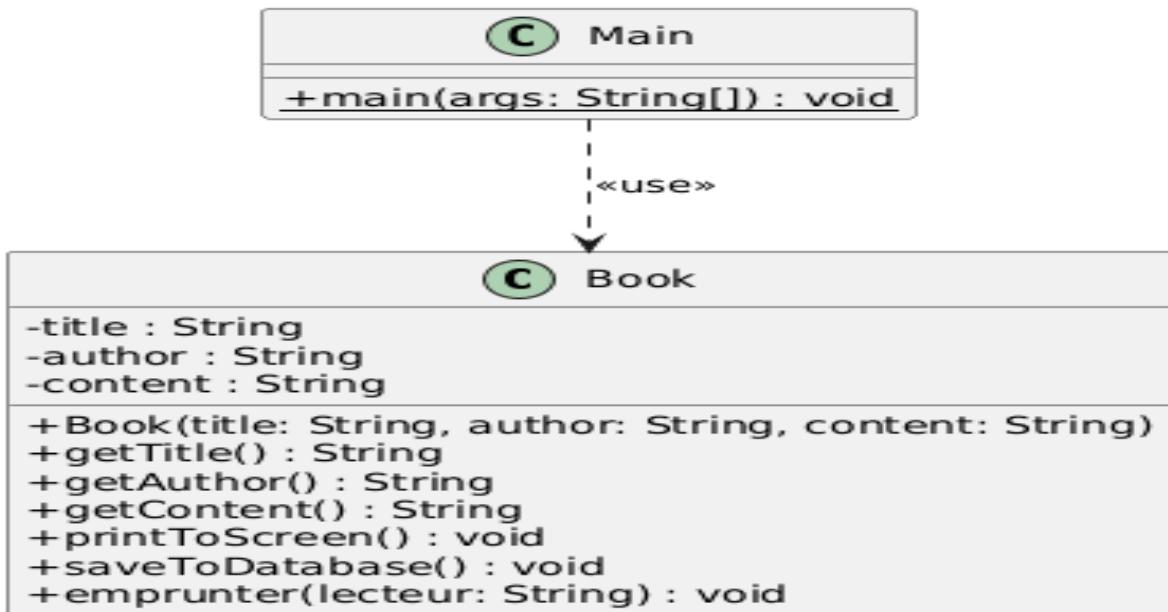


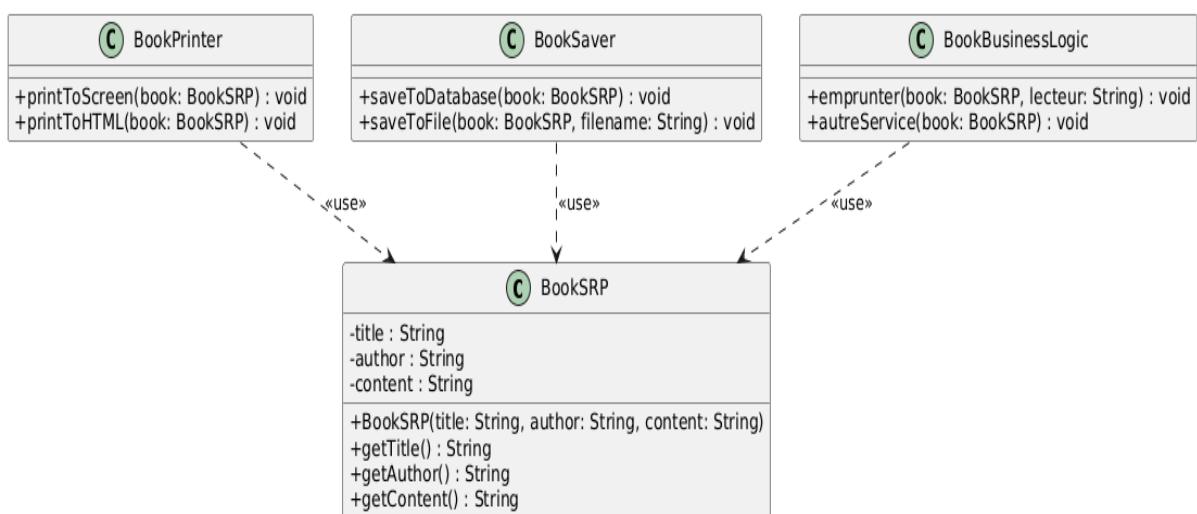
Diagrammes de classes UML pour les principes SOLID Avant et Après Refactoring

