DESIGN PATTERNS - CHAPITRE 11

Facade en Python

Coordonner plusieurs composants derrière une opération simple pour le client.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Coordonner plusieurs composants derrière une opération simple pour le client.
- Distinguer simplification d'une interface et contrôle d'accès à un service.

Prérequis. Savoir utiliser classes, héritage, composition, listes, dictionnaires et exceptions Python. Les mécanismes spécifiques sont expliqués avant leur emploi.

Facade : intention et expérience

Facade : proposer une entrée simple

Une **Facade** coordonne plusieurs opérations d'un sous-système derrière une opération utile au client. `start()` organise ici processeur, mémoire et disque simulés. Les composants restent accessibles si un autre client a besoin de leurs opérations détaillées.

Python

```python
# example.py
# Subsystem classes
class CPU:
    def freeze(self):
        return "Freezing processor"

    def jump(self, position):
        return f"Jumping to position: {position}"

    def execute(self):
        return "Executing"

class Memory:
    def load(self, position, data):
        return f"Loading from position {position}, data: {data}"

class HardDrive:
    def read(self, lba, size):
        return f"Reading sector {lba}, size: {size}"

# Facade
class ComputerFacade:
    def __init__(self):
        self.cpu = CPU()
        self.memory = Memory()
        self.hard_drive = HardDrive()

    def start(self):
        print("Starting computer")
        self.cpu.freeze()
        self.memory.load("0x00", self.hard_drive.read("100", "1024"))
        self.cpu.jump("0x00")
        self.cpu.execute()
        print("Computer started")

# Client code
if __name__ == "__main__":
```

```
    computer = ComputerFacade()
    computer.start()
```

Vérifier et interpréter

Faites remplacer chaque composant par un faux qui enregistre ses appels, puis lancez start(). Vérifiez l'ordre des étapes. La façade simplifie le scénario courant sans empêcher un usage détaillé du sous-système ; elle ne doit pas devenir le point de passage de toutes les opérations sans rapport.

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