

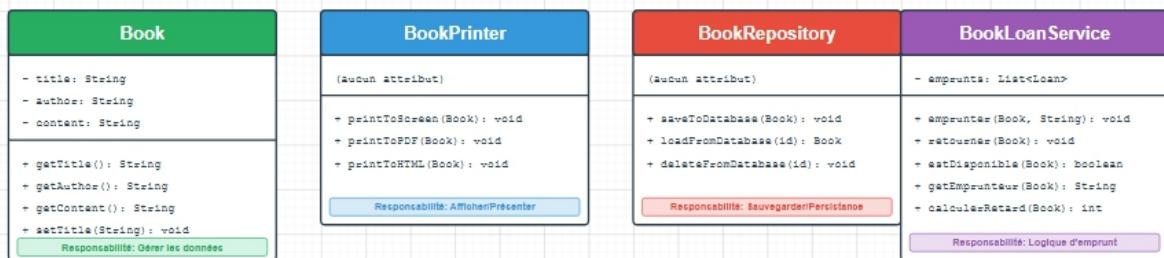


ICT 301 :shema des differentes classes du support de cour et resume sur les principes solid

1-principe srp

SRP (Single Responsibility Principle) : Une classe ne doit avoir qu'une seule raison de changer, c'est-à-dire une seule responsabilité bien définie.

Application du Principe SRP (Single Responsibility Principle)



⌚ Avantages de la Séparation des Responsabilités (SRP)

✓ Book (Données)

- Responsable uniquement des données du livre
- Facile à tester et maintenir
- Réutilisable dans tout le système

✓ BookPrinter (Affichage)

- Gère tous les formats d'affichage
- Changer l'affichage sans toucher aux données
- Ajouter de nouveaux formats facilement

✓ BookRepository (Persistence)

- Gère la sauvegarde en base de données
- Changer de BDD sans impacter le reste
- Centralise la logique de persistance

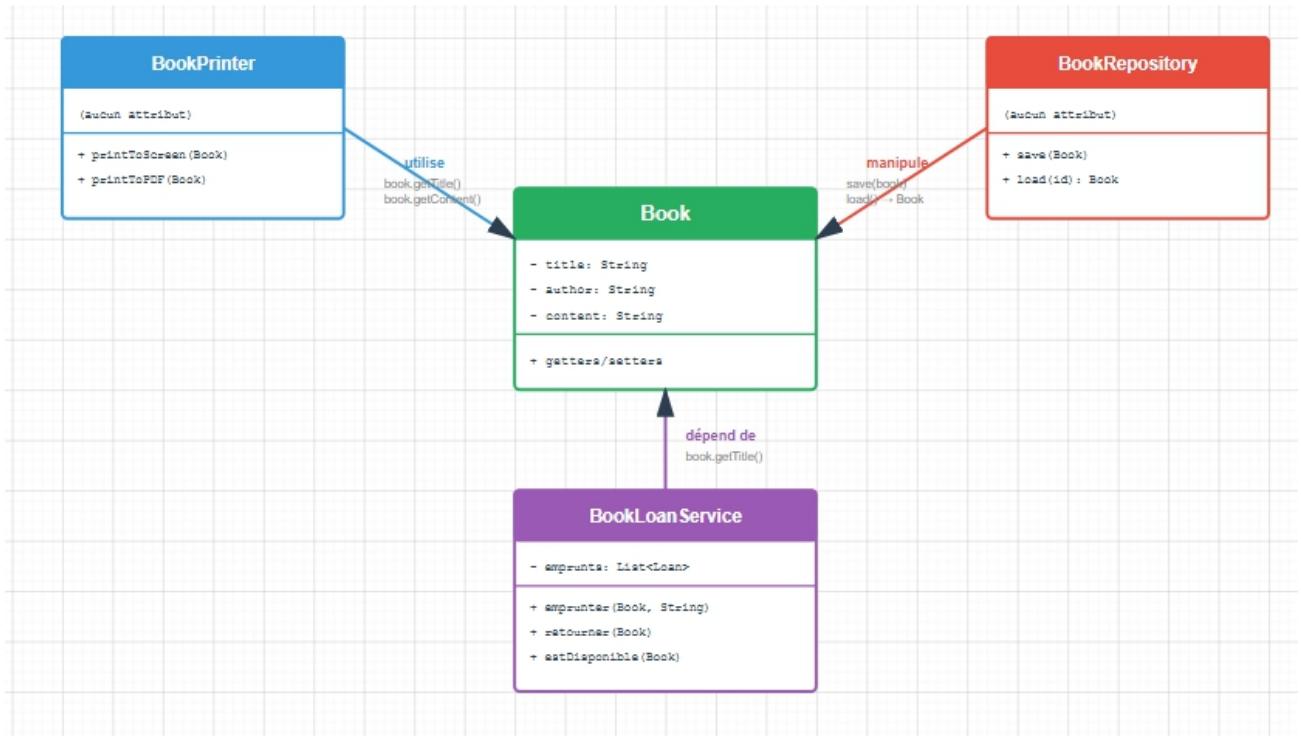
✓ BookLoan Service (Métier)