1. SRP – Single Responsibility Principle (Une classe ne doit avoir qu'une seule responsabilité)

- Avant refactoring (Violation : une classe gère trop de responsabilités)

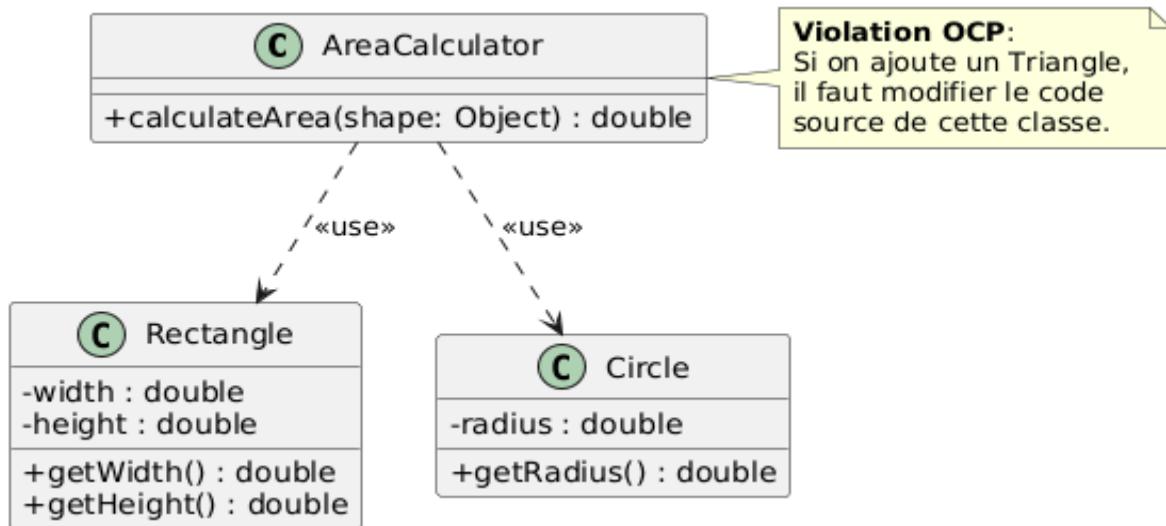


- Après refactoring (Responsabilités séparées en plusieurs classes)

