DESIGN PATTERNS - CHAPITRE 12

Flyweight en Python

Partager un état intrinsèque réutilisable en gardant l'état de chaque usage à l'extérieur.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Partager un état intrinsèque réutilisable en gardant l'état de chaque usage à l'extérieur.
- Identifier les attributs partagés et les risques d'une modification du cache.

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

Flyweight : intention et expérience

Flyweight : partager un état intrinsèque

Un **Flyweight** partage les données identiques entre plusieurs usages. Le caractère, la taille et la police forment ici l'état intrinsèque conservé dans le cache ; la position, propre à chaque affichage, reste un argument de `render()`. Ne modifiez pas les attributs d'un objet partagé : tous ses utilisateurs verraient ce changement. Le cache n'est pas borné dans cette démonstration.

Python

```python
# example.py
# Flyweight
class Character:
    def __init__(self, char, font_size, font_family):
        self.char = char
        self.font_size = font_size
        self.font_family = font_family

    def render(self, position_x, position_y):
        return f"Character {self.char} rendered at ({position_x},{position_y})
with font {self.font_family} size {self.font_size}"

# Flyweight Factory
class CharacterFactory:
    _characters = {}

    @staticmethod
    def get_character(char, font_size, font_family):
        key = (char, font_size, font_family)
        if key not in CharacterFactory._characters:
            CharacterFactory._characters[key] = Character(char, font_size,
font_family)
        return CharacterFactory._characters[key]

# Client code
if __name__ == "__main__":
    text = "Hello World"
    x = 0
    for char in text:
        character = CharacterFactory.get_character(char, 12, "Arial")
        print(character.render(x, 0))
        x += 1
```

Vérifier et interpréter

Demandez deux fois le même triplet caractère, taille et police : les références doivent être identiques. Changez un élément du triplet : un autre objet est attendu. Le partage économise les objets redondants mais retient aussi les entrées du cache ; il exige une politique de validité et, si nécessaire, une limite de taille.

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