure interne externe :

inner join : intersection entre A et B

left ou right join : A left join B = tout A plus intersection

🡪 Quel est l'équivalent d'une close Where avec une fonction d'agrégation ?

Having

🡪 Différence entre une vue et une requête :

Il n’y en a pas. Facilité d'écriture

🡪 Différence entre delete et truncate :

delete : DML ( manipulation de données sans destruction de la table )

trucate : DDL (manipulation des structures avec destruction de la table )

🡪 Vous savez ce que c'est dictinct ?

Affiche la liste des valeurs « distinctes » sans doublons.

🡪 Est-ce que c'est bien de l'utiliser ?

Non. C'est très couteux (à utiliser uniquement en investigation et surtout pas dans le code)

🡪 Quels sont les avantages / Inconvénients des curseurs ?

* Avantages : possibilité de faire du traitement ligne à ligne
* Inconvénient : lent car utilisation de storage temp, et selon la volumétrie impliquée dans le curseur

Les curseurs peuvent être optimisés ou remplacés par des tables tempo / requêtes imbriquées / jointures / vues matérialisées

## Veille technologique

🡪 Quelle est la dernière version de Java :

17 depuis Sept 2021, java 18 mars 2022

🡪 Qu’est-ce que G1 ?

Le garbage Collector Version Java 11

🡪 Qu’est-ce que Kafka ?

Pour streamer des messages ou comme bdd de messages

🡪 Citez des nouveautés de Java 17 depuis Java 11 ?

* Les interfaces et classes sealed
* Le garbage collector
* Les switch et les ternaires

🡪 Qu’est-ce que ZGC ?

Le garbage Collector Version Java 15