

# QCM

1 : Cette écriture est-elle valide : `if (obj == null)`

- a) Vrai
- b) Faux

2 : Comment implémenter une valeur Optional ?

- a) `valeur.optional(of)`
- b) `optional().valeur`
- c) `optional(valeur).of`
- d) `optional.of(valeur)`

3 : Quel est l'un des principes du pattern Null Object

- a) Encapsulation des détails de création de l'objet
- b) Substitution par objet neutre
- c) Utilisation de l'héritage multiple pour implémenter la nullité
- d) Séparation des responsabilités en sous-objets spécialisés

4 : Quels sont les limites du pattern Null Object

- a) Faible performance & Manque de flexibilité
- b) Maintenance complexe & Couplage élevé
- c) Complexité structurelle & Utilité limitée
- d) Difficulté de test & Incompatibilité avec l'héritage

5 : Quel est le lien avec le pattern State

- a) Définir un état neutre/par défaut
- b) Permettre des transitions d'états multiples simultanément
- c) Simplifier la création de nouveaux états en temps réel
- d) Éliminer complètement la gestion des états

6 : Quel est le lien avec le pattern Strategy

- a) Gérer dynamiquement les valeurs Null avec des stratégies alternatives
- b) Créer un nouvel objet implémentant l'interface de valeur Null
- c) Définir une stratégie par défaut en cas de Null détectée
- d) Ignorer les valeurs Null sans déclencher d'exception

7 : Dans le live coding : Que renvoyait la classe NullNotification

- a) Une objet String vide : ""
- b) La méthode `envoie(String notif)` de l'interface Notification
- c) Une exception `NullPointerException`
- d) Rien du tout