Java Design Patterns

Iterator

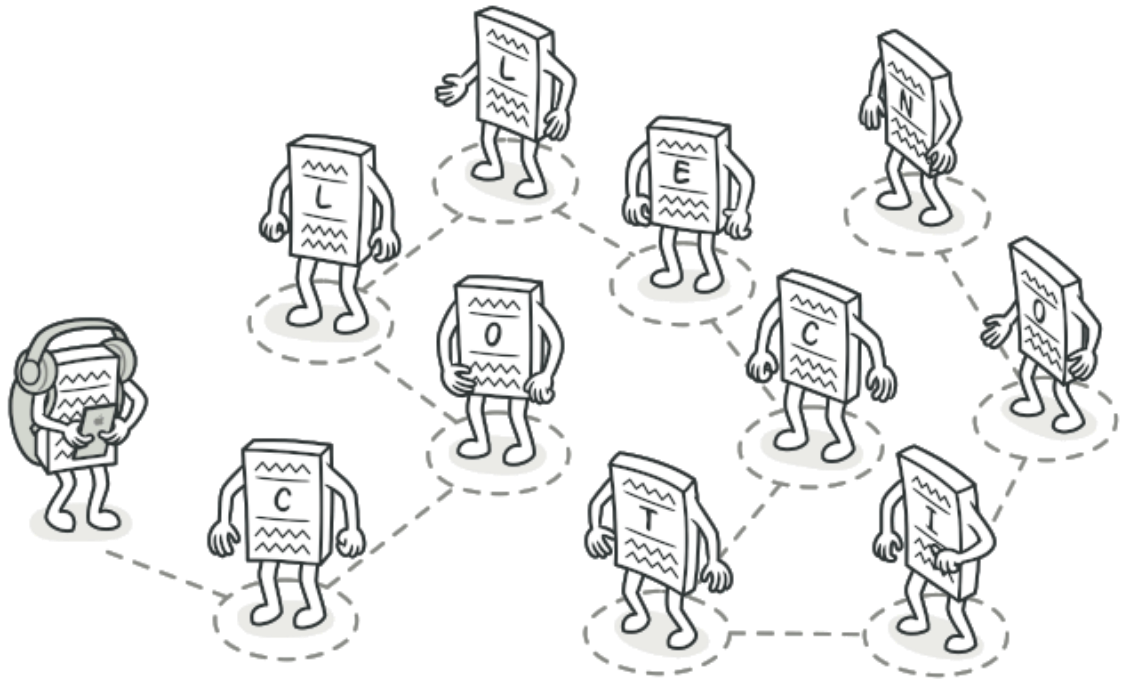
Тема

Iterator

Суть паттерна

Итератор

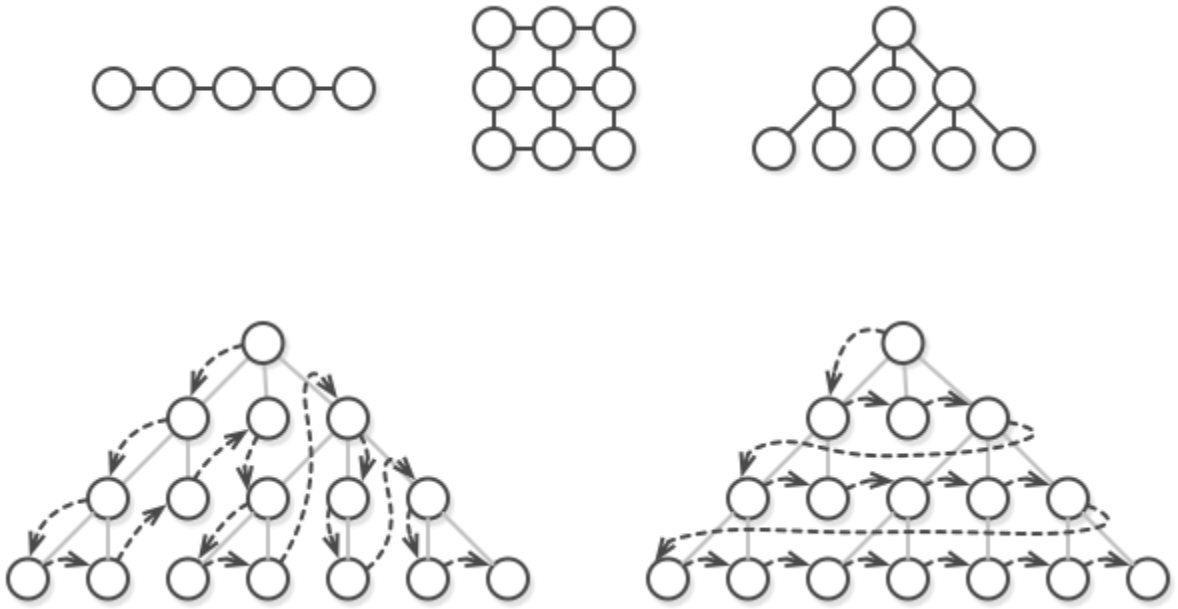
Итератор — это поведенческий паттерн проектирования, который даёт возможность последовательно обходить элементы составных объектов, не раскрывая их внутреннего представления.



