State Pattern

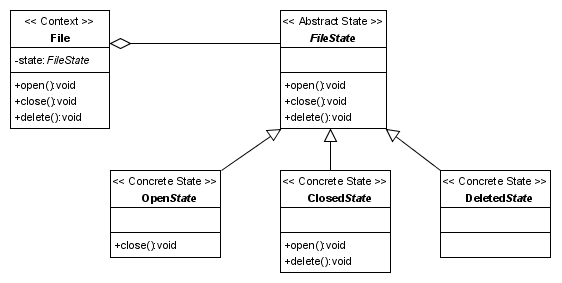
# Concept

## Which design problem is addressed

Es geht darum, dass der Caller nicht wissen muss, in welchem Status sich das aufgerufene Objekt befindet.

## How does the pattern work in general

Für jeden Zustand gibt es eine Klasse die ein gemeinsames Interface implementiert.



Der Context hat die gleichen Methoden implementiert wie das Interface. Der Caller ruft die Methode auf dem Context auf, dieser führt die Methode auf dem aktuellen State aus.

# Potential advantages

* **Erweiterbarkeit und Änderungsstabilität**. Neue Zustände können einfach ins System integriert werden, ohne dabei bestehenden Code ändern zu müssen. Änderungen am zustandsabhängigen Verhalten betreffen lediglich eine Zustandsklasse und nicht den Context.
* **Intuitivität und Verständlichkeit**. Durch die hohe Kohäsion und der Delegation von Verantwortlichkeiten (das Verhalten eines Zustands ist im entsprechenden Zustandsobjekt selbst gekapselt/lokalisiert) wird der Code leicht verständlich. Das mannigfaltige Verhalten eines Objekts wird auf verschiedene Zustandsobjekte verteilt.

# Potential disadvantages

* Hohe Klassenanzahl
* Weniger kompakt als eine einzige Klasse. Allerdings ist dies gerade das Ziel des Entwurfsmusters: Die Aufteilung des Verhaltens von einer einzigen unübersichtlichen Klasse auf mehrere Zustandsobjekte

Implementation possibilities

* In einem **Zeichenprogramm** kann der Maus-Cursor als Werkzeug fungieren. Anstatt das sich der Zustand des Cursors ändert, ändert sich der Zustand der Zeichenfläche bzw. des Editors. Führte ein Klick in die Zeichenfläche im Zustand "Pinselwerkzeug" noch zur Erstellung eines Punktes, so führt der selbe Klick im Zustand "Füllwerkzeug" zum Ausfüllen der gesamten Zeichenfläche mit einer bestimmten Farbe. Das Verhalten des Editors bzw. seine Reaktion auf Events (Mausklick) ändert sich in Abhängigkeit zu seinen Zustand. Der Zustand wird durch die Auswahl eines Werkzeugs gesetzt.
* **Zustandsautomaten** jeglicher Art
* Repräsentation von Zuständen **von (Netzwerk)Verbindungen** Modellierung einer TCP-Verbindung mit den Operationen open(), close(), confirm(). Das Verhalten dieser Methoden ist jedoch abhängig von dem Zustand der TCP-Verbindung: Etabliert, Lauschend oder Geschlossen.