
Formation :
Les bonnes pratiques Java

Formateur

Mr. Lakhdhar Khalil

Directeur de Protech-IT

Contact: khalillakhdhar@protech-it.org

- Présentation du formateur
- Le plan de formation
- Objectifs de la formation
- Public concerné
- Connaissances requises
- Documents utiles

Lakhdhar Khalil

- Enseignant à université Suptech
- Développeur sénior et fondateur de protech-it/ATC depuis 2013
- Fondateur de l'association formateurs sans frontières

1. Gestion de mémoire garbage collector
2. La classe Object et quelques interfaces de base

- Pratiquer le Java
- Maîtrise de base de java
- Le codage de base
- La gestion de mémoire en java
- Introduire les cadres en java et java EE

- Développeurs
- Assistant
- Full stack développeurs

- Connaissances de base en informatique et algorithmique.

C'est quoi Garbage collector?

En java, garbage signifie objets non référencés.

Garbage Collection est un processus de récupération automatique de la mémoire inutilisée du runtime. En d'autres termes, c'est un moyen de détruire les objets inutilisés.

Pour ce faire, nous utilisons la fonction `free ()` en langage C et `delete ()` en C++. Mais, en java, il est effectué automatiquement. Ainsi, java offre une meilleure gestion de la mémoire.

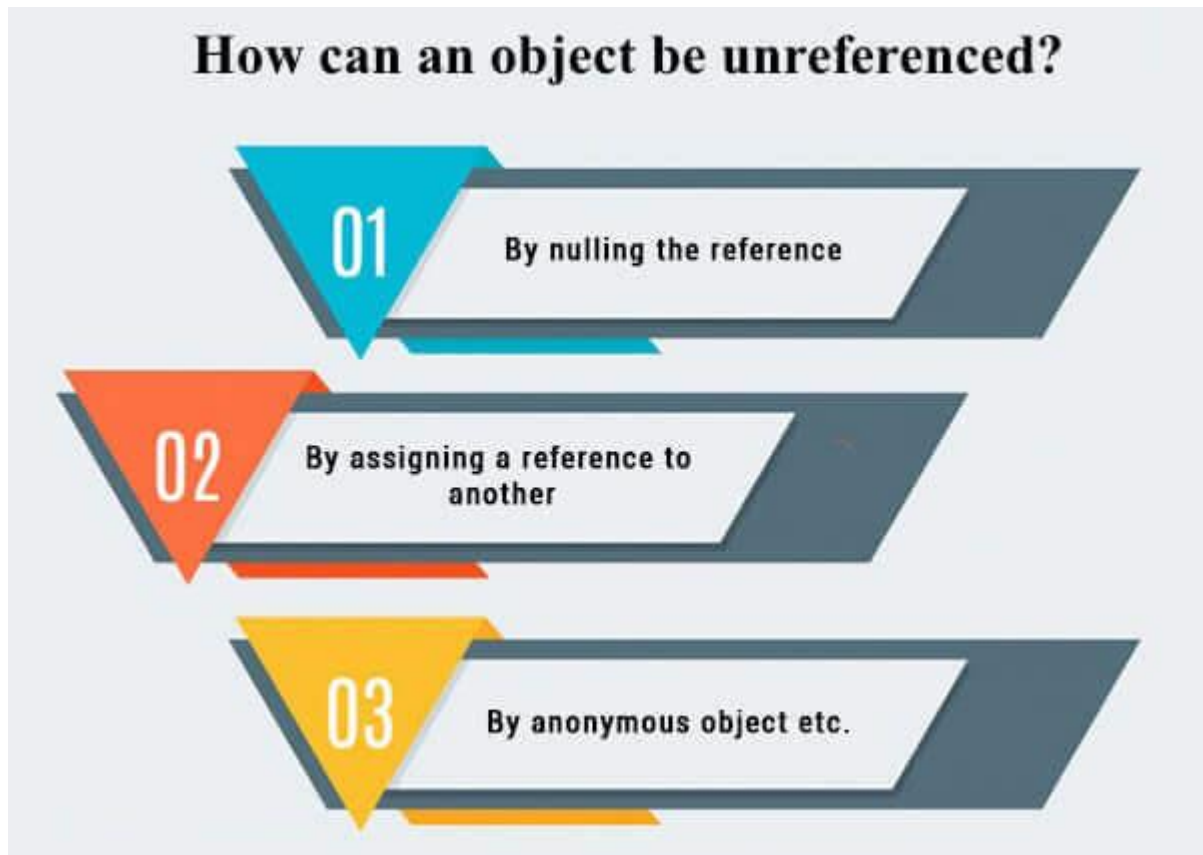
C'est quoi Garbage collector?

