**СОСТОЯНИЕ** - позволяет объектам менять поведение в зависимости от своего состояния. Извне создаётся впечатление, что изменился класс объекта, т.к. полностью изменилось поведение, это похоже на уже другой объект. Можно рассматривать как надстройку над Стратегией: оба паттерна используют композицию, чтобы менять поведение основного объекта, делегируя работу вложенным объектам-помощникам, но в Стратегии эти объекты не знают друг о друге и никак не связаны, а в Состоянии сами конкретные состояния могут переключать контекст

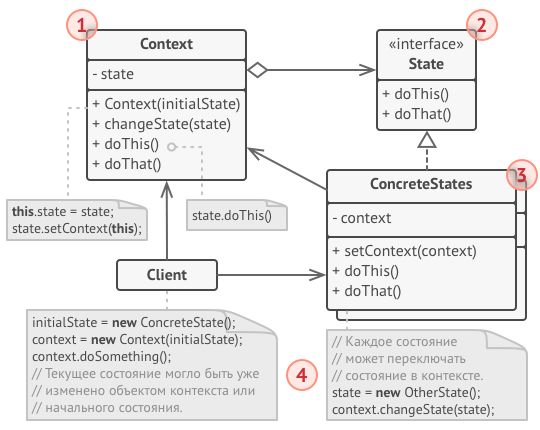
ПРИМЕНЕНИЕ:

- когда есть объект, поведение которого кардинально меняется в зависимости от внутреннего состояния, причём типов состояний много, и их код часто меняется

- когда код класса содержит множество больших, похожих друг на друга, условных операторов, которые выбирают поведения в зависимости от текущих значений полей класса

- когда вы сознательно используете табличную машину состояний, построенную на условных операторах, но вынуждены мириться с дублированием кода для похожих состояний и переходов

РЕАЛИЗАЦИЯ:

**1. КОНТЕКСТ**: хранит ссылку на объект состояния и делегирует ему часть работы, зависящей от состояний. Контекст работает с этим объектом через общий интерфейс состояний. Контекст должен иметь метод для присваивания ему нового объекта-состояния.

**2. СОСТОЯНИЕ**: описывает общий интерфейс для всех конкретных состояний.

**3. КОНКРЕТНЫЕ СОСТОЯНИЯ**: реализуют поведения, связанные с определённым состоянием контекста. Иногда приходится создавать целые иерархии классов состояний, чтобы обобщить дублирующий код. Состояние может иметь обратную ссылку на объект контекста. Через неё не только удобно получать из контекста нужную информацию, но и осуществлять смену его состояния.

**4.** И контекст, и объекты конкретных состояний могут решать, когда и какое следующее состояние будет выбрано. Чтобы переключить состояние, нужно подать другой объект-состояние в контекст.

ПЛЮСЫ:

- избавляет от множества больших условных операторов машины состояний

- концентрирует в одном месте код, связанный с определённым состоянием

- упрощает код контекста

МИНУСЫ:

- может неоправданно усложнить код, если состояний мало и они редко меняются