
Java Avancé

Cours 3 : Qualité Logicielle

Arsène Lapostolet & Nada Nahle

Concepts

Polymorphisme

- Contrats de services (interfaces)
- Cacher les détails au code client

ex : List avec LinkedList/ArrayList

Inversion et injection de dépendances

Injection de dépendances : Externaliser des comportement dans des classes (dépendances), et les fournir en paramètre du constructeur.

Inversion de dépendance : Abstraire les dépendance par un contrat de service (une interface)

- Code modulaire
- Couplages moins forts

-> *Le code est plus facile à tester, maintenir, refactorer*

Conteneur d'injection de dépendance (IoC Container)

Composant qui permet de d'automatiser :

- Construction des classes avec leur dépendances
- Gestion de leur cycle de vie

Exemple de librairies :

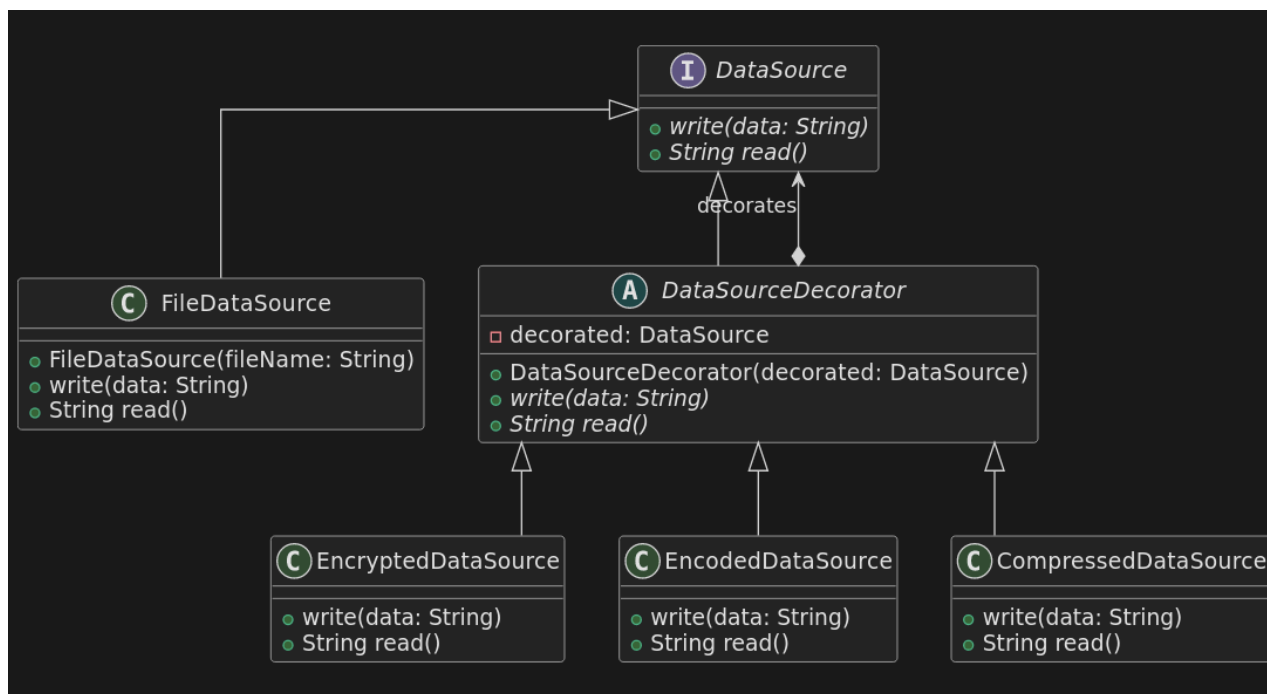
- Spring
- Context Dependency Injection (standard JakartaEE)
- Guice



Des questions ?

