UE Génie logiciel 2
Mercredi 10 novembre 2021

Design patterns Observer, Visitor et State

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 structurels :
  - Adapter
  - Bridge
  - Composite
  - **Decorator**
  - Facade
  - Flyweight
  - Proxy

La Rochelle Université

Les classes de patterns

3 classes de patterns :

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

La Rochelle Université

Patterns de comportement

Objectif : Fournir des solutions pour distribuer les traitements et les algorithmes entre les objets.

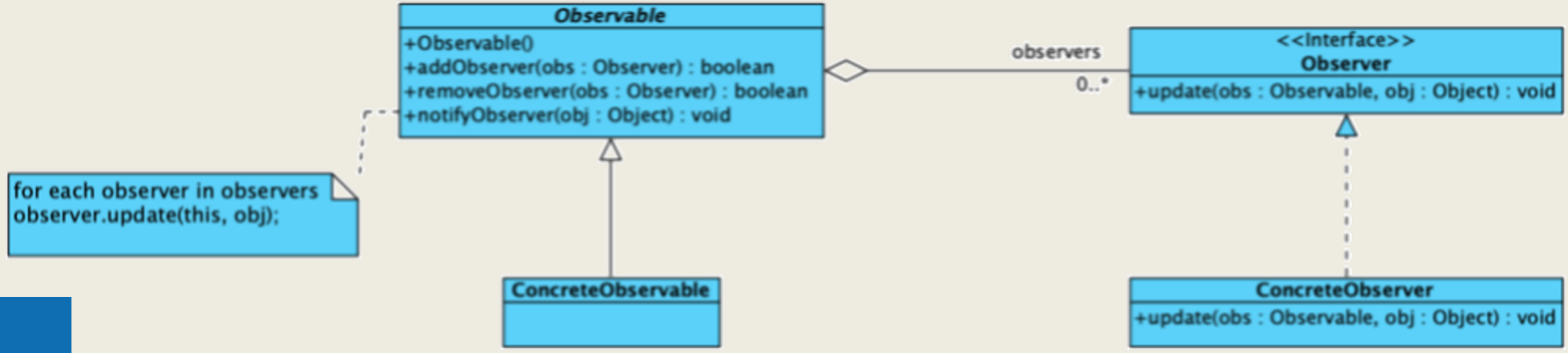
- Observer
- State
- Visitor

La Rochelle Université

Observer

Problématique de base : Permettre à un objet de réagir aux comportement d'un autre sans pour autant les lier « en dur ».

- **Solution :** Définir un objet comme `Observable` et donc capable de notifier des changements à des `Observer`, quels qu'ils soient



Un exemple détaillé sera présenté au TP2

La Rochelle Université

State

Problématique de base : Adapter le comportement d'un objet en fonction d'un contexte / de son état interne

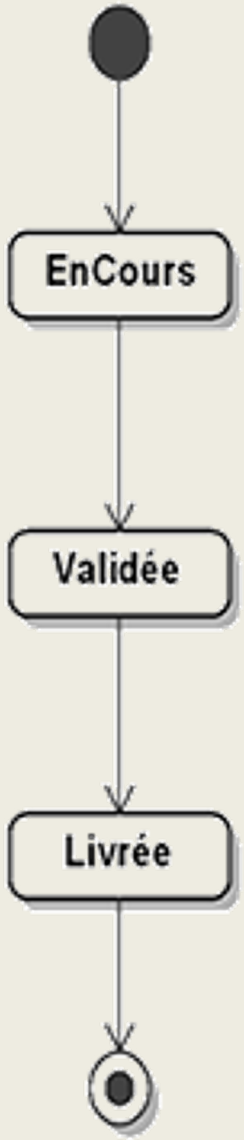
Solution : Gérer un état qui permet de rediriger le comportement vers la bonne classe .

