LE PATTERN ETAT

Intention

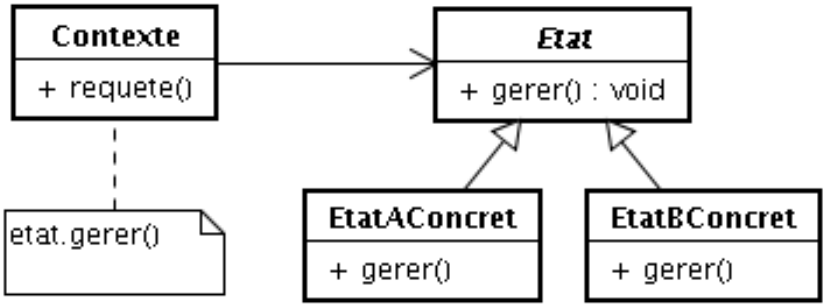
Permet à un objet de modifier son comportement, quand son état interne change. Tout se passera comme si l'objet changeait de classe.

Indications d'utilisation

On utilisera le pattern Etat lorsque :

- Le comportement d'un objet dépend de son état, et ce changement de comportement doit intervenir dynamiquement, en fonction de cet état.
- Les opérations comportent de grands pans entiers de déclarations conditionnelles fonctions de l'état de l'objet.
- Cet état est généralement désigné par une ou plusieurs énumération.
- Souvent, plusieurs opérations différentes contiendront la même structure conditionnelle.
- Le pattern Etat place dans une classe séparée, chacune des branches de la condition. Ceci permet de traiter l'état de l'objet comme un objet à part entière, qui peut varier indépendamment des autres objets.

Structure



Constituants

Contexte

- Définit l'interface intéressant les clients.
- Gère une instance d'une sous-classe EtatConcret qui définit l'état en cours.

Etat

- Définit une interface qui encapsule le comportement associé avec un état particulier de Contexte.

EtatConcret

- Chaque EtatConcret implémente un comportement associé avec l'état de contexte.

Collaborations

Le contexte délègue les requêtes spécifiques d'état à l'objet EtatConcret courant.

Un contexte peut passer en argument sa propre référence à l'objet Etat en charge de gérer la requête. Cet objet peut donc accéder au contexte, si nécessaire.

Contexte est l'interface primaire pour les utilisateurs. Ceux-ci peuvent composer un contexte à l'aide d'objets Etat. Une fois qu'ils ont configuré un contexte, les utilisateurs n'ont plus qu'à « traiter » directement avec les objets Etat.

C'est soit à Contexte, soit aux sous-classes EtatConcret, qu'il revient de décider de l'état qui succède à un autre état, et selon quelles modalités.