**SPRING**

2 services de base fournis par le framework SPRING :

1. Gestion du cycle de vie d'un bean
2. Injection de dépendances

SPRING est un framework de développement d’applications JAVA qui intervient dans l’intéraction entre les couches et permet de faire la gestion du cycle de vie d’un bean et l’injection de dépendances.

Utile dans la transition : Couche présentation -> service

Service -> dao

Frameworks (momentum / où) : JUnit, Hibernate

**Conteneur** : (Classe ou XML) zone contient les composants susceptibles d'être injectés et

manage leur cycle de vie.

**Injection de Dépendance** : Spring injecte les composants nécessaires à d'autres

composants qui en dépendent.

**Approche Statique**

* Déclaration du conteneur et du bean
* Définition relation entre les deux

**Approche Dynamique**

* Chargement du conteneur
* Prise en main du bean qui m’intéresse
* Utilisation du bean qui m’intéresse

**Couplage Fort** (new pour instanciation d’une classe)

* Problème lors d’une réutilisation (obligation de changer le code en dur)
* On peut difficilement tester le service

TODO le 31/03/2020

**Procédure pour l’injection de dépendance**

1. Créer projet maven quickstart
2. Ajouter dépendances spring framework
3. Créer des classes dao et service
4. Créer le fichier xml conteneur
5. Déclarer classes dao et service dans conteneur (injecter bean dao dans bean service)
6. Créer classe main
7. Dans le main, mettre en œuvre l’approche en 3 étapes
   1. Chargement conteneur
   2. récupération du bean service
   3. manipulation du bean service (ex: lecture d'un élève dont l'id est 1)

TODO le 01/04/2020

* Cycle de vie d’un bean
* Autocablage « autowire »
* Couplage faible
* Spring Data : techno qui permet de se brancher à une BDD
* Spring Boot
* Web avec Spring MVC techno qui s’appuie sur Spring Framework

2 Valeurs Ajoutées de **Spring Data**:

* Code beaucoup plus light
* Gestion des exceptions simplifiées

Adaptabilité à la source de données

Reactive Streams est une initiative visant à fournir une norme pour le traitement de flux asynchrone avec une contre-pression non bloquante.

**Mise en œuvre du couplage faible**

1. Procédure pour l’injection de dépendance
2. Création d’une interface pour la couche dao « IEleveDao »
3. Au niveau de la couche service

**Mise en oeuvre** **SPRING Data**

1. créer projet maven quicktstart
2. ajouter dépendance spring framework
3. créer interface dao (hérite de jpaRepository)
4. créer le fichier xml conteneur, avec les bean (ppc,datasource,entitymanagerfactory,transaction manager,jparepository)
5. tester la dao dans une classe test et tester create et findby
6. coder le service dans le conteneur (mettre en propriete l’interface dao + ajouter le getter et setter de l’interface)
7. déclarer le service dans le conteneur (injecter bean dao dans bean service)
8. tester le service dans une classe test et tester créerEleve et lireEleve
   1. chargement conteneur
   2. récupération du bean service
   3. manipulation du bean service (ex: lecture d'un élève dont l'id est 1)