

1 Bonus : Diagramme de Classes

Voici le diagramme de classes UML décrivant le résultat final du TP, incluant les patrons Builder, Observer et Decorator.

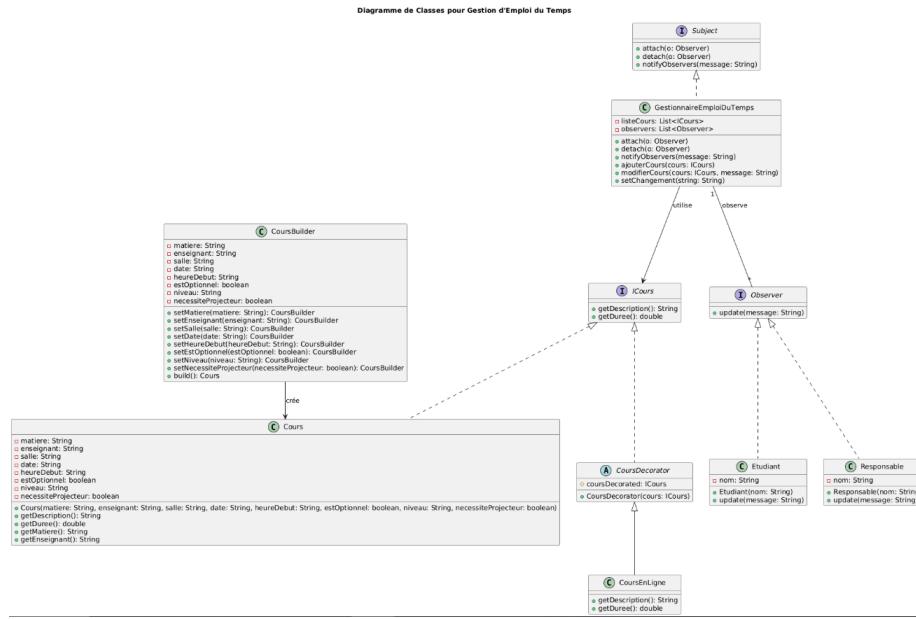


FIGURE 1 – Diagramme de classes pour la gestion de l'emploi du temps

2 Bonus : Respect des Principes de Conception Logicielle

Oui, le code respecte les principes de conception logicielle, en particulier les principes SOLID :

- **Single Responsibility Principle (SRP)** : Chaque classe a une responsabilité unique. Par exemple, `CoursBuilder` se concentre sur la construction, `GestionnaireEmploiDuTemps` sur les notifications.
- **Open-Closed Principle (OCP)** : Ouvert à l'extension (nouveaux décorateurs ou observateurs) mais fermé à la modification.
- **Liskov Substitution Principle (LSP)** : Les sous-classes remplacent les bases sans problème (ex. : `CoursEnLigne` pour `ICours`).
- **Interface Segregation Principle (ISP)** : Interfaces petites et spécifiques (`ICours`, `Observateur`).
- **Dependency Inversion Principle (DIP)** : Dépendances sur des abstractions (interfaces).

Aucun principe n'est violé. Le code est modulaire et extensible.