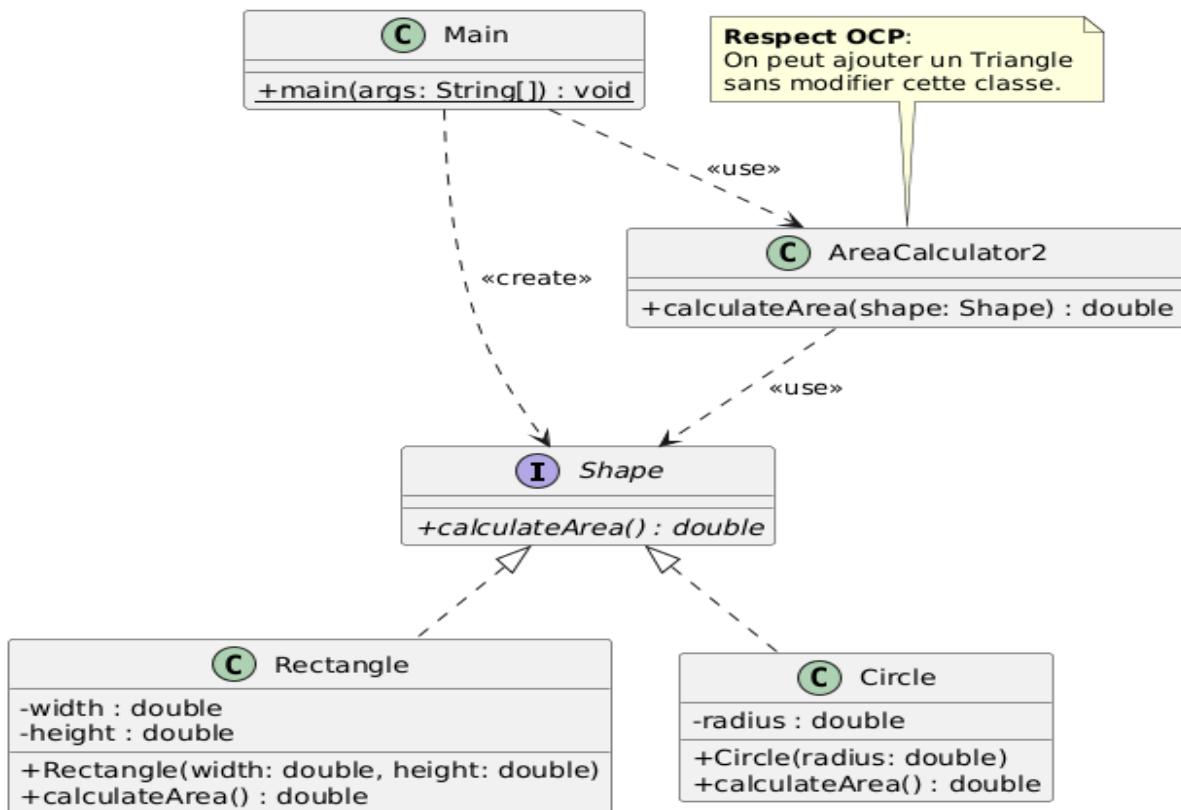


2. OCP – Open-Closed Principle (Ouvert à l'extension, fermé à la modification)

- Avant refactoring (modification nécessaire pour ajouter une nouvelle forme)



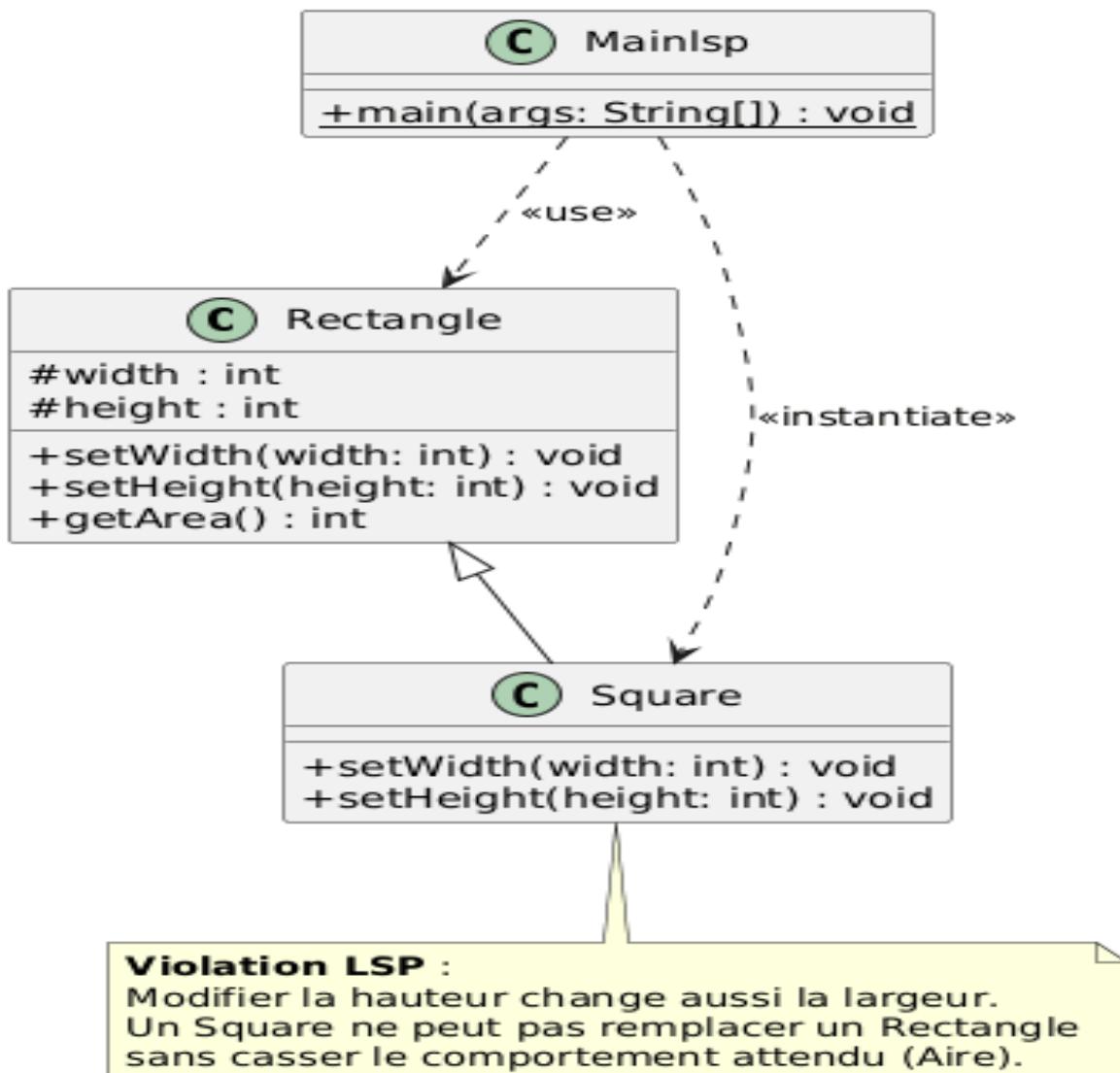
- Après refactoring (extension via interface, sans modifier le code existant)



