

SOLIDité d'un code bien conçu

Sébastien Mosser - INF5153
Chapitre 4 - Capsule 3
Automne 2020

UQAM | Département d'informatique

credits photos: Pixabay

ace

CC BY NC SA

Dans les épisodes précédents

- Mise en oeuvre du principe de "*Information Hiding*"
 - Les données sont encapsulées,
 - les objets exposent des services (méthodes).
- Alignement entre logique d'affaire et services exposés
- Attention aux fuites de données (via *getters* et *setters*)
 - Mécanismes de copies à prendre en compte
- Trop encapsuler = ne plus pouvoir travailler
 - Mais pas assez = porte ouverte à toutes les fenêtres

Encapsulation

UQAM | Département d'informatique

Dans les épisodes précédents

- Les Types :
 - décrivent de manière *abstraite* les *services rendus* par un objet
 - On peut *affecter à une variable typé T* tout élément dont le type concret est *T* ou un de ses sous-types.
 - P-ex., `List<Card> result = new ArrayList<>();`
 - Sont mis en oeuvre par "Généralisation" ou "Réalisation".
 - Abstraction** : Iterable, Comparable, Clonable, ...
- Modularisation des interfaces :
 - Une classe doit dans la mesure du possible *implémenter plusieurs interfaces spécialisées* plutôt qu'une grosse.

Types & Interfaces

UQAM | Département d'informatique

Dans les épisodes précédents

ENCART CAMÉRA



- L'**espace d'état** modélisé par des objets peut devenir **très grand**
- On s'intéresse uniquement aux **états abstraits**
 - P.-ex. : la pile de carte est **vide, mélangée**, ...
 - **Attention**: *On ne conçoit pas jusqu'à un automate fini !*
- Définition de **relations entre objets** :
 - P.-ex., Équivalence, Égalité, ...
- Faire en sorte de **minimiser l'espace d'état**
 - Pas d'**information en double à synchroniser**
- Comment modéliser l'**absence de valeur** ?

États des objets

Principes sous-jacents : SOLID & Demeter

ENCART CAMÉRA



- **Responsabilité Unique** (*Single Responsibility*)
 - Un objet fait une et une seule chose
- **Principe Ouvert/Fermé** (*Open/Closed Principle*)
 - Une évolution du projet minimise le nombre de modifications et exploite les capacités d'extensions.
- **Ségrégation des interfaces** (*Interface Segregation*)
 - On préfère des interfaces spécialisées à des fourre-tout
- **Loi de Demeter** (*Principle of least knowledge*)
 - "On ne parle pas avec les gens qu'on ne connaît pas".

Responsabilité Unique (SOLID)



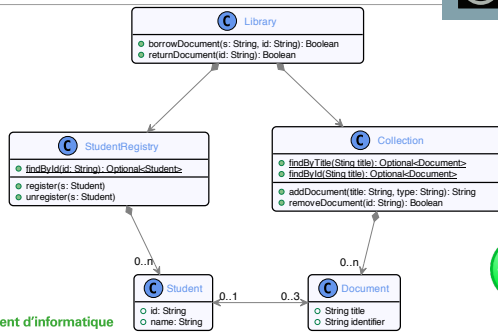
ENCART CAMÉRA



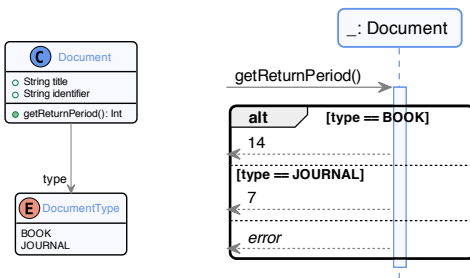
 Library

- registerStudent(s: String): Boolean
- unregisterStudent(s: String): Boolean
- addDocument(title: String, type: String): String
- removeDocument(id: String)
- borrowDocument(s: String, id: String): Boolean
- returnDocument(id: String): Boolean

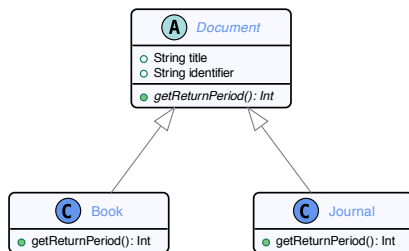
Responsabilité Unique (SOLID)



Principe Ouvert/Fermé (SOLID)



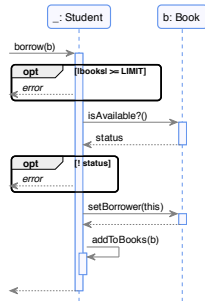
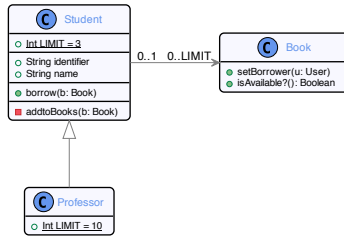
Principe Ouvert/Fermé (SOLID)



Substitution de Liskov (SOLID)

