

Répartition équitable des tâches pour l'exposé sur les patrons de conception GoF de comportement

Vue d'ensemble

11 patrons comportementaux à couvrir pour **5 personnes** (environ 2 patrons par personne + intro/conclusion)

Personne 1 : Introduction + Observer & Command

Charge : ~20%

Tâches :

- **Introduction générale (10 min)**
 - Qu'est-ce qu'un patron de conception ?
 - Les 3 catégories : création, structure, **comportement**
 - Vue d'ensemble des 11 patrons comportementaux
 - Comment présenter un exposé efficace (méta-apprentissage)
 - **Observer (Observateur) (12 min)**
 - Concept : notifier automatiquement les dépendants
 - Exemple concret : système de notifications (réseaux sociaux, newsletter)
 - Démo code : observable/observer simple
 - Exercice : identifier des cas d'usage dans la vie quotidienne
 - **Command (Commande) (12 min)**
 - Concept : encapsuler une requête en objet
 - Exemple concret : télécommande universelle, éditeur de texte (undo/redo)
 - Activité interactive : simuler un système de commandes avec la classe
-

Personne 2 : Strategy & State

Charge : ~20%

Tâches :

- **Strategy (Stratégie)** (15 min)
 - Concept : encapsuler des algorithmes interchangeables
 - Exemples concrets :
 - Méthodes de paiement (carte, PayPal, crypto)
 - Algorithmes de compression (ZIP, RAR, 7z)
 - Calcul de tarifs (standard, étudiant, senior)
 - Démo code : système de paiement avec différentes stratégies
 - Exercice pratique : implémenter une stratégie de tri
 - **State (État)** (15 min)
 - Concept : changer le comportement selon l'état interne
 - Exemples concrets :
 - Distributeur automatique
 - Lecteur audio (play, pause, stop)
 - Statut de commande (en attente, expédiée, livrée)
 - Comparaison avec Strategy
 - Activité : diagramme d'états d'une porte automatique
-

Personne 3 : Chain of Responsibility & Mediator

Charge : ~20%

Tâches :

- **Chain of Responsibility (Chaîne de responsabilité)** (15 min)
 - Concept : déléguer une requête le long d'une chaîne
 - Exemples concrets :
 - Support client (niveau 1, 2, 3)
 - Validation de formulaire (plusieurs validateurs)
 - Système d'approbation hiérarchique (manager → directeur → PDG)
 - Démo code : pipeline de traitement de requêtes
 - Exercice interactif : créer une chaîne de filtres anti-spam
- **Mediator (Médiateur)** (15 min)
 - Concept : centraliser les communications complexes
 - Exemples concrets :
 - Chat room (les utilisateurs ne communiquent pas directement)
 - Contrôleur de trafic aérien
 - Interface graphique (composants qui interagissent)
 - Diagramme : avant/après médiateur
 - Activité : refactoriser un système couplé avec un médiateur

Personne 4 : Iterator, Visitor & Memento

Charge : ~20%

Tâches :

- **Iterator (Itérateur)** (10 min)
 - Concept : parcourir une collection sans exposer sa structure
 - Exemples concrets :
 - Playlist musicale
 - Menu de restaurant
 - Parcours d'arbre binaire
 - Démo : itérateur personnalisé vs itérateurs natifs
 - Mini-exercice : utiliser différents types d'itérateurs
- **Visitor (Visiteur)** (10 min)
 - Concept : ajouter des opérations sans modifier les classes
 - Exemples concrets :
 - Export de documents (PDF, HTML, Markdown)
 - Calcul d'impôts sur différents types de produits
 - Analyse syntaxique (AST)
 - Démo code : visiteur sur une structure composite
 - Discussion : avantages/inconvénients
- **Memento (Mémento)** (10 min)
 - Concept : sauvegarder/restaurer l'état d'un objet
 - Exemples concrets :
 - Undo/Redo dans un éditeur
 - Sauvegarde de jeu vidéo
 - Historique de navigation
 - Comparaison avec Command pour l'undo
 - Exercice rapide : implémenter un système de snapshots

Personne 5 : Template Method, Interpreter & Conclusion

Charge : ~20%

Tâches :

- **Template Method (Méthode Template) (12 min)**
 - Concept : définir le squelette d'un algorithme
 - Exemples concrets :
 - Processus de cuisson (préparation → cuisson → service)
 - Framework de tests (setup → test → teardown)
 - Connexion à différentes bases de données
 - Démo code : classe abstraite avec template method
 - Exercice : identifier les étapes variables vs fixes
 - **Interpreter (Interpréteur) (10 min)**
 - Concept : interpréter un langage ou une grammaire
 - Exemples concrets :
 - Calculatrice d'expressions mathématiques
 - Moteur de règles métier
 - Expressions régulières simplifiées
 - Note : patron moins utilisé, présentation concise
 - Demo simple : évaluer une expression arithmétique
 - **Synthèse et Conclusion (12 min)**
 - Tableau comparatif des 11 patrons
 - Guide de sélection : quel patron pour quel besoin ?
 - Patrons souvent combinés ensemble
 - Anti-patterns à éviter
 - Quiz final interactif (Kahoot/Mentimeter)
 - Questions/Réponses
 - Conseils pour réussir un exposé technique
-

Éléments interactifs à intégrer par tous :

Techniques pédagogiques :

1. **Démarrage** : Question ou problème concret (1-2 min)
2. **Analogie du quotidien** : relier au vécu des étudiants
3. **Diagramme UML simplifié** : visualisation claire
4. **Code commenté** : 10-15 lignes maximum par exemple
5. **Exercice rapide** : 3-5 minutes d'application
6. **Comparaison** : similitudes/différences avec d'autres patrons

Outils interactifs suggérés :

- **Kahoot/Mentimeter** : quiz en temps réel
- **Miro/Mural** : diagrammes collaboratifs
- **Live coding** : IDE partagé (VS Code Live Share)
- **Études de cas** : projets réels utilisant ces patrons

- **QR codes** : liens vers ressources complémentaires
-

Structure type par patron (30 min) :

1. **Problème concret** (3 min) - Pourquoi ce patron existe ?
 2. **Solution proposée** (5 min) - Comment le patron résout le problème
 3. **Structure UML** (3 min) - Diagramme simplifié
 4. **Exemple de code** (7 min) - Implémentation commentée
 5. **Cas d'usage réels** (5 min) - Applications concrètes
 6. **Exercice/Discussion** (5 min) - Interaction avec l'audience
 7. **Points clés** (2 min) - Récapitulatif
-