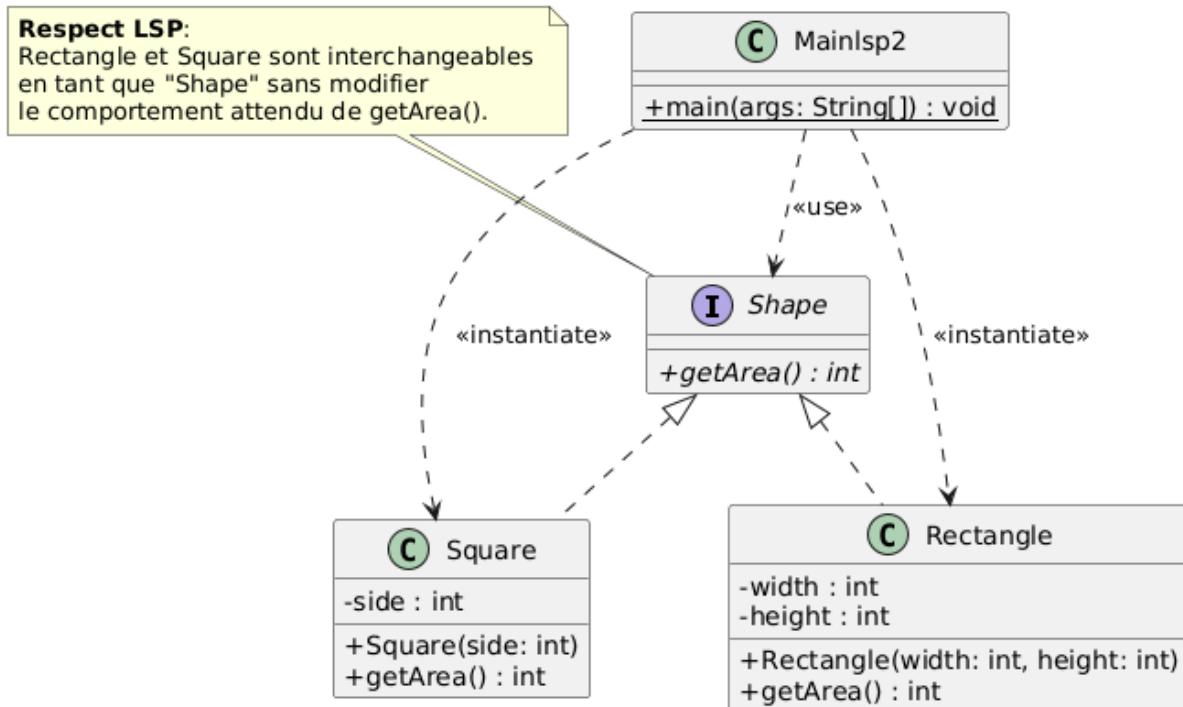
3. LSP – Liskov Substitution Principle

(Les sous-classes doivent être substituables sans casser le comportement)

