

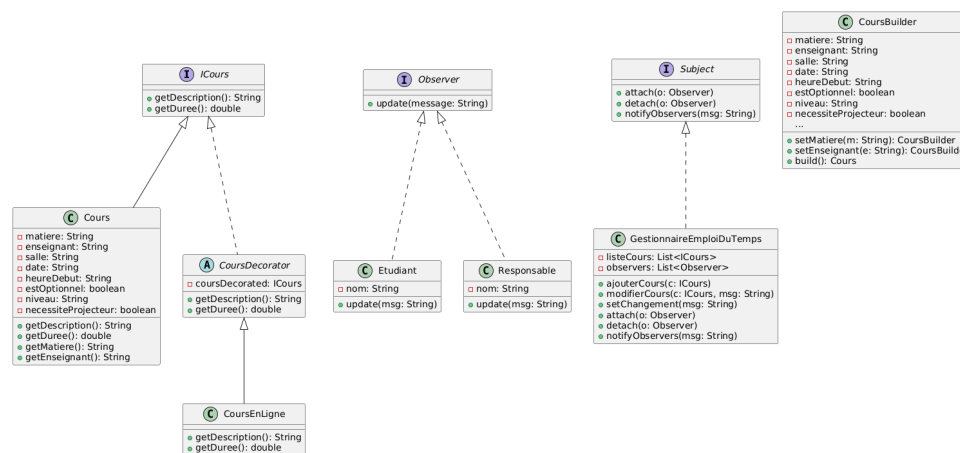
Rapport TP : Gestion d'Emploi du Temps

NEDJAR Mohammed

November 20, 2025

1 Diagramme de classes

Ci-dessous le diagramme de classes représentant notre implémentation :



2 Réponse à la question sur SOLID

Est-ce que notre code respecte les principes de conception logicielle SOLID ?

- **S - Single Responsibility Principle** : Oui, chaque classe a une seule responsabilité. Par exemple, la classe `CoursBuilder` est uniquement responsable de construire un objet `Cours`.
- **O - Open/Closed Principle** : Le code est en partie respectueux, car on peut ajouter de nouveaux types de cours via le `CoursDecorator` sans modifier les classes existantes.
- **L - Liskov Substitution Principle** : Respecté, les classes qui implémentent `ICours` peuvent être utilisées de manière interchangeable.
- **I - Interface Segregation Principle** : En partie respecté. L'interface `ICours` est simple et ne force pas les classes à implémenter des méthodes inutiles.
- **D - Dependency Inversion Principle** : Partiellement respecté. Certaines classes dépendent encore de classes concrètes (par exemple `GestionnaireEmploiDuTemps` manipule directement `ICours` et `Observer`), mais l'utilisation d'interfaces limite ce problème.