

Le Pattern D'Etat

Le pattern d'état s'inscrit parmi les patterns comportementaux du GoF. Il a pour but d'aider à améliorer les interactions entre les objets et dans les objets pour en assurer une meilleure flexibilité.

Structure du pattern d'Etat

Le pattern d'état se découpe en trois parties :

Le contexte ("*context*" en anglais) :

C'est la classe principale qui va changer de comportement en fonction des états

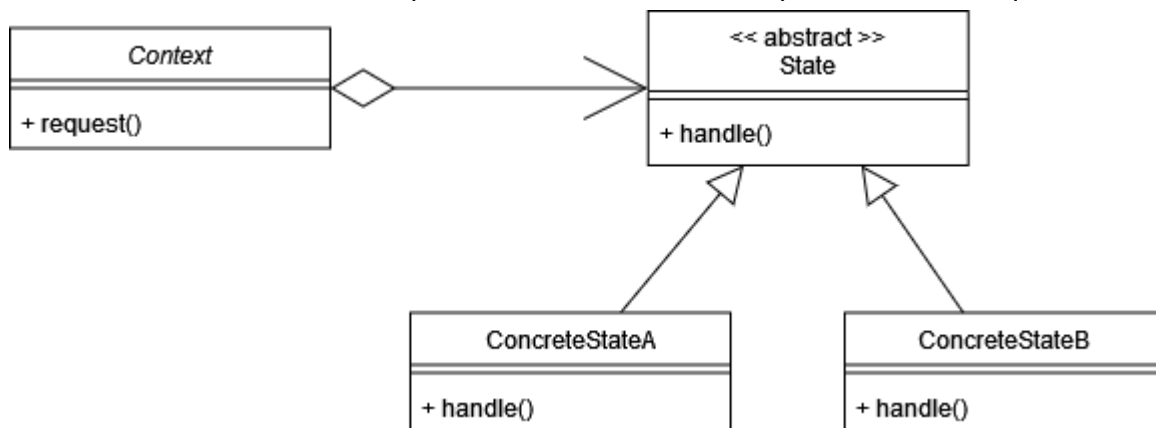
L'Etat ("*state*" en anglais) :

C'est une classe qui va être utilisée pour générer les différents états

L'Etat concret ("*concrete state*" en anglais) :

Ils représentent les différents états que peut avoir la classe contexte.

Ensembles, ils donnent vie au pattern d'état dont voici la représentation basique en UML.



Avantages / Inconvénients

Avantages :

- **SRP (Single Responsibility Principle)** : La classe Context ne gère pas les changements d'état.
- **OCP (Open/Closed Principle)** : L'ajout d'états se fait par de nouvelles classes et non en ajoutant des IF imbriqués
- Facilement applicable et maintenable.

Inconvénients :

- Sur-utilisation : utiliser ce pattern avec un objet qui change rarement d'état ou qui n'en possède très peu peut représenter une sur-utilisation du pattern
- Difficile à comprendre : Dans le cas d'un objet qui aurait des dizaines d'état changeant souvent, la lecture du code peut être compliquée pour un débutant du pattern