- Avant refactoring (Violation avec Square héritant de Rectangle → bug)



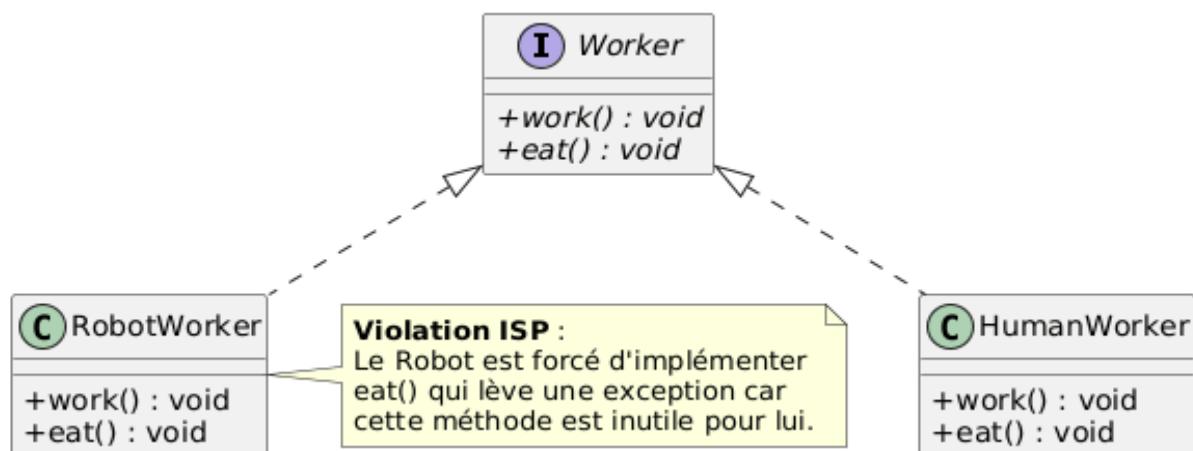
- Après refactoring (correction : interface commune ou séparation)



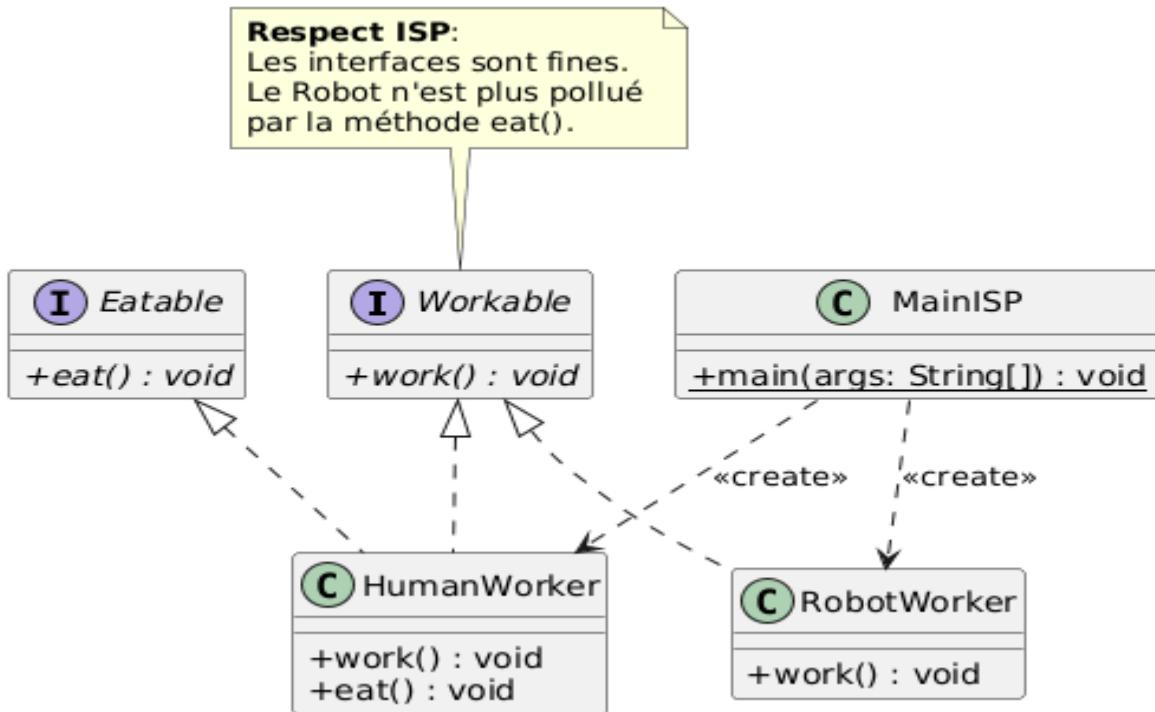
4. ISP – Interface Segregation Principle