Проблема

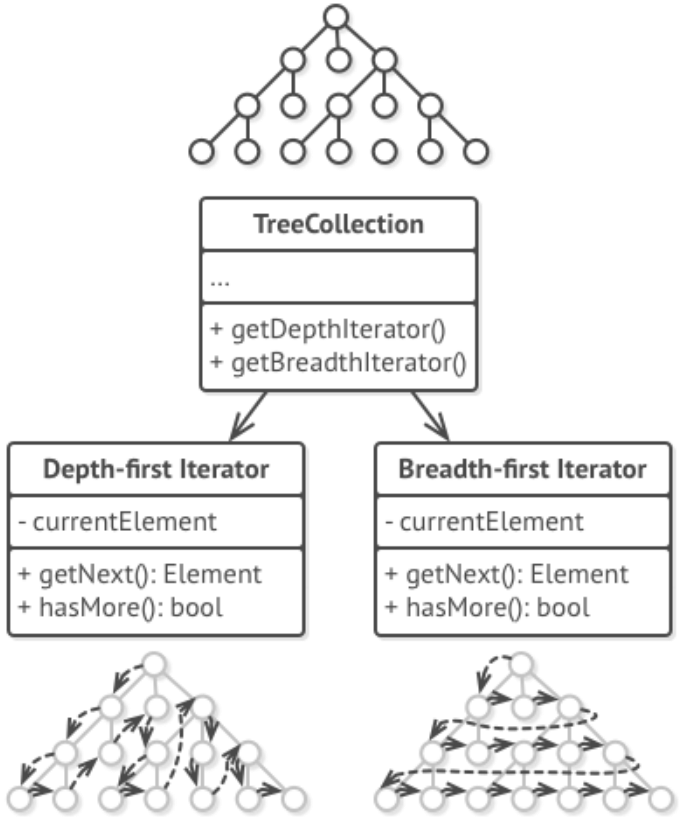
Постановка задачи

Коллекции — самая частая структура данных, которую вы можете встретить в программировании. Это набор объектов, собранный в одну кучу по каким-то причинам.



Решение задачи

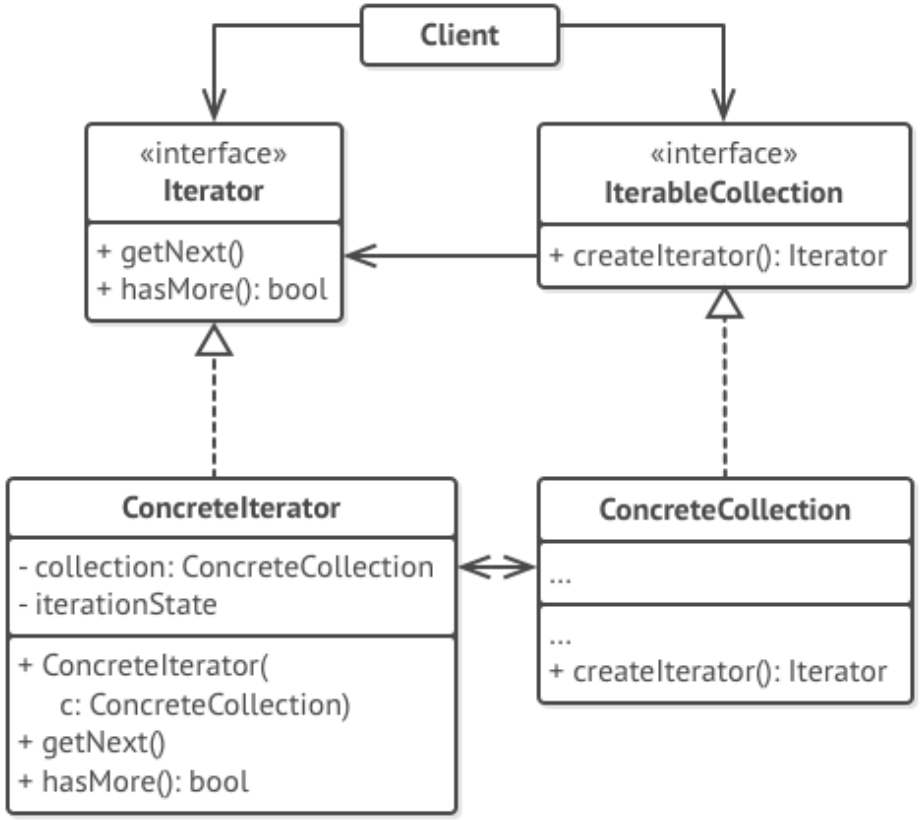
Идея паттерна Итератор в том, чтобы вынести поведение обхода коллекции из самой коллекции в отдельный класс.



