Simplifiez le développement d'applications Java avec Spring

15 heures Moyenne

Mis à jour le 15/12/2020

Mettre en œuvre une inversion de contrôle

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées

- Respecter les bonnes pratiques de développement en vigueur
- Mettre en place une injection de dépendances

Question 1

Dans l'acronyme SOLID, qu'est-ce que le principe de responsabilité unique signifie ?

- ○ Dans l'application, chaque responsabilité ou fonctionnalité est assurée par une seule classe.
- ✓ ◉ Une classe ne devrait avoir qu'une seule responsabilité.

Selon Robert C. Martin, « A class should have only one reason to change ».
Une classe ne devrait avoir qu'une seule raison d'être modifiée.

Question 2

L'inversion de contrôle est un des principes SOLID.

- ○ Vrai
- ✓ ◉ Faux

L'inversion de contrôle est un moyen parmi d'autres de respecter les principes SOLID. Cependant, mettre en place une inversion de contrôle ne garantit pas que l'application respecte les principes SOLID !

Question 3

Cochez les affirmations correctes parmi les propositions suivantes.

Attention, plusieurs réponses sont possibles.

- ✓ ☑ L'**inversion de dépendances** est un des 5 principes **SOLID**.
- ☐ L'**inversion de dépendances** permet d'injecter les dépendances dans les classes qui en ont besoin.
- ✗ ☐ L'**inversion de dépendances** repose sur le concept d'**abstraction**.
- ☐ L'utilisation d'interfaces implique de faire de l'**inversion de dépendances.**
- ✓ ☑ Pour respecter le principe d'**inversion de dépendances**, il est possible d'utiliser des interfaces.

Question 4

Afin de respecter les principes SOLID, les modules de haut niveau et de bas niveau doivent dépendre d'abstraction.

- ✓ ◉ Vrai
- ○ Faux

Selon le principe d'inversion des dépendances (le D de SOLID) :

- *Les modules de haut niveau ne doivent pas dépendre des modules de plus bas niveau. Les deux doivent dépendre d'abstractions.*
- *Les abstractions ne doivent pas dépendre des détails. Les détails doivent dépendre des abstractions.*

Question 5

Le design pattern *Factory* va-t-il à l'encontre des principes *SOLID* ?

- ○ Oui
- ✓ ◉ Non

Le design pattern Factory peut contribuer à respecter le principe de responsabilité unique (le S de SOLID) en participant à la mise en place d'une inversion de contrôle et en délégant la responsabilité de l'instanciation à une classe dédiée.

Question 6

L'injection de dépendances est-elle une forme d'inversion de contrôle ?

- ✓ ◉ Oui
- ○ Non

Question 7

Afin de configurer et réaliser l'injection de dépendances, Spring utilise un « IoC container ». L'approche est alors plutôt déclarative ou plutôt séquentielle ?

- ✓ ◉ Approche déclarative
- ○ Approche séquentielle

L'IoC container de Spring permet d'avoir une approche plutôt déclarative où chaque élément participant à l'injection de dépendances y est déclaré ainsi que ses besoins en dépendances.

La classe d'injection de dépendances que j'ai créée de toutes pièces en début de chapitre 4 adopte une approche séquentielle. Les objets sont instanciés et injectés dans un ordre défini par l'enchaînement des appels implémentés dans cette classe.

MIEUX UTILISER SPRING

IMPLÉMENTER DES DAO

Le professeur

Loïc Guibert

Architecte logiciel et développeur Java EE freelance. --- Auteur pour OpenClassrooms.
Master informatique, spécialité Génie Logiciel