СОСТОЯНИЕ(STATE) НА PYTHON

2019

Шаблоны проектирования

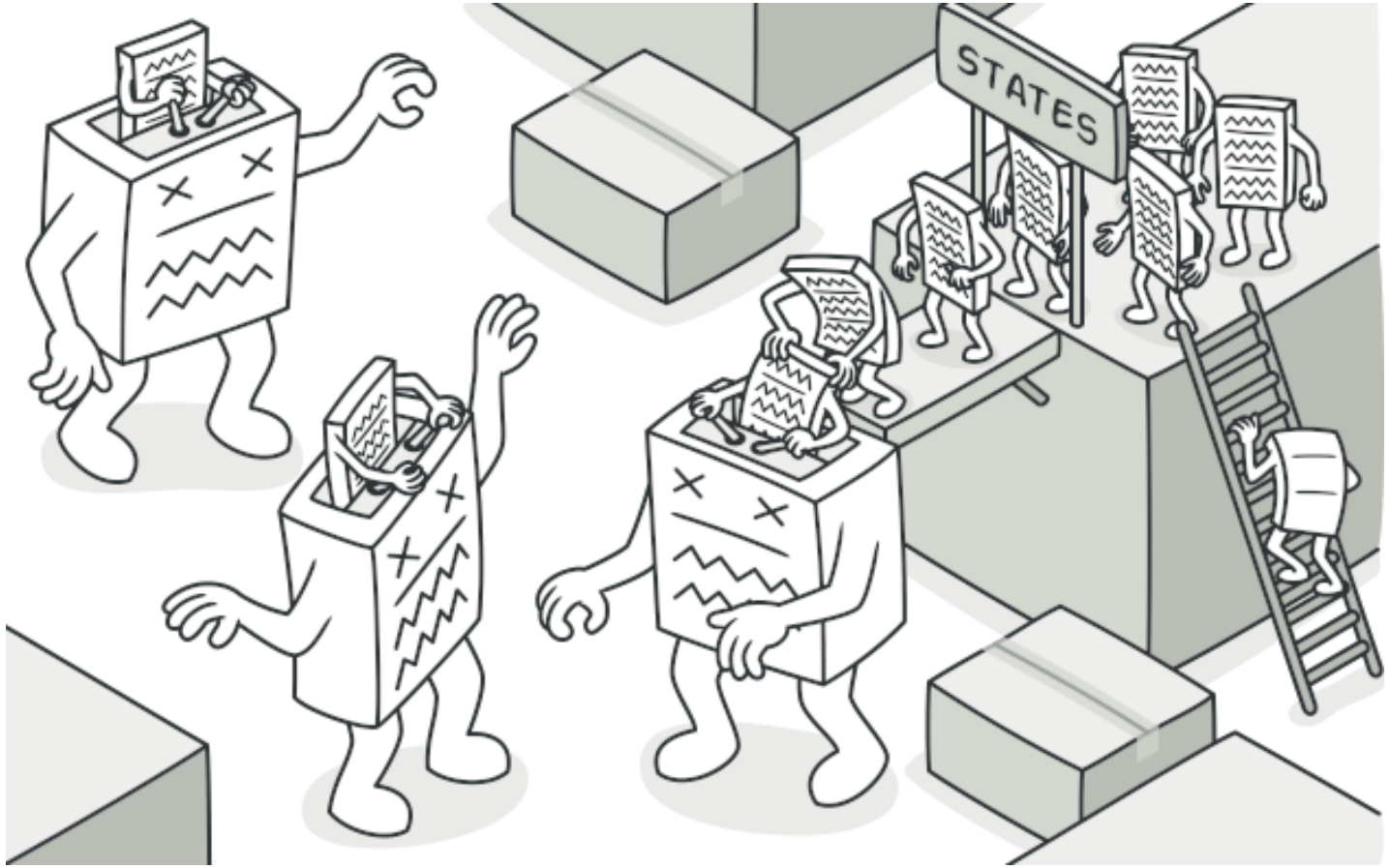
Порождающие

Структурные

Поведенческие

Поведенческие шаблоны связана с распределением обязанностей между объектами. Это такие шаблоны проектирования, определяющие алгоритмы и способы реализации взаимодействия различных объектов и классов.