La Rochelle Université

State - Exemple

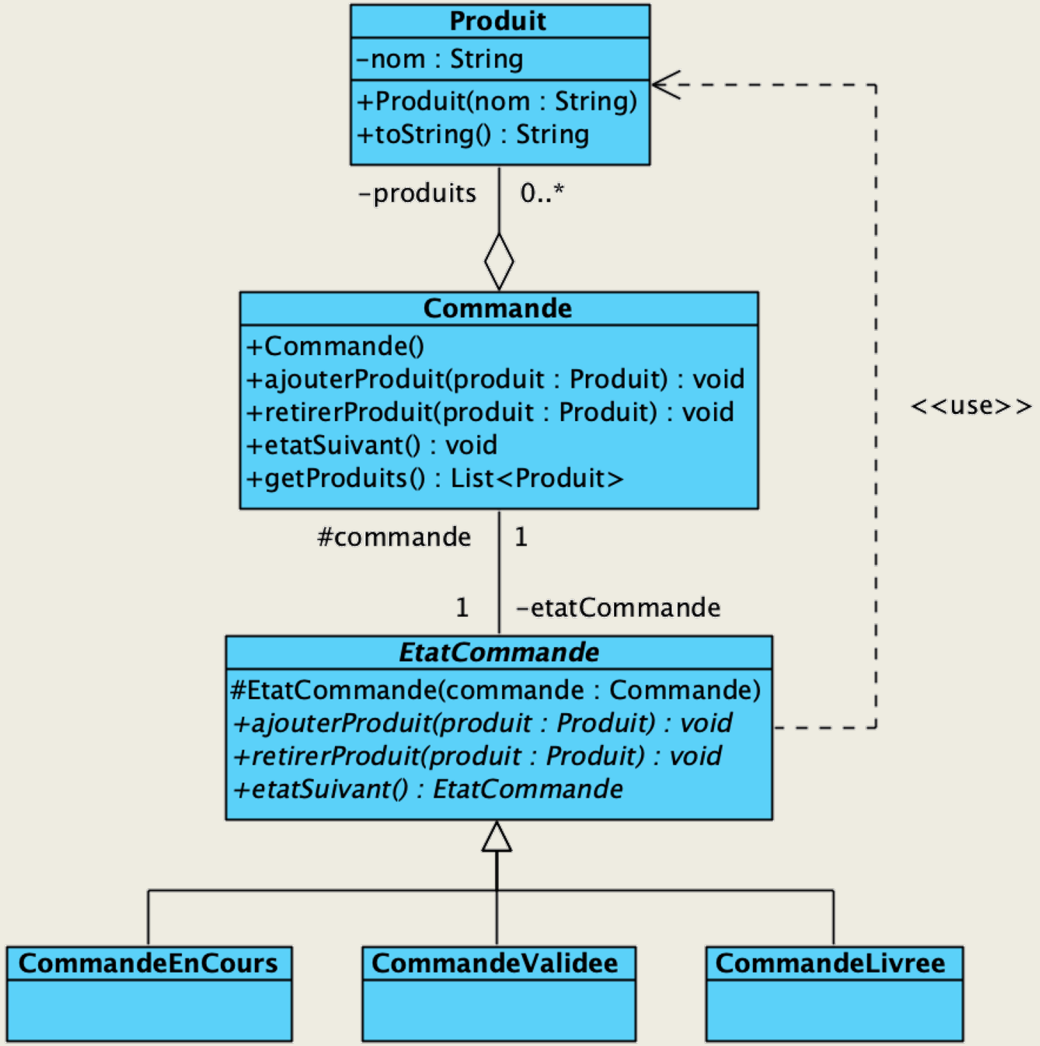
Une commande possède un cycle de vie particulier :

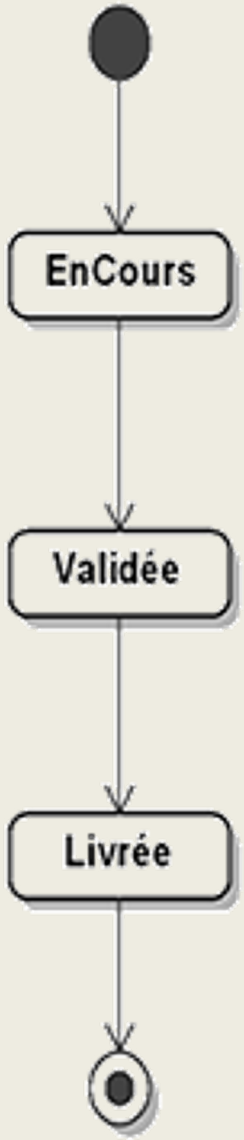
- Un état EnCours dans lequel le client peut ajouter et retirer des produits;
- Un état Validée où la commande est réglée;
- Un état Livrée lorsque le client a reçu sa commande.