Patrons de conception

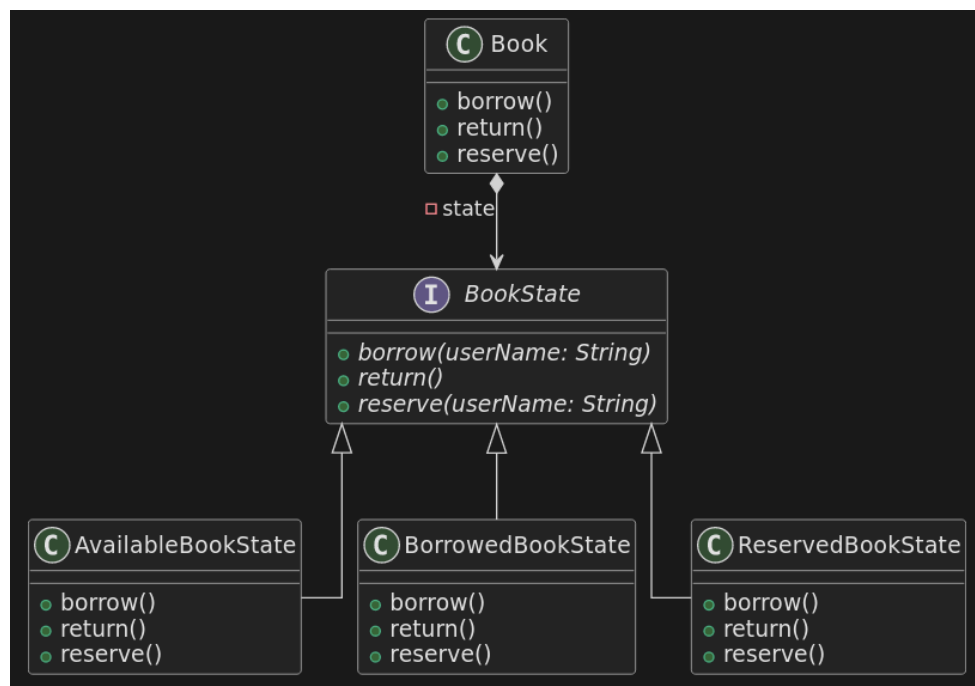
Décorateur



Fabrique

- Création d'objet complexe
- Découplage de création d'objet

Etat





Des questions ?

Ecriture du code

Nommage

- Ne pas faire d'abréviations
- Ne pas utiliser d'acronymes
- Toujours décrire l'objectif du symbole dans son nom
- Ne pas avoir peur d'avoir des noms de symboles longs
- Utiliser des noms en anglais
- Utiliser des noms prononçables
- Ne pas utiliser de nombres magiques

Fonctions / méthodes

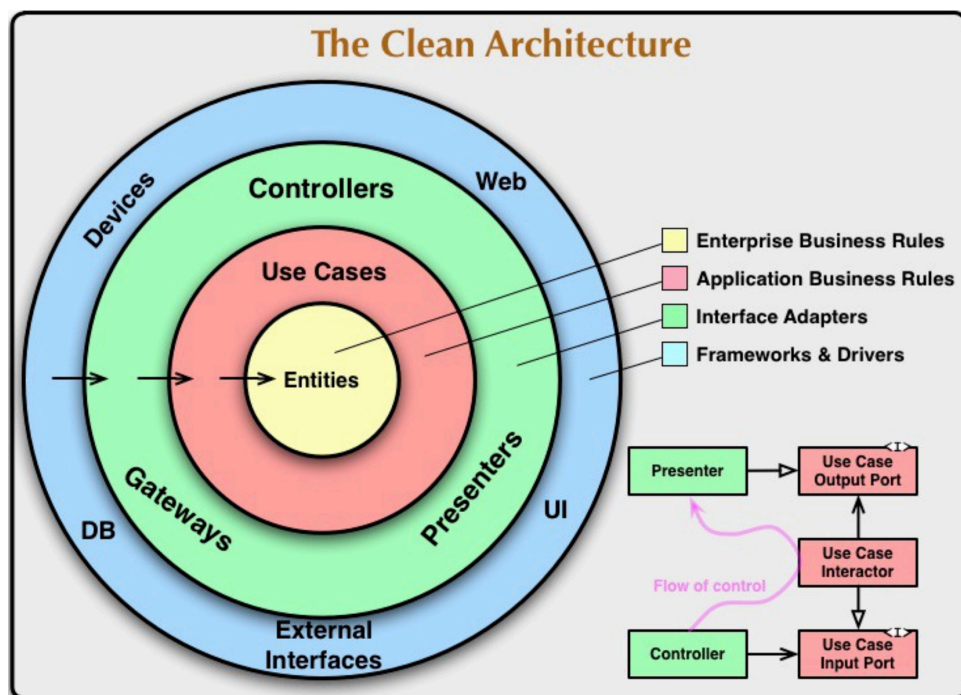
- Pas plus de 20 lignes, 10 lignes idéalement
- Doit faire une seule chose
- Séparer les Commandes et les Requêtes :
 - Requête : calculer une valeur
 - Commande : Faire une action
- Utiliser les méthodes/fonctions et leur nom pour décrire des blocs logiques
- Pas plus de 4 arguments pour méthode/fonction



Des questions ?

Architecture

Architecture en couche



- Couche métier
 - Entité
 - Objet-valeur
 - Associations
 - Service
- Couche application
- Couche interface
- Couche infrastructure



Des questions ?