

NOMMER LE PATTERN...

M1 MIAGE IL-DP

2022-2023

- 1. Nommer les design patterns qui correspondent aux caractérisations ci-dessous :**
 - a. Enveloppe un ensemble d'objets pour en simplifier l'accès**
 - b. Enveloppe un objet pour lui ajouter une responsabilité**
 - c. Enveloppe un objet pour en contrôler et en gérer l'accès**
 - d. Enveloppe un objet pour en modifier l'interface**

- 2. Nommer les design patterns qui correspondent aux caractérisations ci-dessous :**
 - a. Permet de définir un objet qui change d'état et dont le comportement est fonction de l'état**
 - b. Permet de diffuser automatiquement l'état d'un objet qui change d'état**

VRAI OU FAUX ?

#2

M1 MIAGE IL-DP

2022-2023

- 1. Pour pouvoir mettre en œuvre un objet dont le comportement dépend d'un état, il faut utiliser le pattern State**
- 2. Pour mettre en œuvre un objet dont le comportement dépend d'un état, le pattern State est une alternative à l'utilisation d'une variable d'état et d'une conditionnelle qui teste l'état courant**
- 3. La principale différence entre State et Stratégie réside dans le mécanisme d'évolution : le changement de stratégie est externe (méthode `setStratégie()`) alors que c'est l'état lui-même qui provoque le changement d'état (opération interne)**

- 4. En mode « pull », un observateur reçoit une notification de changement d'état du sujet et doit faire une demande au sujet s'il veut connaître le nouvel état**
- 5. Tous les observateurs doivent être instances de la même classe (Observateur « concret »)**
- 6. Grâce à la méthode d'abonnement, on limite le couplage entre les classes qui définissent le sujet et celles qui définissent les observateurs**
- 7. Dans la mise en œuvre du pattern Observateur, l'interface Observateur ne dépend pas du mode de transfert de l'état (« pull » ou « push »)**

- 8. L'intention du pattern Proxy est de permettre le contrôle de l'accès à un objet au moyen d'un autre objet qui se substitue à l'objet contrôlé**
- 9. La présence d'un objet proxy doit être transparente pour l'objet client**
- 10. La mise en œuvre du pattern Proxy impose à la classe Proxy d'implanter le même supertype que la classe SujetRéal**
- 11. Une conséquence de l'utilisation du pattern Proxy est de réduire le couplage entre les classes**

- 12. Un médiateur est un objet qui centralise les interactions au sein d'un ensemble d'objets**
- 13. L'utilisation du pattern Médiateur est sans effet sur le couplage entre classes**
- 14. On peut voir un médiateur comme une façade interne à un ensemble d'objets en interaction**
- 15. La mise en œuvre du pattern Médiateur interdit toute interaction directe entre les objets en interaction (c.-à-d. sans passer le médiateur).**

16. Les patterns Adaptateur, Façade, Médiateur et Proxy introduisent un objet intermédiaire entre un objet client et l'objet (ou les objets) qui réalise(nt) le service demandé par le client