

**COMPTE RENDU :Inversion de contrôle & Injection de dépendances**

Filière :

Ingénierie informatique & réseau

(IIR)

|  |
| --- |
| **REALISE PAR : ENCADRE PAR :**  HAJIB HIBATALLAH Mme.BADRI TIJANE |

**Introduction :**

La création d'un projet informatique dans les normes peut être un processus complexe, mais voici quelques étapes générales que vous pouvez suivre pour y parvenir :

Définir les exigences : La première étape consiste à définir les exigences du projet, c'est-à-dire ce que vous voulez réaliser. Il est important d'avoir une vision claire du projet et de définir les objectifs à atteindre.

La scalabilité est la capacité d'un système à s'adapter à une charge de travail accrue ou à une augmentation du nombre d'utilisateurs. Il existe deux types de scalabilité couramment utilisés dans le domaine de l'informatique :

Scalabilité verticale : La scalabilité verticale, également appelée mise à l'échelle ascendante, consiste à augmenter la capacité d'un système en ajoutant des ressources supplémentaires à un seul serveur ou à une seule machine. Cela peut inclure l'ajout de processeurs plus puissants, de mémoire vive, de stockage ou de bande passante réseau. La scalabilité verticale est généralement plus simple à mettre en œuvre, mais elle peut être limitée par les capacités matérielles de la machine unique.

Scalabilité horizontale : La scalabilité horizontale, également appelée mise à l'échelle descendante, consiste à augmenter la capacité d'un système en ajoutant des ressources supplémentaires à plusieurs serveurs ou machines. Cela peut inclure l'ajout de serveurs supplémentaires à un cluster ou l'utilisation de technologies de conteneurisation pour créer des instances de serveurs supplémentaires. La scalabilité horizontale est généralement plus complexe à mettre en œuvre, mais elle peut offrir une capacité quasi illimitée.

Le couplage en fait référence au niveau de dépendance entre les différents composants d'un système. Voici les deux types de couplage couramment utilisés lors de la création de projets :

Couplage fort : Le couplage fort se produit lorsque deux composants sont fortement dépendants l'un de l'autre. Cela signifie qu'un changement dans l'un des composants aura un impact significatif sur l'autre. Par exemple, si un composant est modifié, il peut nécessiter la modification de nombreux autres composants qui en dépendent étroitement.

Couplage faible : Le couplage faible se produit lorsque deux composants sont faiblement dépendants l'un de l'autre. Cela signifie qu'un changement dans l'un des composants n'aura pas d'impact significatif sur l'autre. Les composants peuvent être modifiés indépendamment les uns des autres sans nécessiter de modifications importantes dans les autres composants.

**L’inversion de contrôle :**

L'inversion de contrôle (IoC) est un concept de programmation qui consiste à inverser le flux de contrôle d'un programme en déléguant la responsabilité de l'exécution à un conteneur ou à un framework. Au lieu que le programme contrôle lui-même l'exécution de ses différentes parties, il externalise cette responsabilité à une infrastructure ou à un conteneur qui se charge de l'exécution.

En pratique, cela signifie que les dépendances entre les différents modules du programme sont gérées par le conteneur plutôt que par les modules eux-mêmes. Le conteneur crée et instancie les objets nécessaires, gère les cycles de vie des objets, injecte les dépendances, et ainsi de suite. Cela permet de réduire le couplage entre les différents modules du programme, ce qui les rend plus indépendants et plus faciles à tester et à maintenir.

En résumé, l'inversion de contrôle est un concept clé de la programmation qui permet de déléguer la gestion des dépendances entre les différents modules d'un programme à un conteneur ou à un framework. Cela permet de réduire le couplage entre les différents modules et de rendre le code plus facile à maintenir et à tester.



**Création sans utilisation de Spring :**

* Les développeurs doivent écrire tout le code à partir de zéro, sans l'aide d'un framework.
* Les développeurs doivent gérer manuellement les dépendances entre les différentes bibliothèques et modules utilisés dans le projet.
* Le code est généralement plus verbeux et peut prendre plus de temps à écrire et à maintenir.
* Les développeurs peuvent utiliser des frameworks tiers pour certaines fonctionnalités, mais doivent les intégrer manuellement dans leur projet.

