

# TP\_0\_ICT301\_KENGNE\_MICHEL\_VANECK

**Nom : KENGNE MICHEL VANECK**

**Matricule : 23U2543**

## Principes SOLID

- **SRP - Single Responsibility Principle**

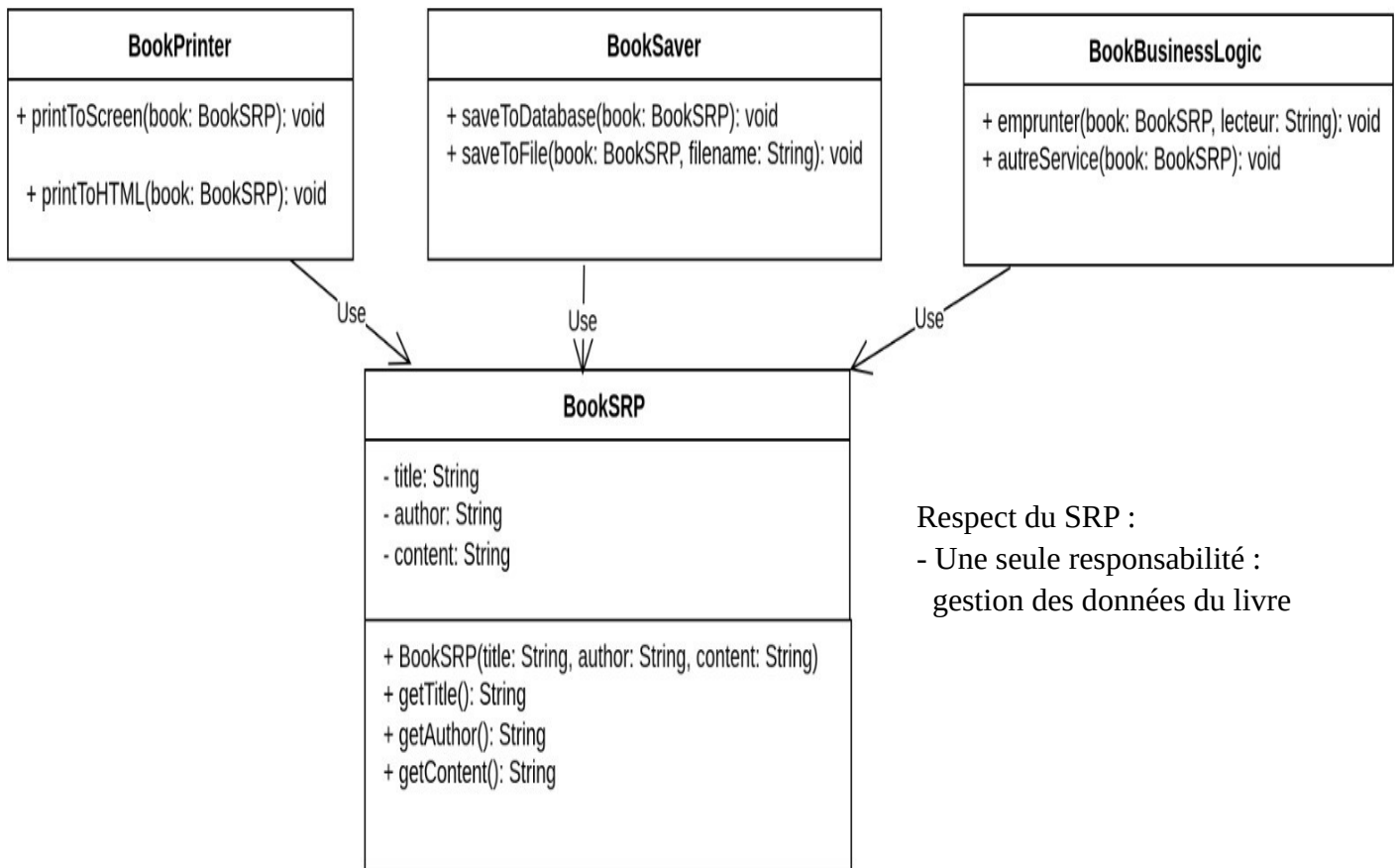
Avant refactoring :

