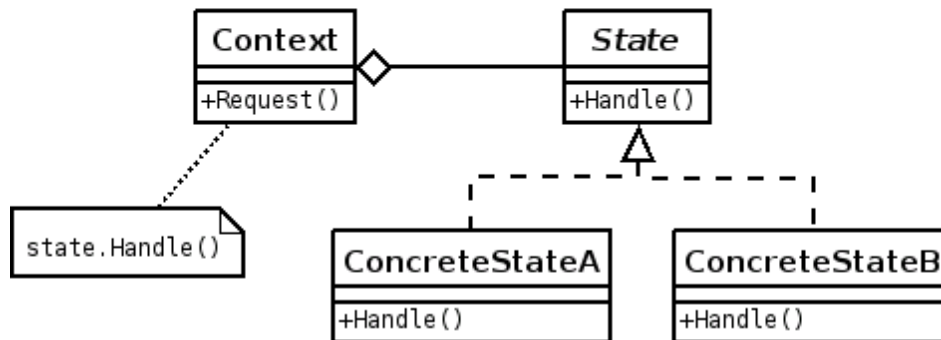


## Design Pattern : state

### Intent

*Allow an object to alter its behavior when its internal state changes. The object will appear to change its class.*

### Structure



### Éléments caractéristiques

- les comportements d'une ou plusieurs méthodes changent selon l'état, leur exécution est déléguée à l'objet qui représente l'état
- l'objet contrôle les changements d'état

### Exemples rencontrés

#### TD – Attractions : gestion du “hors service”.

La situation « en fonctionnement » ou « hors service » des attractions peut être gérée par le design pattern State. Le comportement d'un objet Ride est modifié selon chacune de ces situations. En effet le traitement des méthodes `accessGranted()`, `setAccessStrategy()`, `outOfOrder()` et `open()` est délégué à l'état. Cet état est géré par des attributs dont la valeur est une instance de classes internes (`NormalState` et `OutOfOrderState`). Le changement d'état est déclenché lors de l'exécution de la méthode `outOfOrder()` de `NormalState`, qui permet de passer vers l'état `OutOfOrder`, et de l'exécution de la méthode `open()` de `OutOfOrder`, qui permet de passer vers l'état `NormalState`.