Avantage de la collecte des ordures

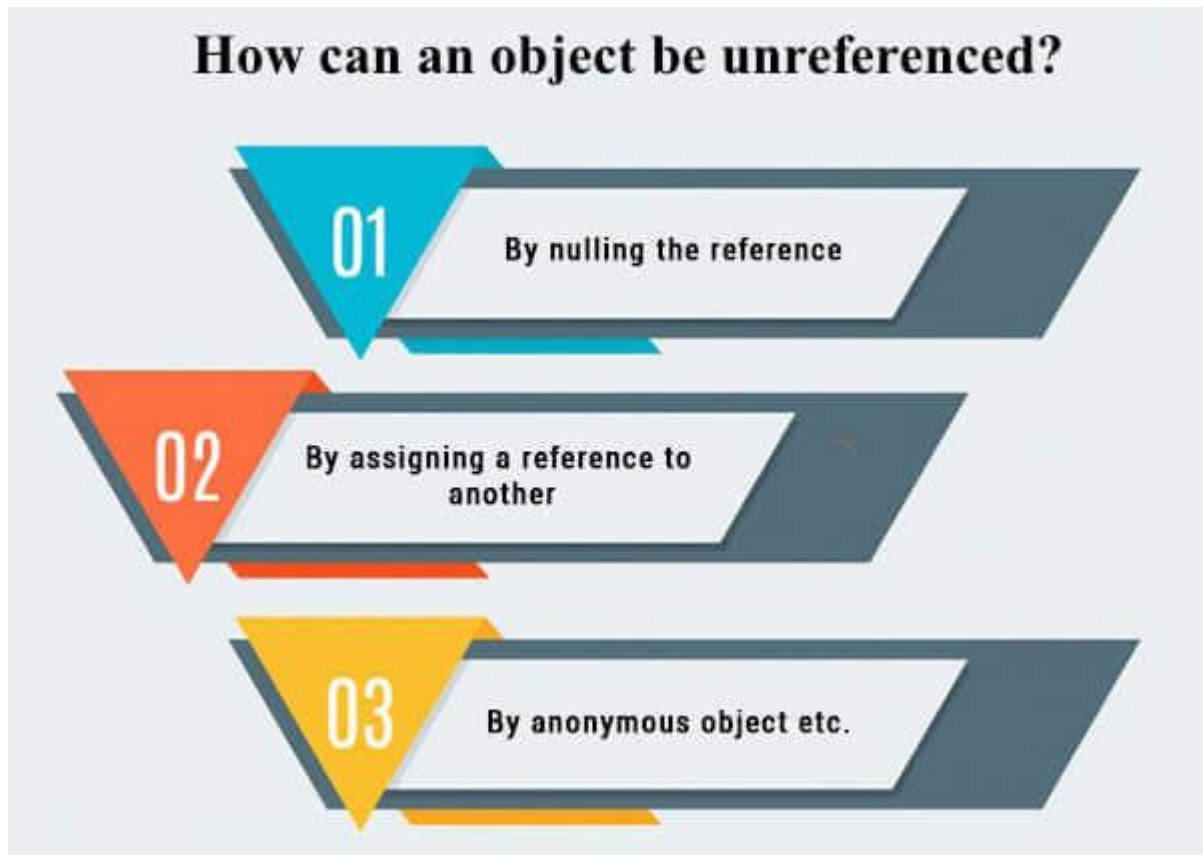
Cela rend la **mémoire** Java **efficace** car le garbage collector supprime les objets non référencés de la mémoire du tas.

Cela est **automatiquement fait** par le garbage collector (une partie de JVM), nous n'avons donc pas besoin de faire des efforts supplémentaires.

C'est quoi Garbage collector?



C'est quoi Garbage collector?