Book
- title: String - author: String -content: String
+ Book(title: String, author: String, content: String) + getTitle(): String + getAuthor(): String + getContent(): String + printToScreen(): void + saveToDatabase(): void + emprunter(lecteur: String): void

Violation du SRP :

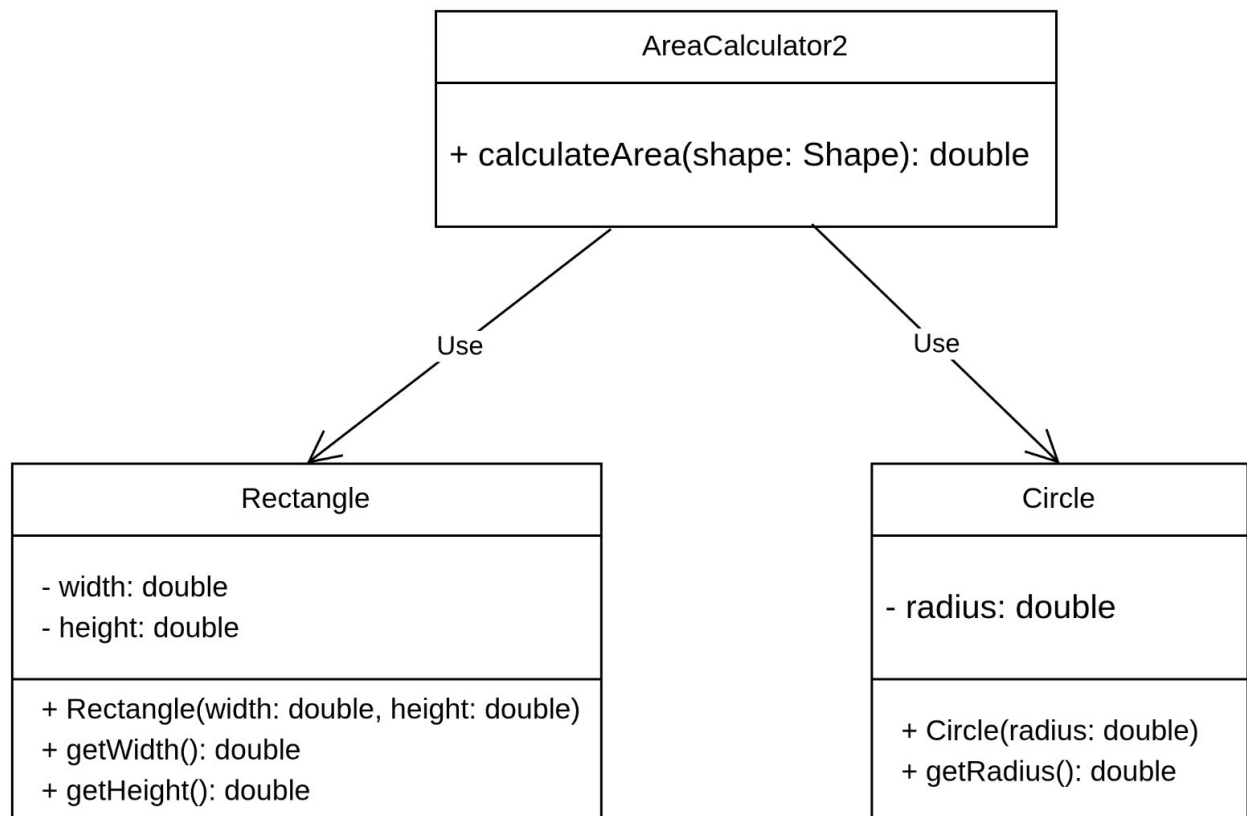
- Gestion des données
- Présentation (printToScreen)
- Persistance (saveToDatabase)
- Logique métier (emprunter)

Après refactoring :