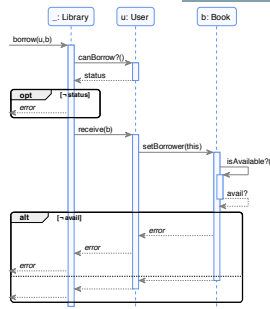
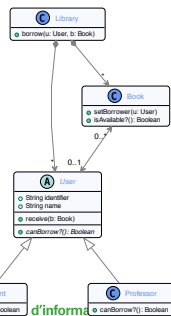


ENCART CAMÉRA



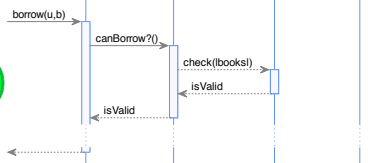
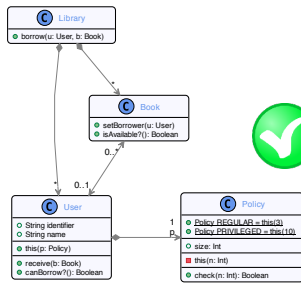
Substitution de Liskov (SOLID)

ENCART CAMÉRA



Substitution de Liskov (SOLID) - Alternative

ENCART CAMÉRA

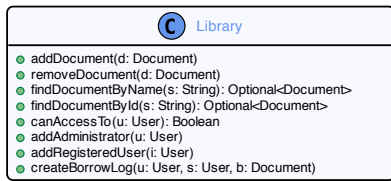


Composition over *inheritance*

Ségrégation des Interfaces (SOLID)

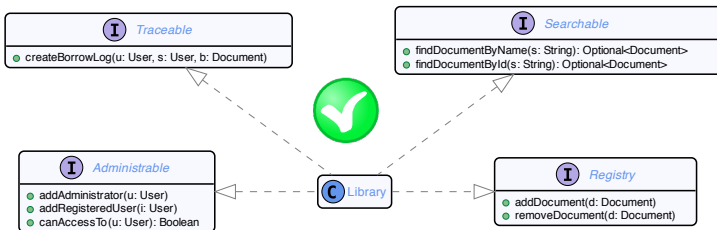


ENCART CAMÉRA



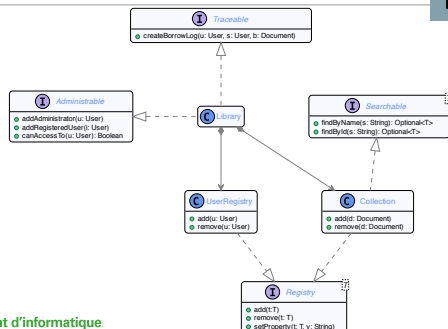
Ségrégation des Interfaces (SOLID)

ENCART CAMÉRA



Amélioration de l'assignation des responsabilités

ENCART CAMÉRA



Injection des Dépendances (SOLID)

ENCART CAMÉRA

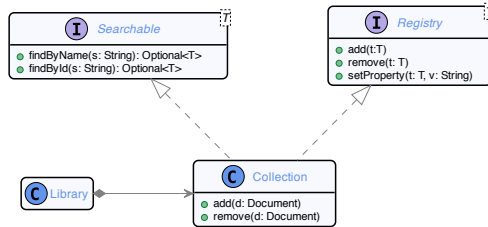


- L'**injection de dépendance** est liée à l'**Inversion de Contrôle**
 - Aussi appelé "*Principe de Hollywood*" (on vous rappellera ...)
- Un objet va **spécifier ses dépendances requises**, à travers l'interface
 - *J'ai besoin d'un objet qui sache faire X (quel qu'il soit concrètement)*
- L'objet n'est **pas responsable de la création** de sa dépendances
 - *Le "runtime" garantit qu'il fournira un objet conforme au bon moment*

Injection des Dépendances (SOLID)

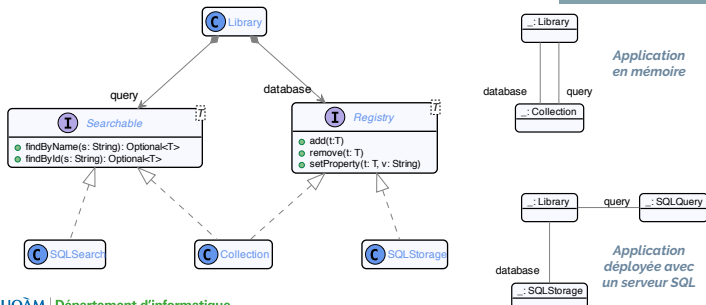


ENCART CAMÉRA



Injection des Dépendances (SOLID)

ENCART CAMÉRA

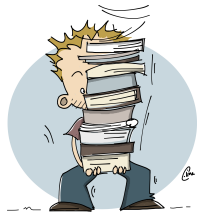


UQÀM

Département d'informatique

FACULTÉ DES SCIENCES
Université du Québec à Montréal

ENCART CAMÉRA



<https://mosser.github.io/>



<https://ace-design.github.io/>

Abonne toi à la chaine,
et met un pouce bleu !