1. usage

En annulant une référence:

```
Employé e = new employé ();
```

```
e = null ;
```

2) En attribuant une référence à une autre:

```
Employé e1 = new employé ();
```

```
Employé e2 = new employé ();
```

```
e1 = e2; // maintenant le premier objet référencé par e1 est disponible pour le garbage collection
```

3) Par objet anonyme:

```
new employé ();
```

1. Etude approfondie de tous les mots-clés

Voir le cours: rappel java

Pour:

1. Abstraction
2. Try catch
3. Finally
4. Les classe loaders

1. Etude approfondie de tous les mots-clés

Démonstration

Créer une classe java `health` permettant de calculer le IMC d'un être humain avec $IMC = \text{poid}(\text{kg}) / \text{taille}^2 (\text{m})$

1. Etude approfondie de tous les mots-clés

Exercice 1:

Créer une classe java ayant les attributs suivants:

- Nom, prenom, age (static) avec une valeur par défaut 18
- Augmentez l'âge à travers une méthode grandir
- Créer une classe test pour cette application

1. Etude approfondie de tous les mots-clés

Démonstration

Créer Deux classe PC et laptop tel que laptop hérite de PC

1. Etude approfondie de tous les mots-clés

Exercices:

Créer deux classes de votre choix tel que A hérite de B

1. Etude approfondie de tous les mots-clés

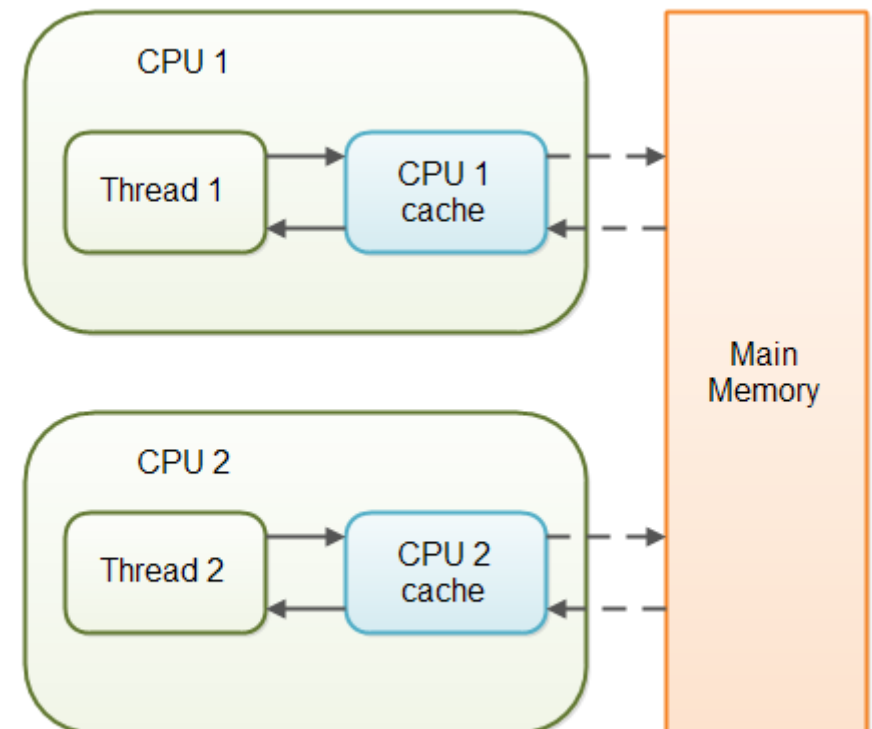
Le volatile:

mot clé Java est utilisé pour marquer une variable Java comme "stockée dans la mémoire principale". Plus précisément, cela signifie que chaque lecture d'une variable volatile sera lue à partir de la mémoire principale de l'ordinateur, et non à partir du cache du processeur, et que chaque écriture dans une variable volatile sera écrite dans la mémoire principale, et pas seulement dans le cache du processeur. .

