

Определение паттерна Компоновщик

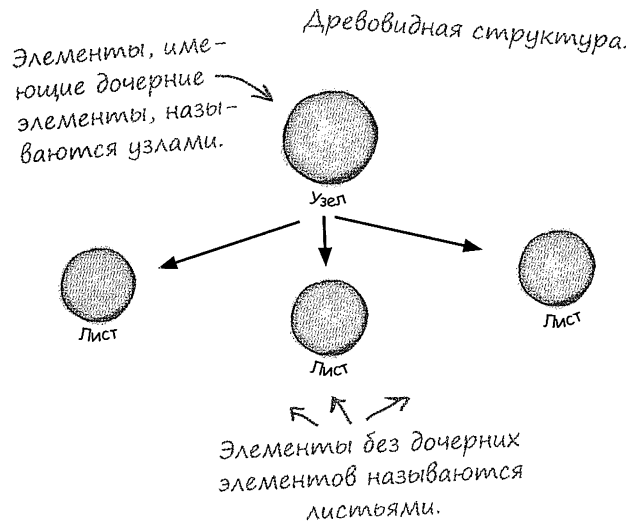
Да, вы не ошиблись, мы собираемся привлечь очередной паттерн для решения этой проблемы. Мы еще не закончили с паттерном Итератор — он остается частью нашего решения. Тем не менее проблема управления меню вышла на новый уровень, недоступный для паттерна Итератор. Итак, мы сделаем шаг назад и решим ее при помощи паттерна Компоновщик.

Не будем ходить вокруг да около — начнем прямо с формального определения:

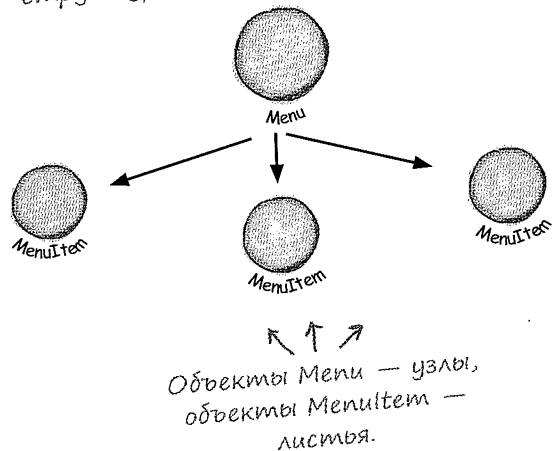
Паттерн Компоновщик объединяет объекты в древовидные структуры для представления иерархий «часть/целое». Компоновщик позволяет клиенту выполнять однородные операции с отдельными объектами и их совокупностями.

Рассмотрим это определение в контексте наших меню: паттерн дает возможность создать древовидную структуру, которая может работать с вложенными группами меню и элементами меню. Размещая меню и элементы в одной структуре, мы создаем иерархию «часть/целое». Иначе говоря, дерево объектов, которое состоит из отдельных частей (меню и элементы меню), но при этом может рассматриваться как единое целое (одно большое «суперменю»).

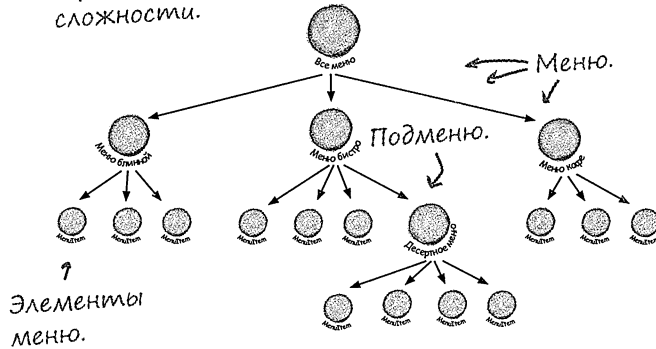
Построив «суперменю», мы можем использовать этот паттерн для того, чтобы «выполнять однородные операции с отдельными объектами и их комбинациями». Что это означает? Если у вас есть древовидная структура из меню, подменю (и, возможно, под-подменю) с элементами, любое меню представляет собой «комбинацию», так как оно может содержать другие меню и команды меню. Отдельными объектами являются только элементы меню — они не могут содержать других объектов. Применение в архитектуре паттерна Компоновщик позволит нам написать простой код, который применяет одну и ту же операцию (например, вывод!) ко всей структуре меню.