(Interfaces spécifiques plutôt qu'une grosse interface générale)

- Avant refactoring (interface "fat" forçant des méthodes inutiles)

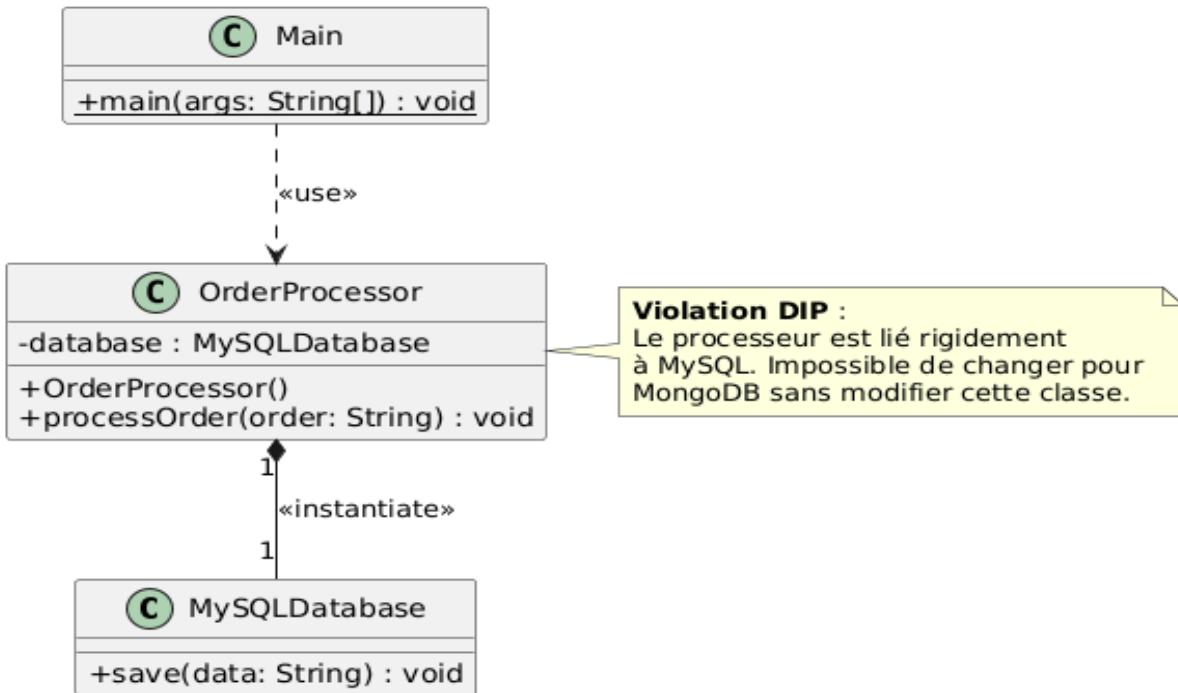


- Après refactoring (interfaces séparées et ciblées)



5. DIP – Dependency Inversion Principle (Dépendre d'abstractions, pas de détails concrets)

- Avant refactoring (dépendance directe vers les implémentations bas niveau)



- Après refactoring (inversion via abstractions/interfaces)