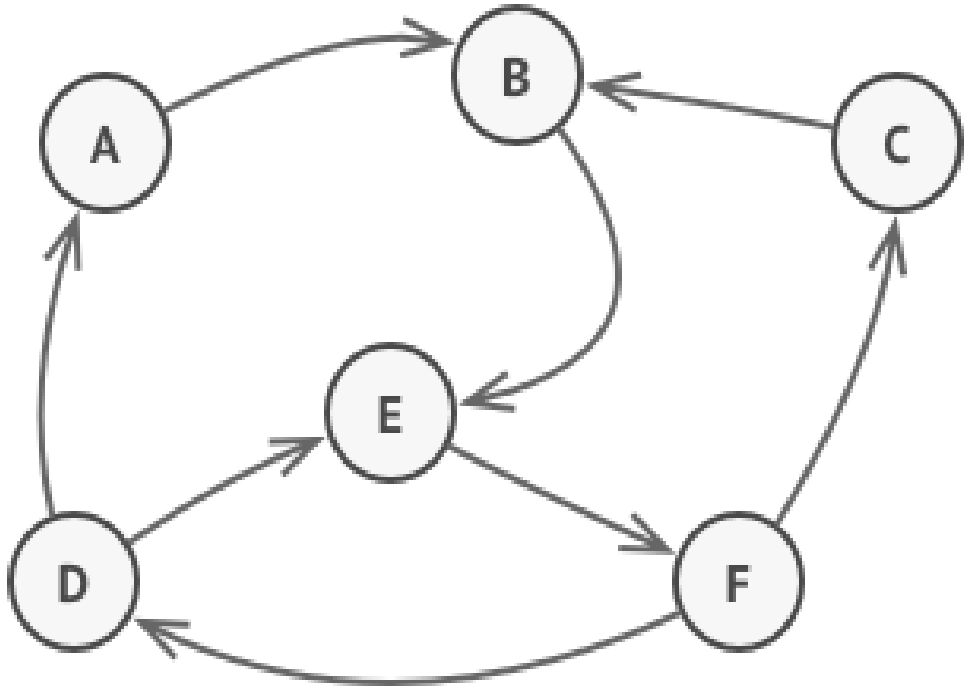
Состояние — это поведенческий паттерн проектирования, который позволяет объектам менять поведение в зависимости от своего состояния.



Проблема ☹

Паттерн **Состояние(State)** невозможно рассматривать в отрыве от концепции **машины состояний.**

Конечный автомат — абстрактный автомат, число возможных внутренних состояний которого конечно.