- Gère la logique d'emprunt et retour
- Evolution des règles métier indépendante
- Testable isolément

💡 Principe Clé du SRP

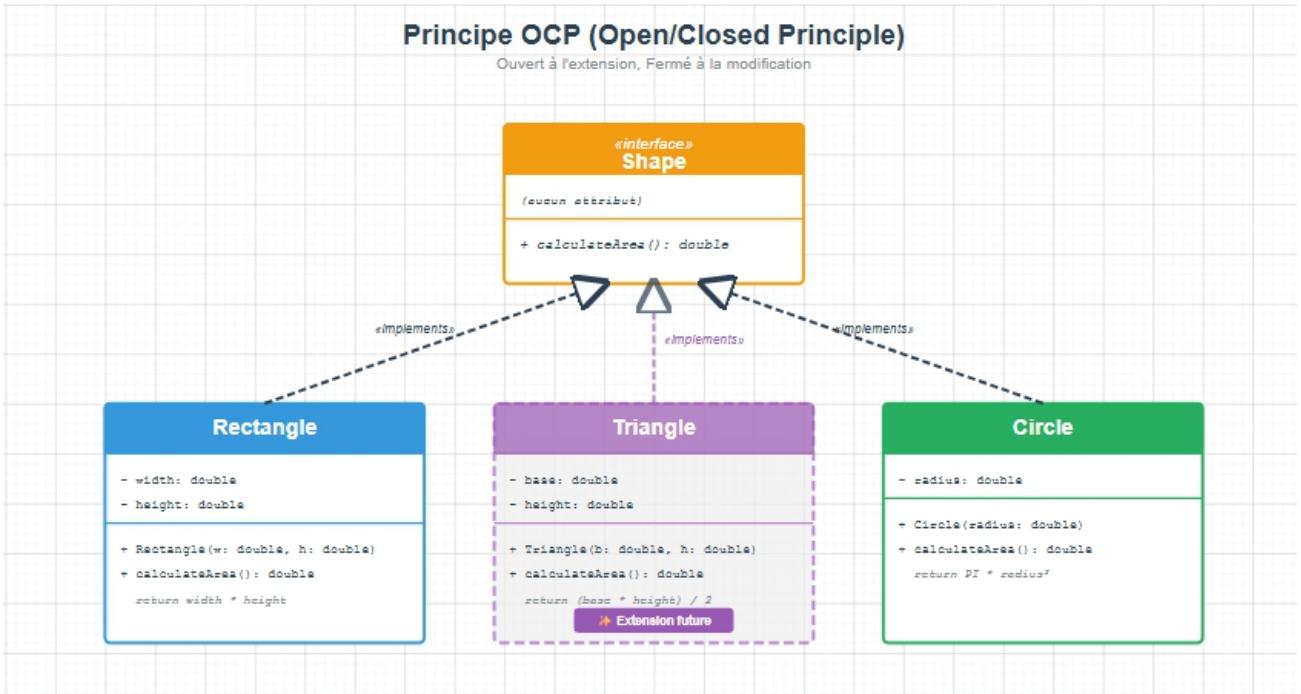
Chaque classe a UNE SEULE raison de changer = UNE SEULE responsabilité
Cela rend le code plus maintenable, testable et évolutif !

