

QCM – Pattern Décorateur

1. À quoi sert un design pattern ?

- A) À écrire du code le plus rapidement possible
- B) À remplacer la documentation
- C) À compliquer inutilement la structure du programme
- D) À rendre le code plus modulable et réutilisable

2. Quelles sont les trois catégories de patterns définies par le GoF ?

- A) Classe, Objet, Méthode
- B) Interface, Héritage, Polymorphisme
- C) Création, Structure, Comportement
- D) Abstrait, Concret, Statique

3. Quelle est la principale problématique que résout le pattern Décorateur ?

- A) Ajouter dynamiquement des fonctionnalités à un objet sans modifier sa classe
- B) Fusionner plusieurs classes en une seule
- C) Créer de nouveaux objets plus rapidement
- D) Supprimer des méthodes inutiles

4. Quelle est la différence entre un ConcreteComponent et un ConcreteDecorator ?

- A) ConcreteDecorator contient les fonctionnalités de base, ConcreteComponent ajoute des fonctionnalités
- B) ConcreteComponent contient les fonctionnalités de base, ConcreteDecorator ajoute des fonctionnalités
- C) ConcreteDecorator est toujours abstrait, ConcreteComponent est concret
- D) ConcreteComponent ne peut jamais être utilisé directement

5. Pourquoi utilise-t-on le pattern Décorateur plutôt que de créer des sous-classes pour chaque variation ?

- A) Pour combiner plusieurs objets en un seul
- B) Pour supprimer certaines méthodes inutiles
- C) Pour compliquer la hiérarchie des classes
- D) Pour ajouter des fonctionnalités à un objet de manière flexible