1. Etude approfondie de tous les mots-clés

Le volatile:

volatile mot clé Java garantit la visibilité des modifications apportées aux variables dans les threads. Cela peut sembler un peu abstrait, alors laissez-moi élaborer. Dans une application multithread où les threads fonctionnent sur des variables non volatiles, chaque thread peut copier des variables de la mémoire principale dans un cache CPU tout en travaillant dessus, pour des raisons de performances. Si votre ordinateur contient plus d'un processeur, chaque thread peut s'exécuter sur un processeur différent. Cela signifie que chaque thread peut copier les variables dans le cache CPU de différents CPU. Ceci est illustré ici:



1. Etude approfondie de tous les mots-clés

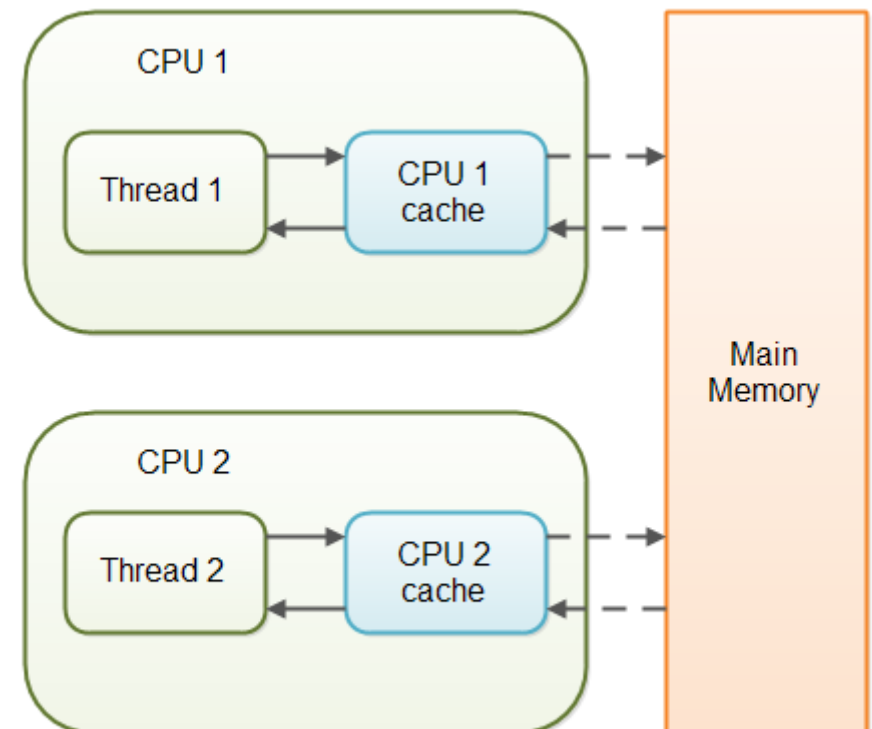
Le bloc d'initialisation en Java

mot clé Java est utilisé pour marquer une variable Java comme "stockée dans la mémoire principale". Plus précisément, cela signifie que chaque lecture d'une variable volatile sera lue à partir de la mémoire principale de l'ordinateur, et non à partir du cache du processeur, et que chaque écriture dans une variable volatile sera écrite dans la mémoire principale, et pas seulement dans le cache du processeur. .

1. Etude approfondie de tous les mots-clés

Le volatile:

volatile mot clé Java garantit la visibilité des modifications apportées aux variables dans les threads. Cela peut sembler un peu abstrait, alors laissez-moi élaborer. Dans une application multithread où les threads fonctionnent sur des variables non volatiles, chaque thread peut copier des variables de la mémoire principale dans un cache CPU tout en travaillant dessus, pour des raisons de performances. Si votre ordinateur contient plus d'un processeur, chaque thread peut s'exécuter sur un processeur différent. Cela signifie que chaque thread peut copier les variables dans le cache CPU de différents CPU. Ceci est illustré ici:



1. Etude approfondie de tous les mots-clés

L'abstraction

Pourquoi et quand utiliser des classes et des méthodes abstraites?
Pour assurer la sécurité - masquez certains détails et n'affichez que les détails importants d'un objet.

Remarque: L'abstraction peut également être réalisée avec des interfaces , sur lesquelles vous en apprendrez plus dans la séance suivante

1. Exercice

Enoncé

Créer un projet java avec 2 classes employée et congé tel que:

Un employé possède: nom,prénom,age,salaire et grade

Congé:debut, fin et durée

N.B il faut lire tout les attributs sous format String et les convertir en tenant en compte la gestion des exceptions pour employée

La durée doit être paire et dans l'intervalle 1-30 pour le congé