Структура

Структура паттерна

1. Итератор описывает интерфейс для доступа и обхода элементов коллекции.
2. Конкретный итератор реализует алгоритм обхода какой-то конкретной коллекции. Объект итератора должен сам отслеживать текущую позицию при обходе коллекции, чтобы отдельные итераторы могли обходить одну и ту же коллекцию независимо.
3. Коллекция описывает интерфейс получения итератора из коллекции. Как мы уже говорили, коллекции не всегда являются списком. Это может быть и база данных, и удалённое API, и даже дерево Компоновщика.
4. Конкретная коллекция возвращает новый экземпляр определённого конкретного итератора, связав его с текущим объектом коллекции.
5. Клиент работает со всеми объектами через интерфейсы коллекции и итератора. Так клиентский код не зависит от конкретного класса итератора, что позволяет применять различные итераторы, не изменяя существующий код программы.



Применимость

Применение паттерна

1. Когда у вас есть сложная структура данных, и вы хотите скрыть от клиента детали её реализации (из-за сложности или вопросов безопасности).
2. Когда вам нужно иметь несколько вариантов обхода одной и той же структуры данных.
3. Когда вам хочется иметь единый интерфейс обхода различных структур данных.

Шаги реализации

Алгоритм реализации паттерна

1. Создайте интерфейс итераторов. В качестве минимума, вам понадобится операция получения следующего элемента. Но для удобства можно предусмотреть и другие методы, например, для получения предыдущего элемента, текущей позиции, проверки окончания обхода и прочих.
2. Создайте интерфейс коллекции и опишите в нём метод получения итератора. Важно, чтобы его сигнатура возвращала общий интерфейс итераторов, а не один из конкретных итераторов.
3. Создайте классы конкретных итераторов для тех коллекций, которые нужно обходить с помощью паттерна. Итератор должен быть привязан только к одному объекту коллекции. Обычно эта связь устанавливается через конструктор.
4. Реализуйте методы получения итератора в конкретных классах коллекций. Они должны создавать новый итератор того класса, который способен работать с данным типом коллекции. Коллекция должна передавать собственную ссылку в созданный итератор.
5. В клиентском коде и в классах коллекций не должно остаться кода обхода элементов. Клиент должен получать новый итератор из объекта коллекции каждый раз, когда ему нужно перебрать её элементы.

Преимущества и недостатки

Плюсы и недостатки

Плюсы:

- Упрощает классы хранения данных.
- Позволяет реализовать различные способы обхода структуры данных.
- Позволяет одновременно перемещаться по структуре данных в разные стороны.

Минусы:

- Не оправдан, если можно обойтись простым циклом.

Отношение с другими паттернами

- Вы можете обходить дерево Компоновщика, используя Итератор.
- Фабричный метод можно использовать вместе с Итератором, чтобы подклассы коллекций могли создавать подходящие им итераторы.
- Снимок можно использовать вместе с Итератором, чтобы сохранить текущее состояние обхода структуры данных и вернуться к нему в будущем, если потребуется.
- Посетитель можно использовать совместно с Итератором. *Итератор* будет отвечать за обход структуры данных, а *Посетитель* — за выполнение действий над каждым её компонентом.

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association