Shema des differentes classes



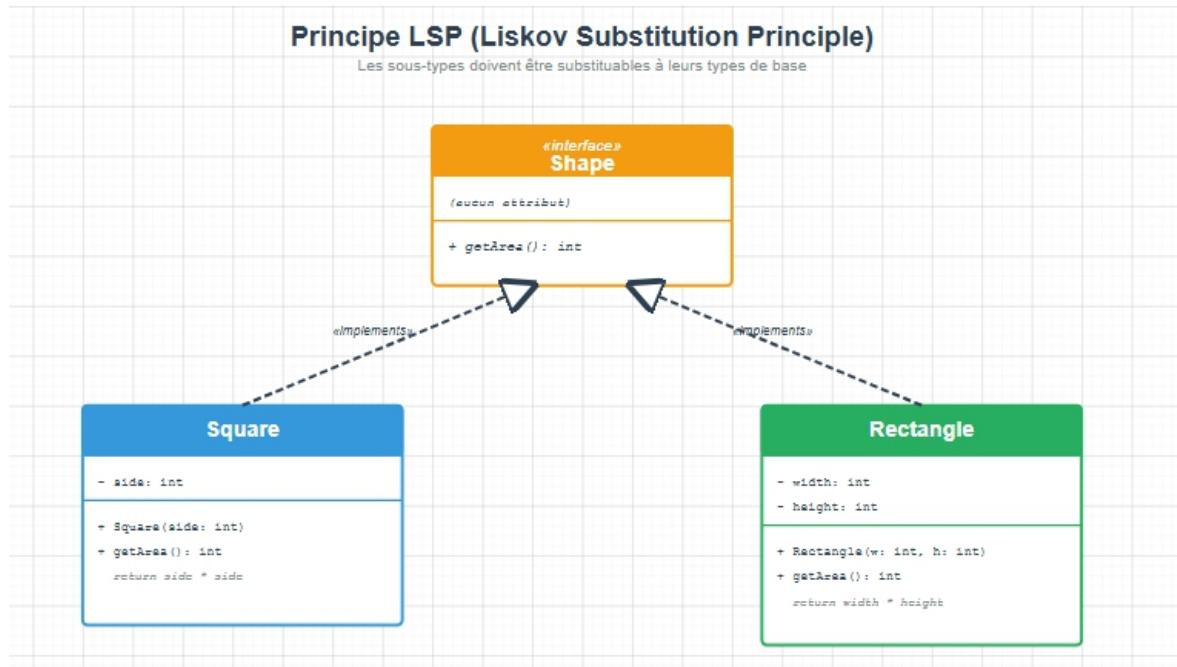
2-principe ocp

OCP (Open/Closed Principle) : Les entités logicielles doivent être ouvertes à l'extension mais fermées à la modification - on ajoute des fonctionnalités sans modifier le code existant.



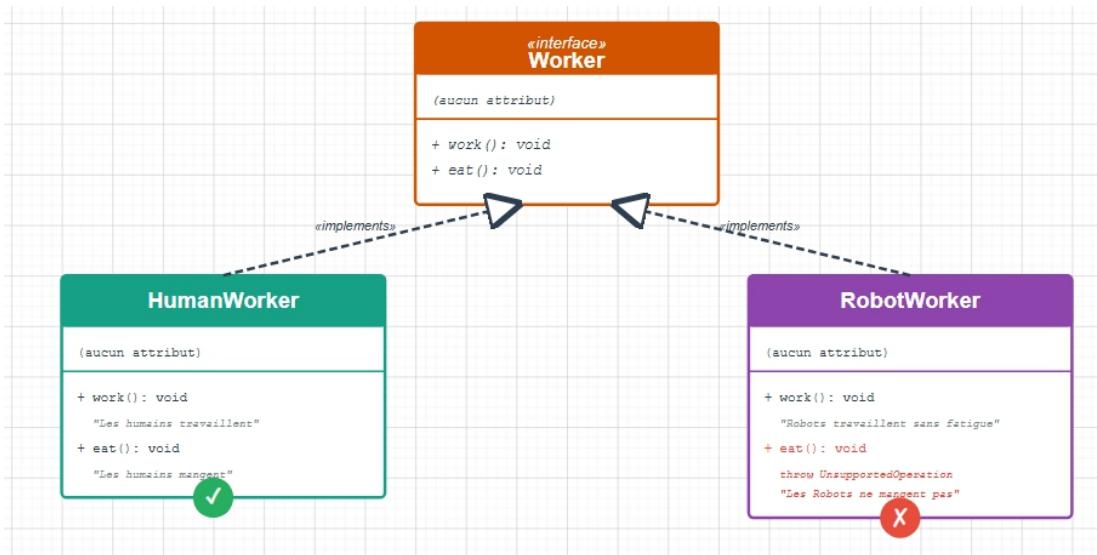
3-principe lsp

LSP (Liskov Substitution Principle) : Les objets d'une classe dérivée doivent pouvoir remplacer les objets de la classe parent sans altérer le bon fonctionnement du programme.



4-principe isp

ISP (Interface Segregation Principle) : Mieux vaut plusieurs interfaces spécifiques qu'une seule interface générale - les clients ne doivent pas dépendre de méthodes qu'ils n'utilisent pas.



5-principe dip

DIP (Dependency Inversion Principle) : Les modules de haut niveau ne doivent pas dépendre des modules de bas niveau, mais tous deux doivent dépendre d'abstractions (interfaces).