La Rochelle Université

State - Exemple





La Rochelle Université

Le code de cet exemple est disponible sur Moodle

Visitor

Problématique de base : réaliser des opérations sur les éléments d'un objet sans changer son fonctionnement

Solution :

→ Utiliser objet tiers (Visitor) capable d'obtenir le résultat souhaité à partir des données de l'objet

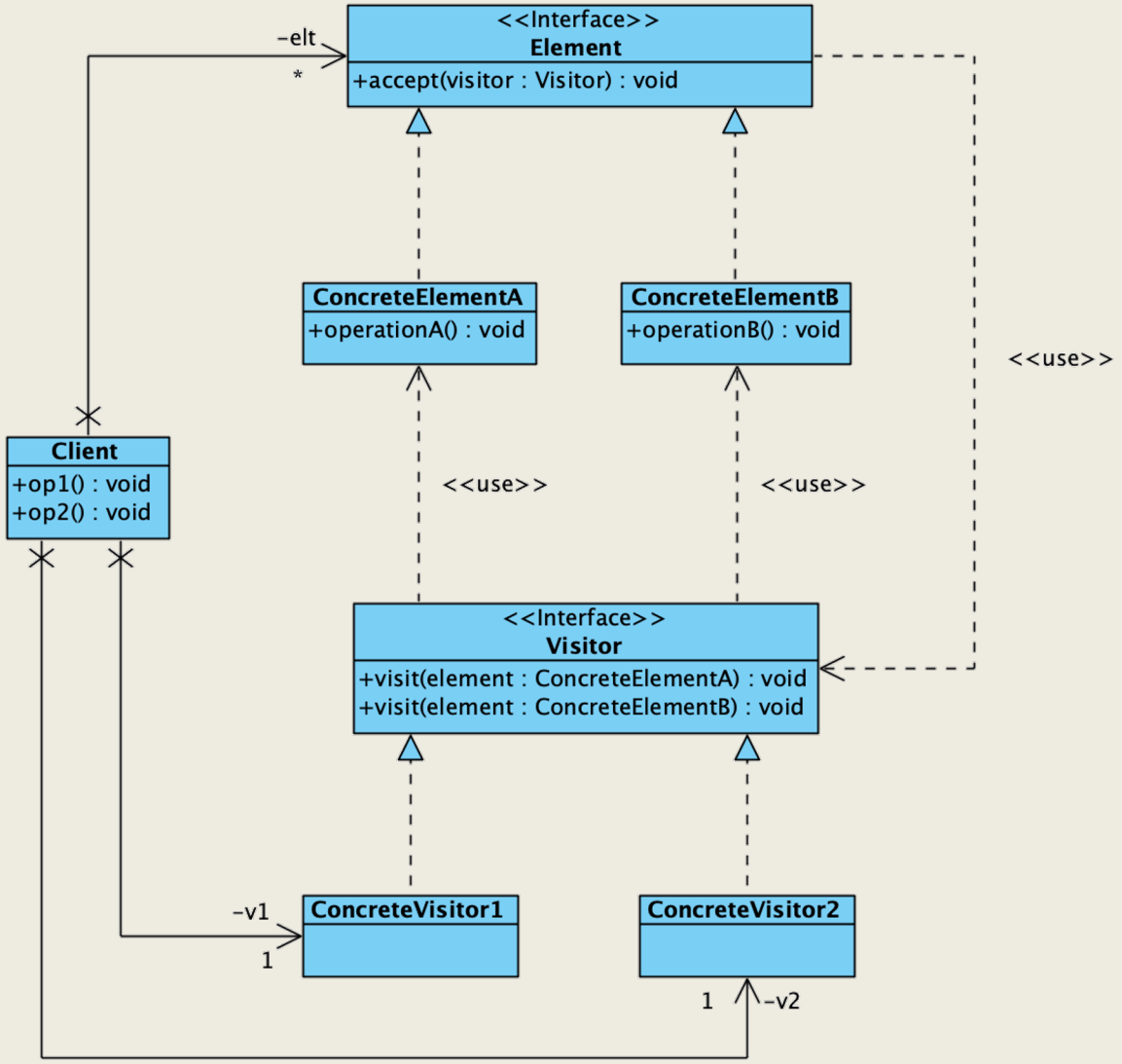
Conséquences :

→ Facilite l'ajout de nouvelles opérations

→ Rassemble les opérations du même type et isole celles différentes

La Rochelle Université

Visitor



Un exemple détaillé sera présenté au TD2

Source

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

La Rochelle Université