DESIGN PATTERNS - CHAPITRE 23

Visitor en Python

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

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Ajouter des opérations à une famille stable de types avec des visiteurs distincts.
- Comparer le coût d'ajouter une opération et celui d'ajouter un type d'élément.

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

Visitor : intention et expérience

Ajouter une opération ou ajouter un type ?

Vous voulez ajouter plusieurs opérations sur une famille de types dont la structure change peu. Le patron **Visitor** place chaque opération dans un visiteur et donne aux éléments une méthode `accept(visitor)`.

L'appel `accept` choisit ici explicitement la méthode adaptée au type : `visit_text` ou `visit_picture`. C'est une forme de double sélection : l'élément et le visiteur contribuent à choisir l'opération. Python ne choisit pas automatiquement une surcharge selon les types des arguments.

Le premier visiteur compte les mots du texte ou de la légende ; le second retourne une étiquette. Ajoutez un visiteur sans modifier les deux éléments. Puis imaginez un troisième type : chaque visiteur devra apprendre à le traiter. Ce compromis favorise l'ajout d'opérations sur des types relativement stables.

Python

```python
# example.py
from dataclasses import dataclass

@dataclass
class Text:
    value: str

    def accept(self, visitor):
        return visitor.visit_text(self)

@dataclass
class Picture:
    caption: str

    def accept(self, visitor):
        return visitor.visit_picture(self)

class WordCount:
    def visit_text(self, element):
        return len(element.value.split())

    def visit_picture(self, element):
        return len(element.caption.split())

class KindLabel:
    def visit_text(self, element):
        return "text"

    def visit_picture(self, element):
        return "picture"

elements = [Text("Hello world"), Picture("A small tree")]
print(sum(element.accept(WordCount()) for element in elements))  # 5
print([element.accept(KindLabel()) for element in elements])     # ['text',
'picture']
```

Pour approfondir, consultez **la documentation de dataclasses**.

Vérifier et interpréter

Ajoutez un visiteur qui retourne les noms des types sans modifier Text ni Picture. Ajoutez ensuite un type Audio sur le papier et listez les visiteurs à adapter. Visitor favorise de nouvelles opérations sur une structure stable, mais rend l'ajout d'un type plus coûteux et peut nécessiter l'accès à ses données internes.

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