DESIGN PATTERNS - FAMILLE

Patrons comportementaux

Répartir les actions, les décisions et les événements entre collaborateurs.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Comparer les intentions des patrons de cette famille.
- Choisir un premier exemple selon le problème rencontré.

Prérequis. Connaître les classes, les objets, les fonctions et les exceptions Python.

Le besoin commun

Répartir les actions, les décisions et les événements entre collaborateurs.

Qui choisit l'action, qui la reçoit et comment le comportement évolue-t-il avec les données ou l'état ?

Commencez par le contrat du client. Décrivez les entrées, le résultat, les erreurs attendues et ce qui doit pouvoir changer. Le nom d'un patron aide ensuite à discuter la solution et ses compromis.

Les patrons à étudier

- **14. Chain of Responsibility**

  Transmettre une requête le long d'une chaîne jusqu'au gestionnaire qui peut répondre.

- **15. Command**

Représenter une action et son destinataire par un objet déclenché par un autre composant.

- **16. Iterator**

  Séparer les données de leur parcours avec le protocole d'itération Python.

- **17. Observer**

  Distribuer un changement à plusieurs abonnés sans intégrer leurs réactions au sujet.

- **18. Strategy**

  Confier une opération à une politique interchangeable reçue par le contexte.

- **19. Template Method**

  Fixer l'ordre d'un algorithme et laisser les sous-classes réaliser certaines étapes.

- **20. Mediator**

  Centraliser les interactions entre collègues dans un coordinateur explicite.

- **21. Memento**

  Capturer un état puis le restaurer sans demander à l'historique de le reconstruire.

- **22. State**

  Changer les opérations permises d'un objet selon son état et ses transitions.

- **23. Visitor**

  Ajouter des opérations à une famille stable de types avec des visiteurs distincts.

- **24. Interpreter**

  Représenter les règles d'une petite grammaire et évaluer leur arbre dans un contexte.

Version web : https://www.clearcode.fr/courses/programming/design-patterns/comportement