

## Analyse des Principes de Conception

### ✅ Principes Respectés:

1. **Single Responsibility Principle (SRP)** - ✅ Respecté
    - Chaque classe a une responsabilité unique et bien définie
    - Cours : représente un cours
    - CoursBuilder : construit des cours
    - GestionnaireEmploiDuTemps : gère les notifications
  2. **Open/Closed Principle (OCP)** - ✅ Respecté
    - Le pattern Decorator permet d'étendre les fonctionnalités sans modifier le code existant
    - On peut ajouter de nouveaux décorateurs sans toucher à CoursDecorator
  3. **Liskov Substitution Principle (LSP)** - ✅ Respecté
    - CoursEnLigne peut remplacer CoursDecorator sans problème
    - Les implémentations d'Observer sont interchangeables
  4. **Dependency Inversion Principle (DIP)** - ✅ Respecté
    - Les classes dépendent d'abstractions (ICours, Observer, Subject)
    - GestionnaireEmploiDuTemps dépend de l'interface Observer, pas des classes concrètes
- 

### ❌ Principes Violés:

#### 1. Interface Segregation Principle (ISP) - ⚠️ PARTIELLEMENT VIOLÉ

**Problème:** L'interface ICours force tous les cours à avoir une durée, même si certains types de cours pourraient ne pas avoir ce concept.