**Création avec utilisation de Spring :**

* Les développeurs peuvent utiliser Spring pour gérer les dépendances entre les différentes bibliothèques et modules utilisés dans le projet, ce qui permet de gagner du temps et de simplifier la configuration.
* Spring fournit un ensemble complet de modules pour le développement d'applications web, la gestion de données, la sécurité, les services web, et bien plus encore, ce qui permet aux développeurs de se concentrer sur la logique métier plutôt que sur l'infrastructure.
* Spring fournit un conteneur d'injection de dépendances (IoC) pour gérer les dépendances entre les différents composants d'une application, ce qui permet de réduire le couplage et de faciliter les tests et la maintenance.
* Les développeurs peuvent utiliser des annotations pour simplifier le code et rendre l'application plus lisible et plus facile à comprendre.

Pour travailler avec Spring, pour faire l’injection de dépendances j’ai utilisé 2 versions : Version XML et version annotations (@Component & @Autowired) .

Version XML :

La version XML fait référence à la spécification de la version d'un document XML, qui est définie dans l'en-tête du fichier. Chaque version définit un ensemble de fonctionnalités et de règles de syntaxe qui peuvent être utilisées dans le document, il est donc important de spécifier la version correcte pour garantir que le document est interprété et traité correctement.

Version Annotations :

@Component : est une annotation de Spring Framework qui permet de déclarer une classe comme étant un composant géré par le conteneur d'inversion de contrôle (IoC) de Spring. L'annotation @Component indique que la classe est une source de bean, c'est-à-dire qu'elle peut être instanciée et utilisée dans une application Spring.

Lorsqu'une classe est annotée avec @Component, Spring la traite automatiquement comme un bean et la gère dans son contexte d'application. Le conteneur d'inversion de contrôle Spring instancie automatiquement la classe et gère son cycle de vie, en s'assurant que toutes les dépendances sont résolues et que la configuration est correctement initialisée

@Autowired : est une annotation de Spring Framework qui permet l'injection de dépendances automatique dans une classe. Elle permet d'injecter automatiquement les dépendances requises par une classe, sans avoir besoin de les instancier ou de les initialiser manuellement.

Lorsqu'une classe est annotée avec @Autowired, Spring tente automatiquement de trouver une instance correspondante de la dépendance requise dans le contexte d'application. Si une telle instance est trouvée, elle est injectée dans la classe à l'aide du mécanisme d'injection de dépendances de Spring.

**Conclusion :**

Spring est un framework de développement d'applications pour la plateforme Java. Il fournit un ensemble de fonctionnalités pour faciliter le développement d'applications d'entreprise, en favorisant une architecture basée sur des composants, l'inversion de contrôle et l'injection de dépendances.

Spring est construit autour de plusieurs modules, tels que Spring Core, Spring MVC, Spring Data, Spring Security, Spring Boot, etc., qui offrent des fonctionnalités spécifiques pour différents aspects du développement d'applications. Spring Core fournit les fonctionnalités de base du framework, notamment l'inversion de contrôle et l'injection de dépendances. Spring MVC fournit un framework de développement Web pour créer des applications Web basées sur le modèle MVC (Modèle-Vue-Contrôleur). Spring Data facilite l'intégration de bases de données dans les applications Spring, et Spring Security fournit des fonctionnalités de sécurité pour les applications Spring.

L'utilisation de Spring permet de simplifier le développement d'applications Java en offrant des fonctionnalités prêtes à l'emploi pour la gestion des dépendances, la configuration, l'inversion de contrôle et l'injection de dépendances. Elle favorise également une meilleure séparation des préoccupations et améliore la testabilité de l'application.

En résumé, Spring est un framework de développement d'applications pour la plateforme Java, qui facilite le développement d'applications d'entreprise en favorisant une architecture basée sur des composants, l'inversion de contrôle et l'injection de dépendances.