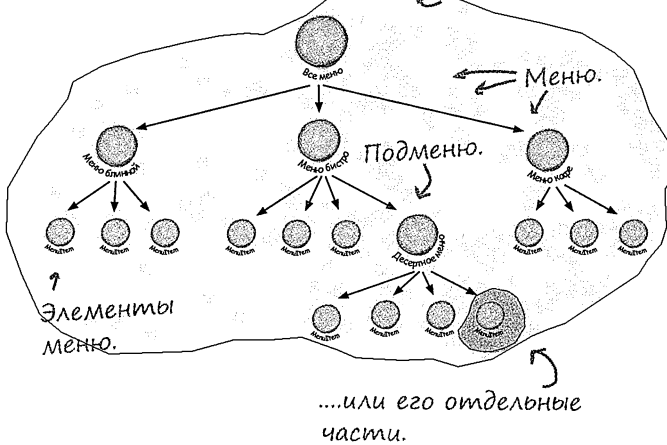
Иерархия Menu и MenuItem может быть представлена древовидной структурой.



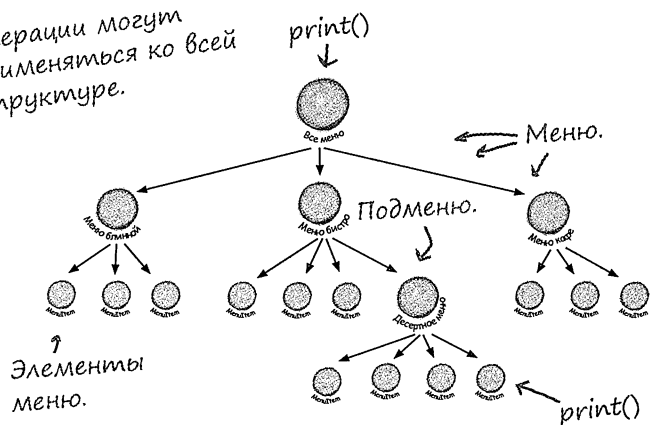
Мы можем создавать
деревья произвольной
сложности.



И обрабатывать их
как единое целое...



Операции могут
применяться ко всей
структуре.