- **OCP - Open/Closed Principle**

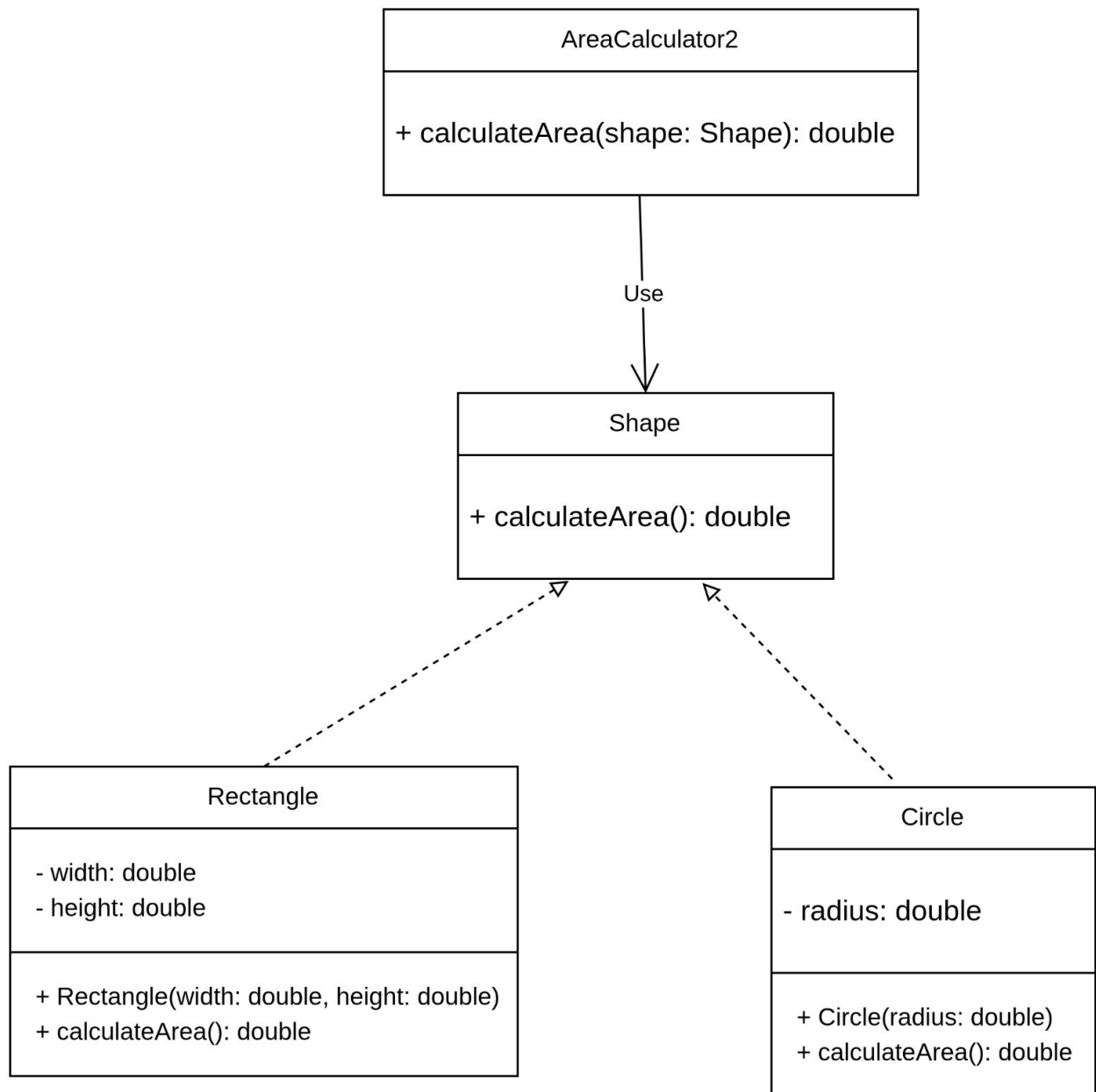
Avant refactoring :



Violation du OCP :

- Modification nécessaire pour ajouter une nouvelle forme
- Méthode `calculateArea` à modifier à chaque ajout

Après refactoring :

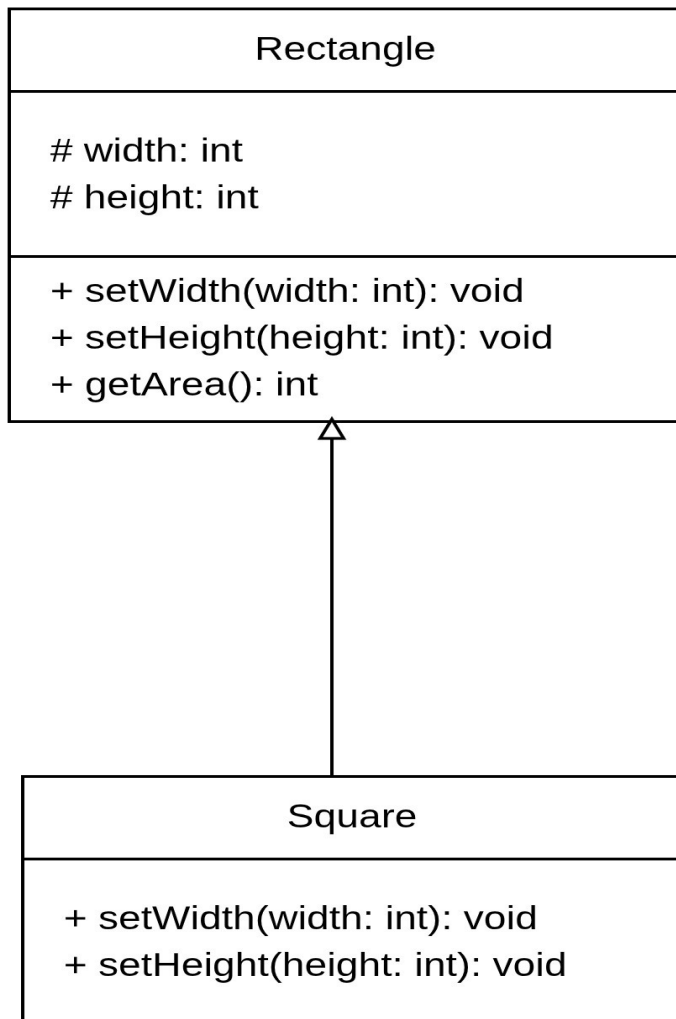


Respect du OCP :

- Ouvert à l'extension  
(nouvelles formes)
- Fermé à la modification  
(AreaCalculator2 ne change pas)

- **LSP - Liskov Substitution Principle**

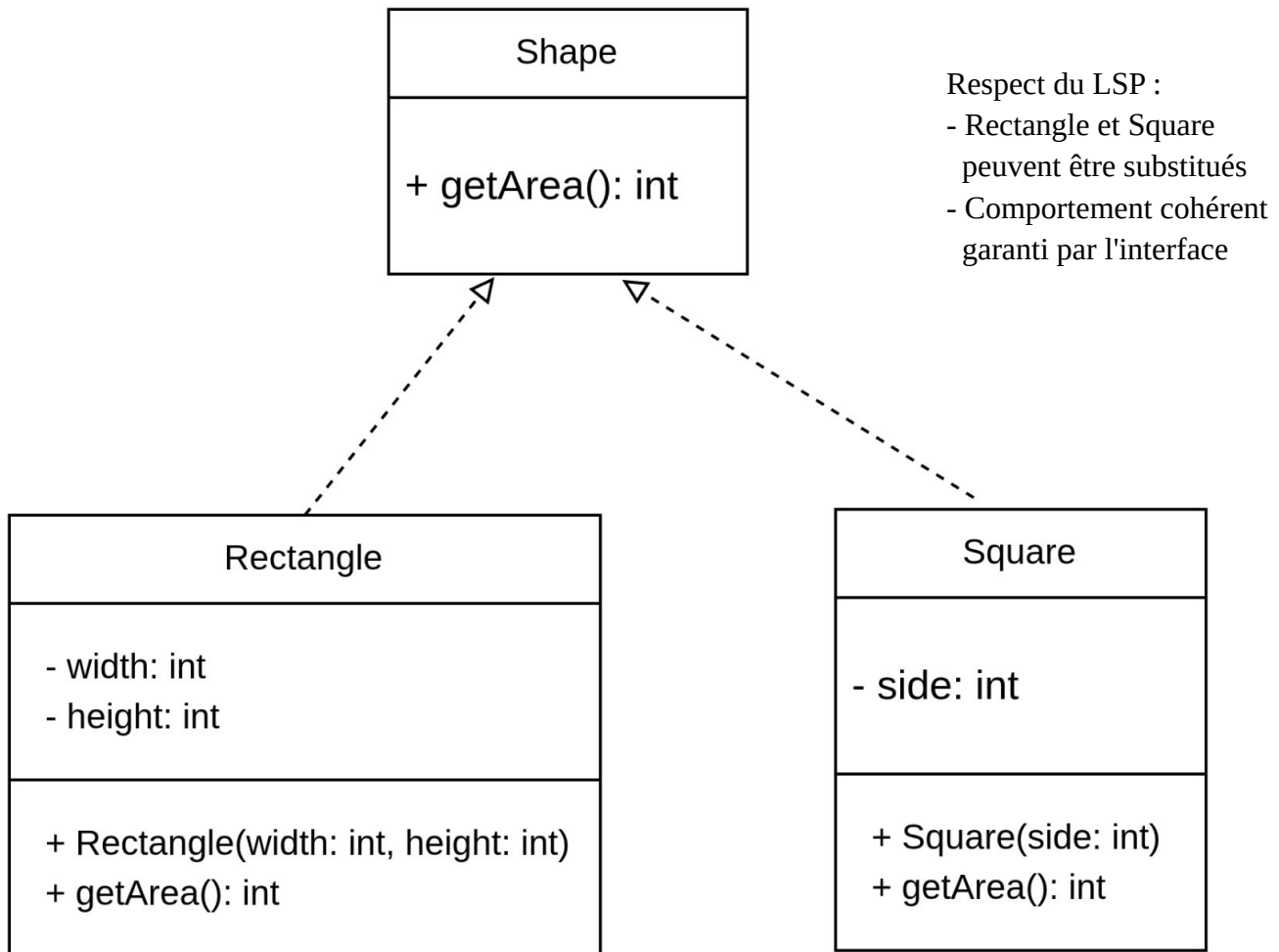
Avant refactoring :



Violation du LSP :

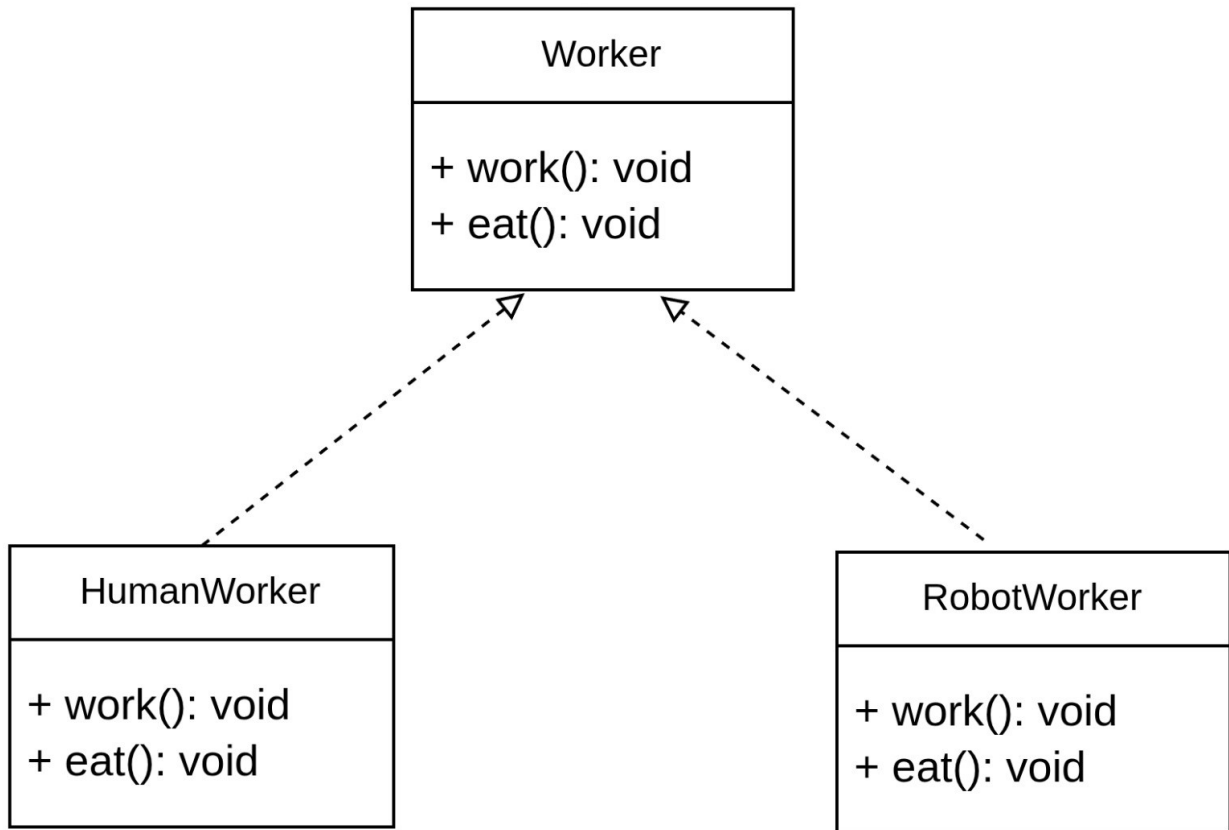
- setWidth modifie aussi height
- setHeight modifie aussi width
- Comportement inattendu lors de la substitution

Après refactoring :



- **ISP - Interface Segregation Principle**

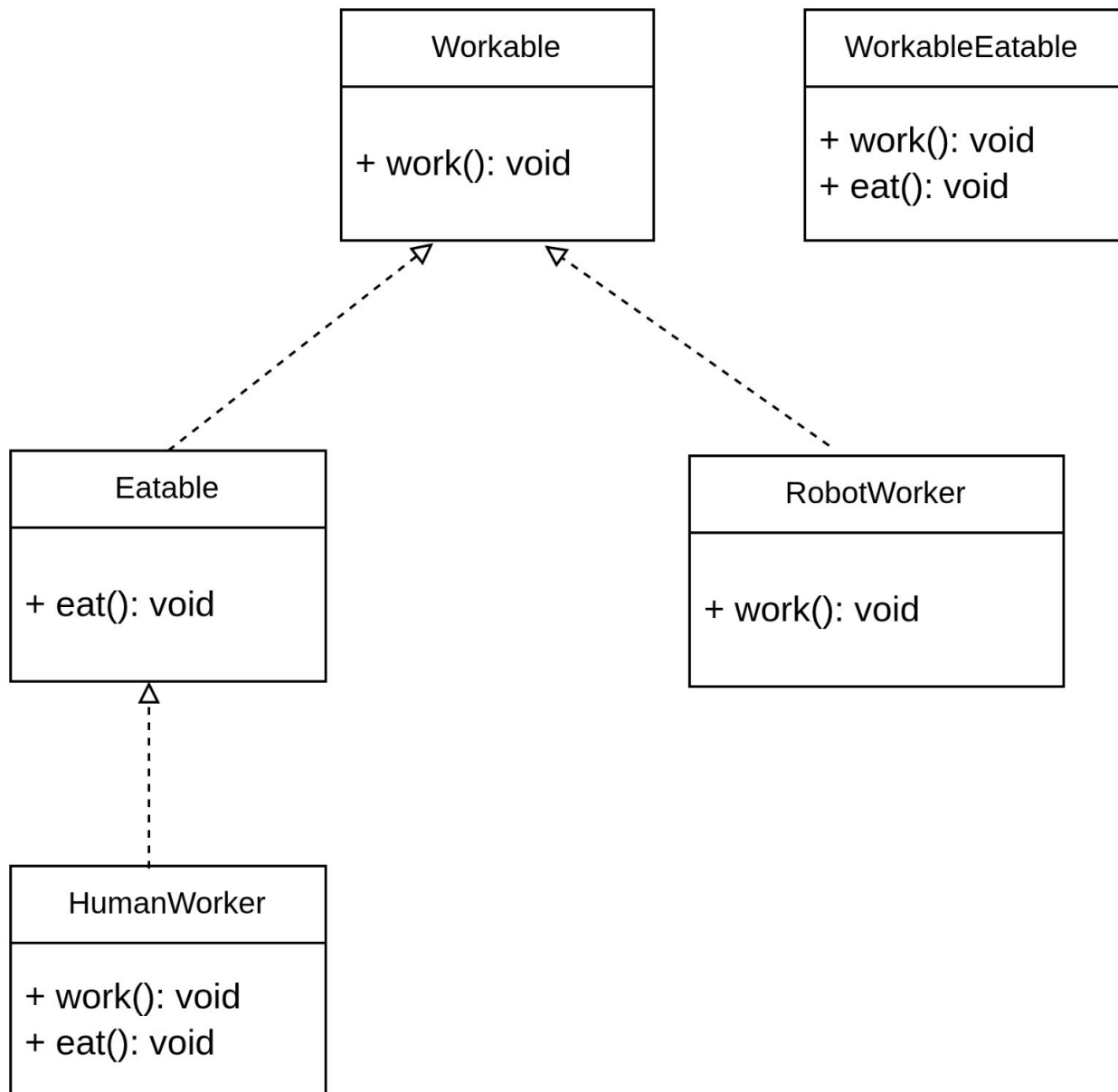
Avant refactoring :



Violation du ISP :

- RobotWorker forcé d'implémenter `eat()`
- Interface trop générale
- Dépendances inutiles

Après refactoring :



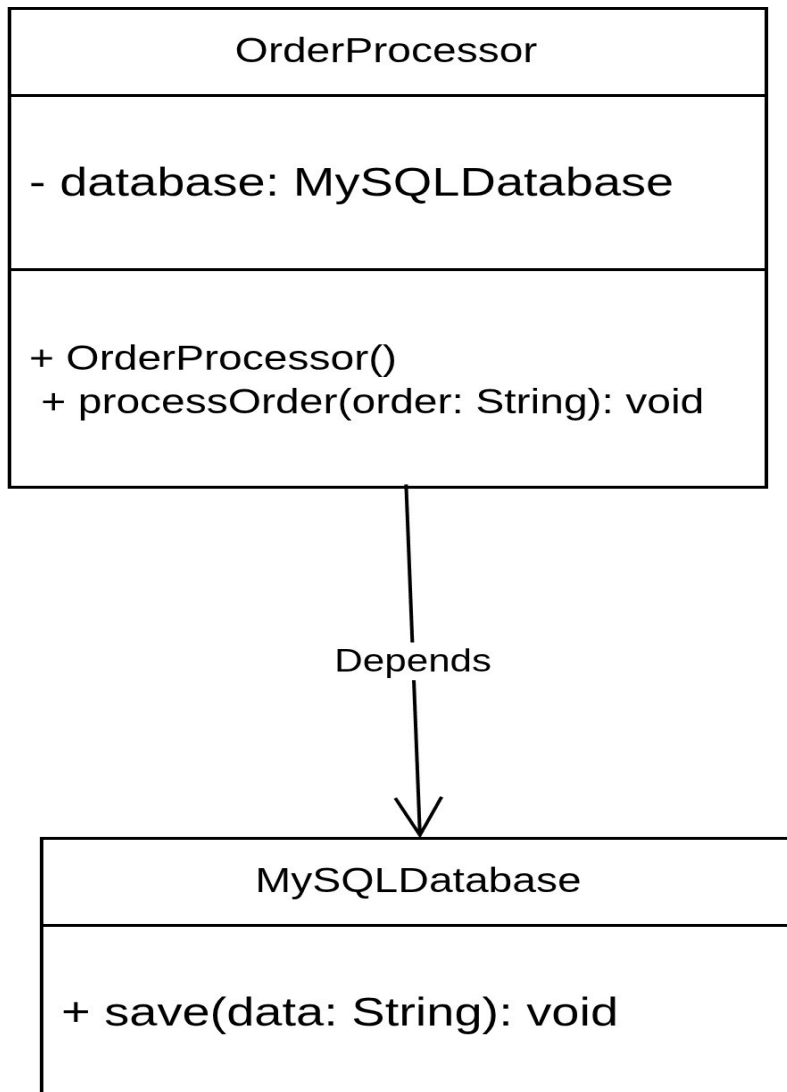
Respect du ISP :

- Interfaces spécifiques
- Pas de dépendances inutiles
- RobotWorker n'implémente que Workable



