

1- Qu'est ce que c'est ?

- Entité qui ne change pas, qui reste constant
- C'est pas un design pattern

2- Intérêt de l'immuabilité

- thread-safe : Mise en cache des entités pour une réutilisation rapide
- Aucun constructeur redéfinies par copies
- Mise en cache : ils peuvent être utilisés en mémoire, réduisant les besoins de copie.
- clés fiables pour Map et Set : les objets immuables sont stables et garantissent une récupération fiable dans les collections.

3 - Entité immuable

- Création avec final
- Est final, donc ne peut pas être changé.
- Utiliser le constructeur pour initialiser les attributs
- Ne pas exposer de méthodes de modifications

4 - Collections immuables

- Fournie un contenu immuable par défaut
- Vues non modifiables
- Collections immuables, avec list.of et set.of
- vavr

5 - Records

- Définition
- Utilisation
- Implémentation d'interface
- Limites

6 - Conclusion :

L'immutabilité permet de sécuriser les objets et les données partagées, notamment dans les environnements multi-thread, tout rendant le code plus simple et maintenable grâce aux optimisations comme le caching et l'absence de duplication d'objets.