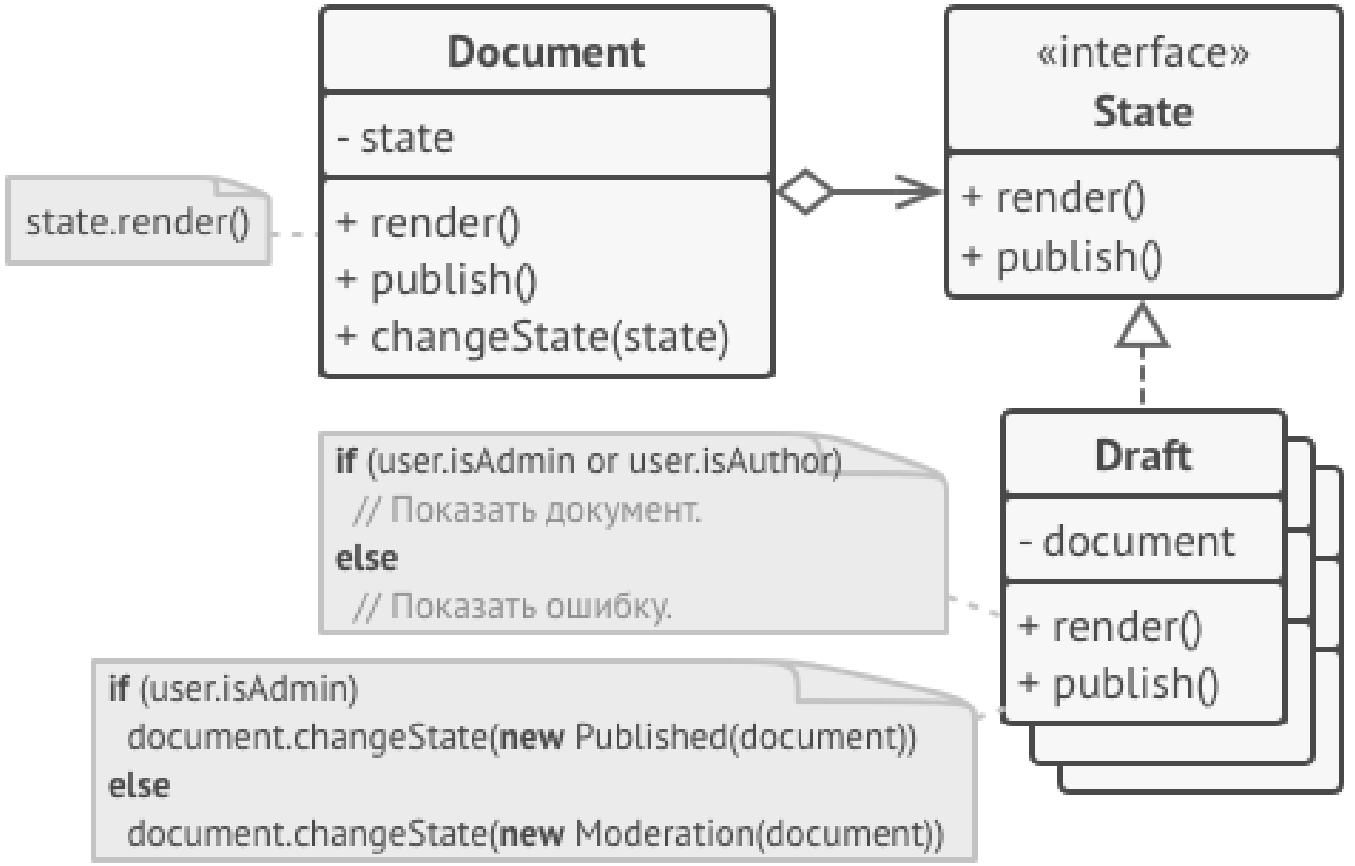


Конечный автомат.

Возможные состояния документа и переходы между ними.

Решение ☺

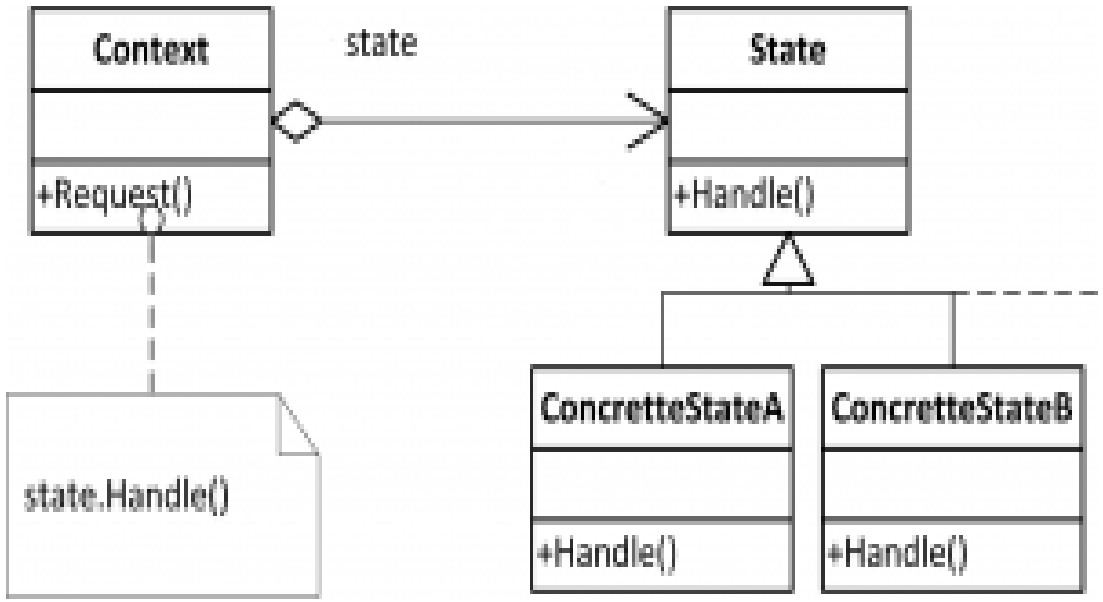
Паттерн **Состояние** предлагает создать отдельные классы для каждого состояния, в котором может пребывать объект, а затем вынести туда поведения, соответствующие этим состояниям.



Документ делегирует работу своему активному объекту-состоянию.

Структура

- **Контекст** хранит ссылку на объект состояния и делегирует ему часть работы, зависящей от состояний.
- **Состояние** описывает общий интерфейс для всех конкретных состояний.
- **Конкретные состояния** реализуют поведения, связанные с определённым состоянием контекста. Иногда приходится создавать целые иерархии классов состояний, чтобы обобщить дублирующий код.



Преимущества и недостатки

+

- Избавляет от множества больших условных операторов машины состояний.
- Концентрирует в одном месте код, связанный с определённым состоянием.
- Упрощает код контекста.

-

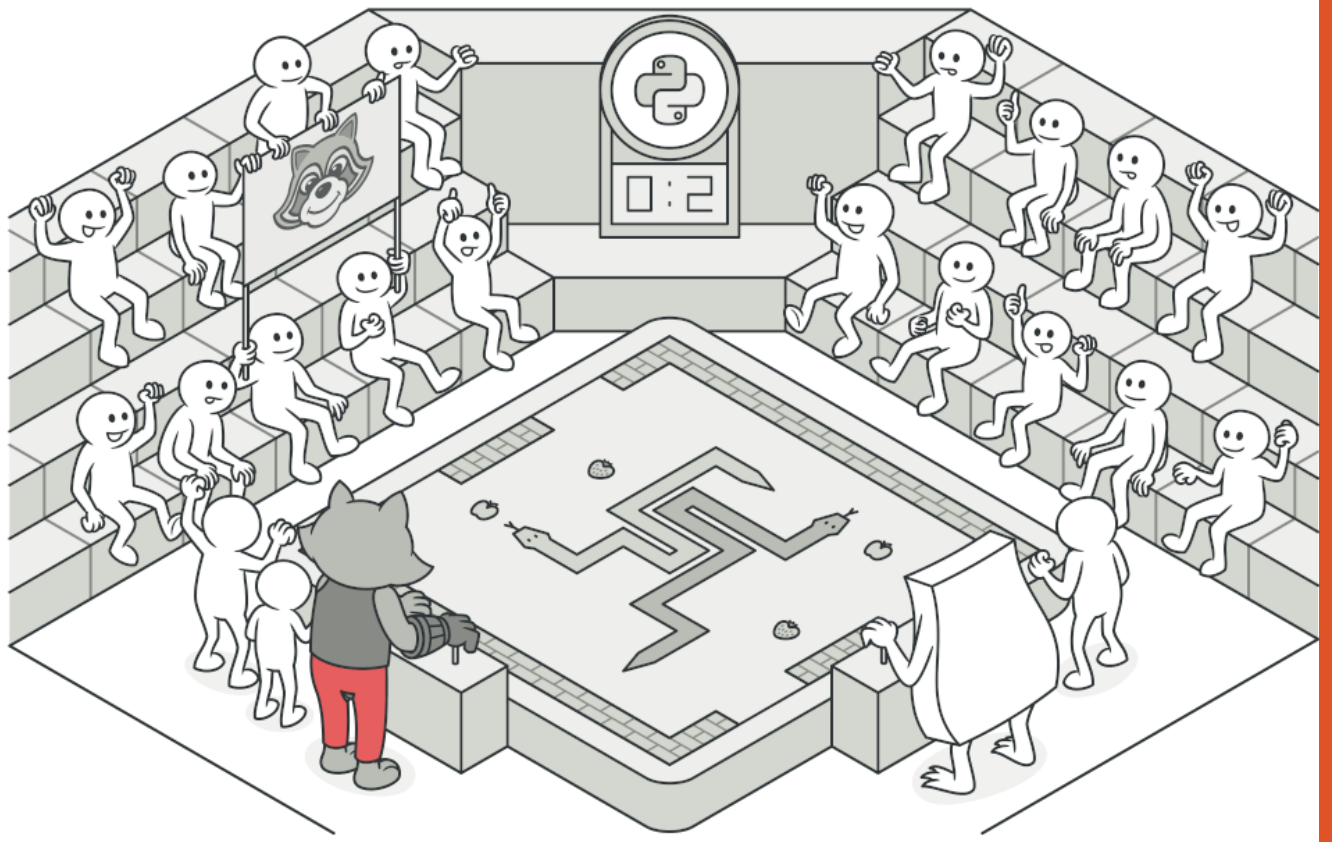
- Может неоправданно усложнить код, если состояний мало и они редко меняются.

Применимость

- Когда у вас есть объект, поведение которого кардинально меняется в зависимости от внутреннего состояния, причём типов состояний много, и их код часто меняется.
- Паттерн предлагает выделить в собственные классы все поля и методы, связанные с определёнными состояниями.
- Когда код класса содержит множество больших, похожих друг на друга, условных операторов, которые выбирают поведения в зависимости от текущих значений полей класса.
- Паттерн предлагает переместить каждую ветку такого условного оператора в собственный класс.
- Паттерн Состояние позволяет реализовать иерархическую машину состояний, базирующуюся на наследовании.

Отношения с другими паттернами

- **Мост**, **Стратегия** и **Состояние** (а также слегка и **Адаптер**) имеют схожие структуры классов — все они построены на принципе «композиции», то есть делегирования работы другим объектам.
- **Состояние** можно рассматривать как надстройку над **Стратегией**. Оба паттерна используют композицию, чтобы менять поведение основного объекта, делегируя работу вложенным объектам-помощникам.



Реализация

https://repl.it/@Ksenia_veh/WideeyedLightheartedRuntimelibrary