

Projet Design Patterns

Objectif :






Concevoir une application console (ou class library) permettant de gérer la planification d'interventions techniques, en appliquant plusieurs Design Patterns.

Contexte fonctionnel

L'application permet :

- La création de techniciens et d'interventions
- La génération automatique d'interventions types (maintenance, urgence...)
- La sauvegarde dans des fichiers simulés ou logs
- L'envoi de notifications internes (console/log)
- La gestion d'autorisations ou rôles d'utilisateurs (lecture/écriture)

Fonctionnalités à développer

-  Création d'interventions
 - Utiliser Factory Method ou Abstract Factory pour créer différents types d'interventions.
 - Chaque intervention a une date, un technicien, une durée, un lieu.
-  Ajout dynamique de fonctionnalités
 - Ajouter des décorateurs comme SuiviGPSDecorator, PiècesJointesDecorator.
 - Utiliser Decorator pour enrichir une intervention sans modifier la classe de base.
-  Centralisation des actions
 - Créer une Façade (GestionnaireInterventions) qui expose des méthodes simples comme CréerIntervention(), AssignerTechnicien(), Sauvegarder().
-  Notifications internes
 - Implémenter le pattern Observer pour notifier des abonnés (console, logs...) lors d'un changement d'intervention.
-  Gestion des droits
 - Simuler un Proxy pour vérifier qu'un utilisateur a le droit d'effectuer une action.

Patterns à utiliser obligatoirement

Pattern	Rôle dans le projet
Factory Method	Générer les différents types d'interventions
Decorator	Ajouter des options sans modifier la classe de base
Facade	Simplifier l'usage du système
Observer	Notifier des composants à chaque modification
Proxy (optionnel)	Ajouter une sécurité d'accès aux fonctionnalités

Livrables attendus

- Code commenté et organisé par pattern
- Diagrammes UML simplifiés (si possible)
- README décrivant les patterns utilisés
- Exemples d'exécution

Plan de travail (découpage 10h)

Étape	Tâche
1	Analyse + mise en place du modèle objet
2	Implémentation des patterns de création
3	Implémentation des décorateurs et façade
4	Ajout des observateurs + notifications
5	Ajout du proxy et sécurité d'accès
6	Intégration, tests unitaires et console
7	Nettoyage, documentation, rendu final