

разделить отправителя и получателя в случае, если она уже является частью структуры системы, а один объект из группы может принять на себя обязанность обработать запрос. Данный паттерн повышает гибкость и за счет того, что цепочку можно легко изменить или расширить.

### Резюме

Как правило, паттерны поведения дополняют и усиливают друг друга. Например, класс в цепочке обязанностей, скорее всего, будет содержать хотя бы один шаблонный метод. Он может пользоваться примитивными операциями, чтобы определить, должен ли объект обработать запрос сам, а также в случае необходимости выбрать объект, которому следует переадресовать запрос. Цепочка может применять паттерн команда для представления запросов в виде объектов. Зачастую интерпретатор пользуется паттерном состояние для определения контекстов синтаксического разбора. Иногда Итератор обходит агрегат, а посетитель выполняет операцию для каждого его элемента.

Паттерны поведения хорошо сочетаются и с другими паттернами. Например, система, в которой применяется паттерн компоновщик, время от времени использует посетитель для выполнения операций над компонентами, а также задействует цепочку обязанностей, чтобы обеспечить компонентам доступ к глобальным свойствам через их родителя. Бывает, что в системе применяется и паттерн декоратор для переопределения некоторых свойств частей композиции. А паттерн наблюдатель может связать структуры разных объектов, тогда как паттерн состояние позволит компонентам варьировать свое поведение при изменении состояния. Сама композиция может быть создана с применением строителя и рассматриваться как прототип какой-то другой частью системы.

Хорошо продуманные объектно-ориентированные системы внешне похожи на собрание многочисленных паттернов, но вовсе не потому, что их проектировщики мыслили именно такими категориями. Композиция на уровне паттернов, а не классов или объектов, позволяет добиться той же синергии, но с меньшими усилиями.