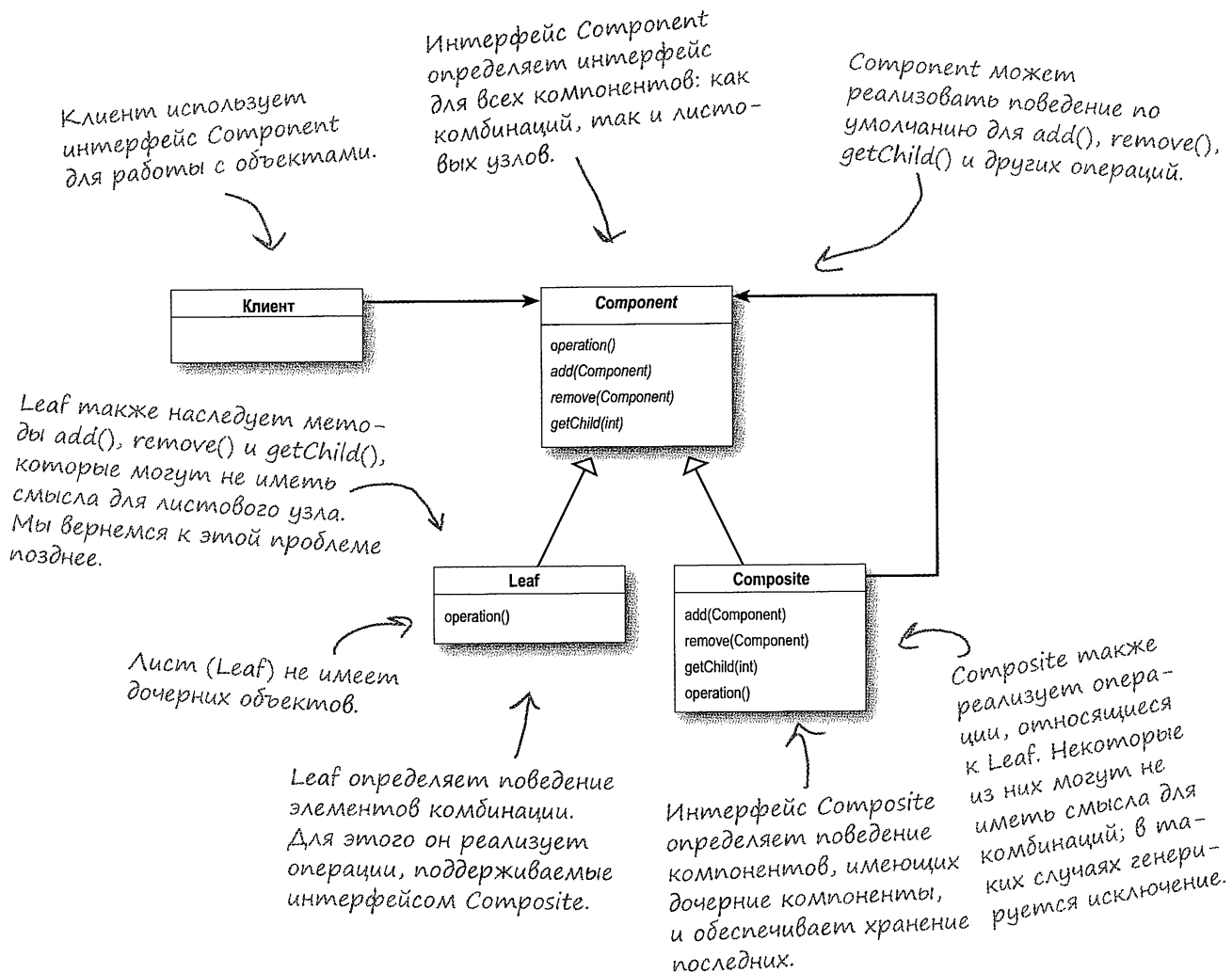


Или к ее частям.

Паттерн Компоновщик
позволяет создавать
древовидные структуры,
узлами которых являются
как комбинации, так
и отдельные объекты.

В такой структуре
одни и те же операции
могут применяться
и к комбинациям,
и к отдельным объектам.
Иначе говоря, во многих
случаях различия между
комбинациями и отдельными
объектами игнорируются.

диаграмма классов паттерна Компоновщик



В: Компоненты, комбинации, деревья? Я уже запутался.

О: Комбинация состоит из компонентов. Компоненты бывают двух видов: комбинации и листовые элементы. Заметили рекурсию? Комбинация содержит дочерние компоненты, которые могут быть другими комбинациями или листьями.

Часто задаваемые вопросы

При такой организации данных образуется древовидная структура (инвертированное дерево), корнем которой является комбинация, а ветвями — комбинации, завершаемые листовыми узлами.

В: А при чем здесь итераторы?

О: Напомню, что мы заново реализуем систему меню на базе нового решения: паттерна Компоновщик. Так что не ждите, что итератор, как по волшебству, превратится в комбинацию. Тем не менее эти два паттерна хорошо работают в сочетании друг с другом.

Вскоре мы рассмотрим пару возможных применений итераторов в реализации комбинаций.

Проектирование меню с использованием паттерна Компоновщик

Как же нам применить паттерн Компоновщик при проектировании системы меню? Для начала необходимо определить интерфейс компонента; этот интерфейс, общий для меню и элементов меню, позволяет выполнять с ними однородные операции. Другими словами, *один и тот же* метод вызывается как для меню, так и для их элементов.

Возможно, вызов некоторых методов для меню или элементов меню *не имеет смысла*, но с этим мы разберемся позднее (притом совсем скоро). А пока в общих чертах посмотрим, как система меню укладывается в структуру паттерна Компоновщик:

