UE Génie logiciel 2
Mardi 16 novembre 2021

Design pattern Decorator

Damien MONDOU, Enseignant chercheur, La Rochelle Université
damien.mondou@univ-lr.fr

La Rochelle Université

Les classes de patterns

3 classes de patterns :

→ Patterns de création d'objets :

- **Abstract Factory**
- Builder
- **Factory Method**
- Prototype
- **Singleton**

La Rochelle Université

Les classes de patterns

3 classes de patterns :

- Patterns de création d'objets
- Patterns de comportement :
  - Chain of responsibility
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - **Observer**
  - **State**
  - Strategy
  - Template Method
  - **Visitor**

La Rochelle Université

Les classes de patterns

3 classes de patterns :

- Patterns de création d'objets
- Patterns de comportement
- Patterns structurels :
  - Adapter
  - Bridge
  - Composite
  - **Decorator**
  - Facade
  - Flyweight
  - Proxy

La Rochelle Université

Patterns structurels

Objectif : faciliter l'indépendance de l'interface d'un objet ou d'un ensemble d'objets vis-à-vis de son implantation. Dans le cas d'un ensemble d'objets, il s'agit aussi de rendre cette interface indépendante de la hiérarchie des classes et de la composition des objets.

→ Decorator

La Rochelle Université

Decorator

Problématique de base : ajouter dynamiquement des fonctionnalités supplémentaires à un objet. Cet ajout de fonctionnalités ne modifie pas l'interface de l'objet et reste donc transparent vis-à-vis des clients.

Solution : Encapsuler l'objet existant et y ajouter des comportements nouveaux.

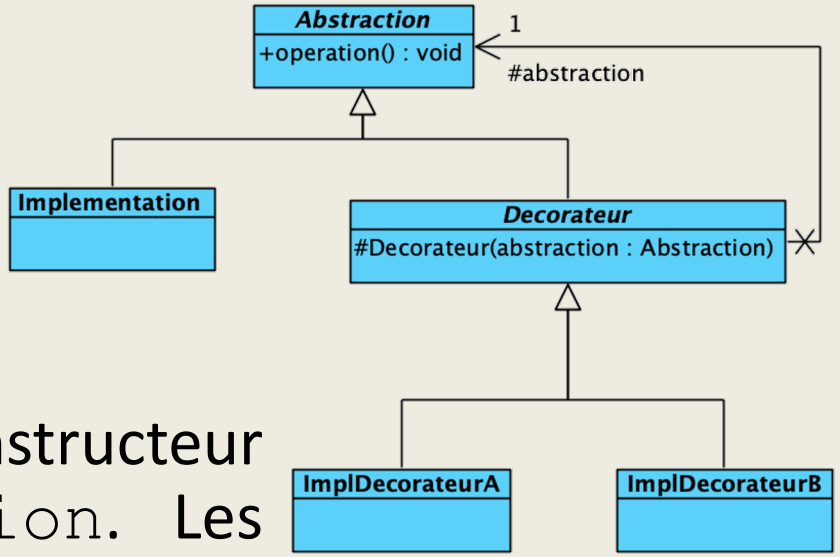
Conséquences :

- Plus souple que l'héritage statique
- Evite de surcharger les classes en haut de la hiérarchie
- Peut amener à créer de multiples petits objets

La Rochelle Université

Decorator - Structure

- `Abstraction` : définit l'interface générale (ex : `Dessert`)
- `Implementation` : cette classe contient l'implémentation de la classe / interface Abstraction correspondant aux fonctionnalités souhaitées à la base (ex : `Crepe` et `Gaufre`)
- `Decorateur` : définit l'interface du decorateur et contient une référence vers un objet Abstraction (ex : `DecorateurIngredient`)
- `ImplDecorateur` : les décorateurs ont un constructeur prenant en paramètre un objet `Abstraction`. Les méthodes des décorateurs appellent les mêmes méthodes de l'objet `Abstraction`. La décoration ajoute des fonctionnalités avant et/ou après ces appels (ex : `Chocolat` et `Chantilly`).



La Rochelle Université

Source

- *Design Patterns en Java* – 4ème édition, Laurent Debrauwer, Edition eni, mars 2018
- Cours d'Arnaud Revel

La Rochelle Université