- **DIP - Dependency Inversion Principle**

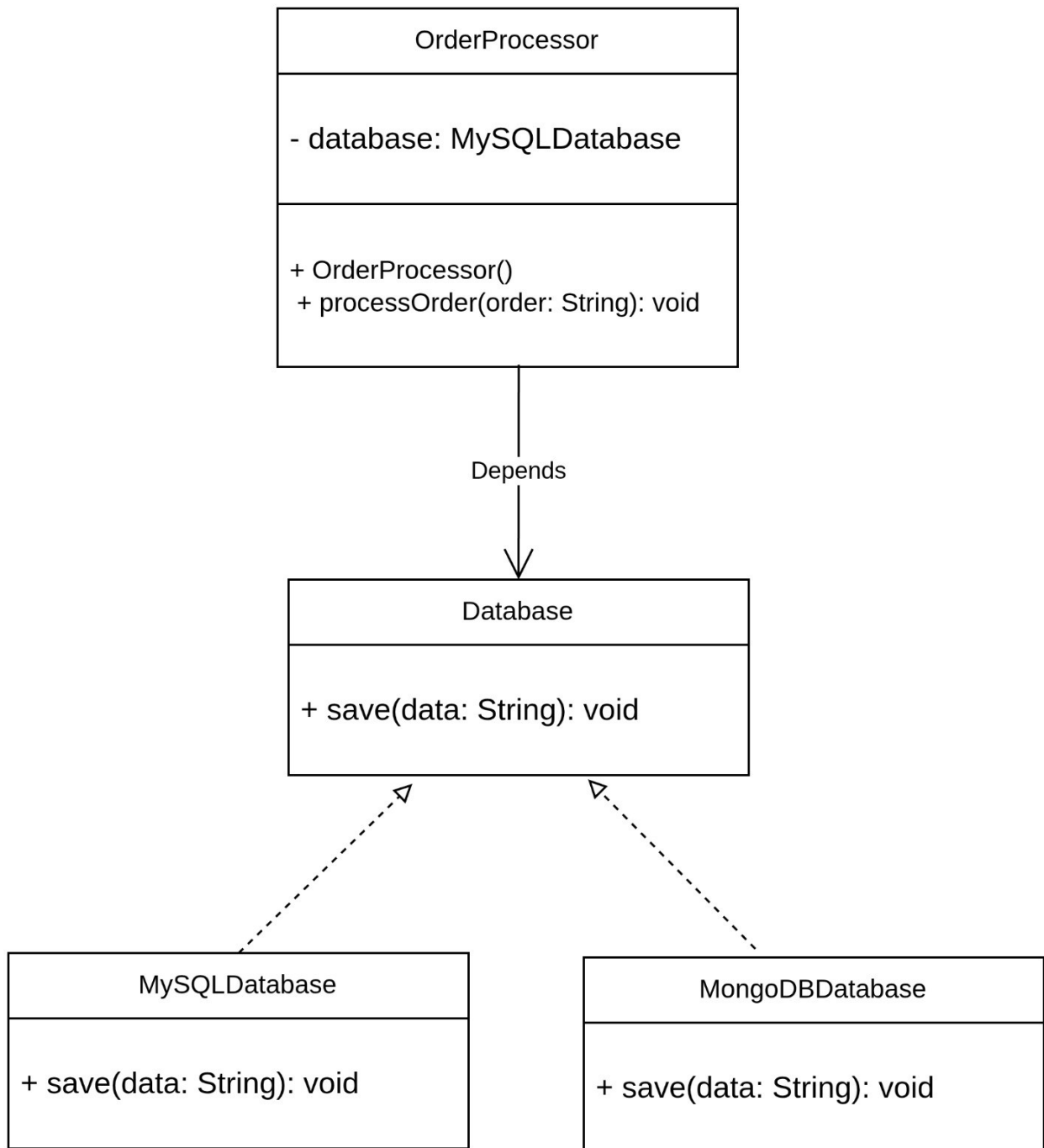
Avant refactoring :



Violation du DIP :

- Dépendance directe à MySQLDatabase
- Couplage fort
- Difficile à tester

Après refactoring :



Respect du DIP :

- Dépendance à l'abstraction
- Couplage faible
- Facile à tester et étendre