

Iterador (Iterator)

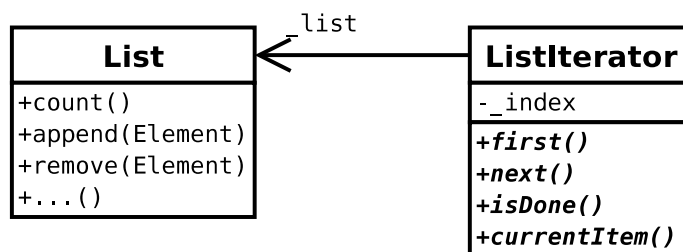
- *Patrón de Comportamiento*

- *Propósito*

Mecanismo de acceso a los elementos que constituyen una estructura de datos sin exponer su representación interna

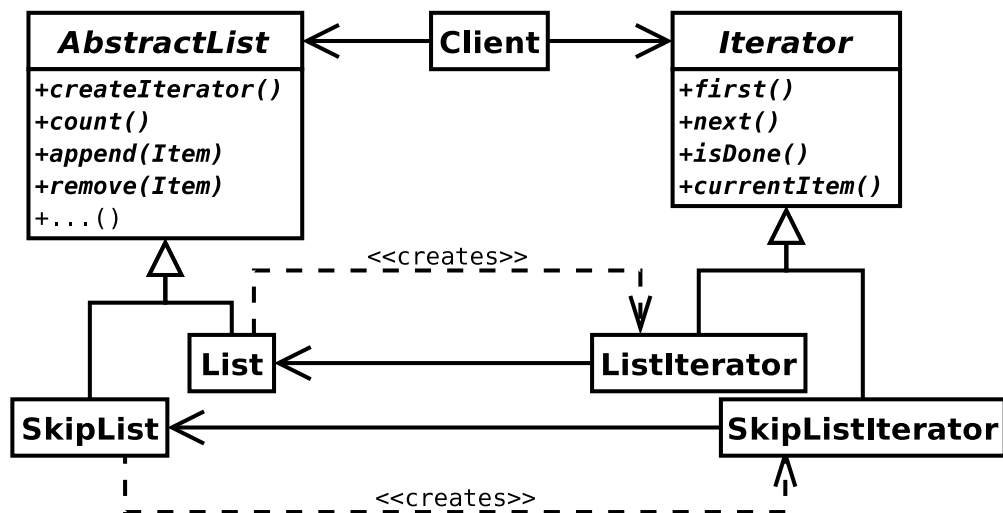
- *Motivación*

- Se desea acceder a los elementos de un contenedor de objetos (por ejemplo, una lista enlazada) sin exponer su representación interna
- Solución: Añadir métodos que permitan recorrer la estructura sin referenciar explícitamente su representación
- Problema: Puede necesitarse más de una forma de recorrer la estructura: se complica la interfaz de la clase
- Problema: Si se necesita un recorrido no previsto es necesario modificar la clase
- Solución: Mover la responsabilidad del recorrido a un objeto *iterador*



■ Motivación (cont.)

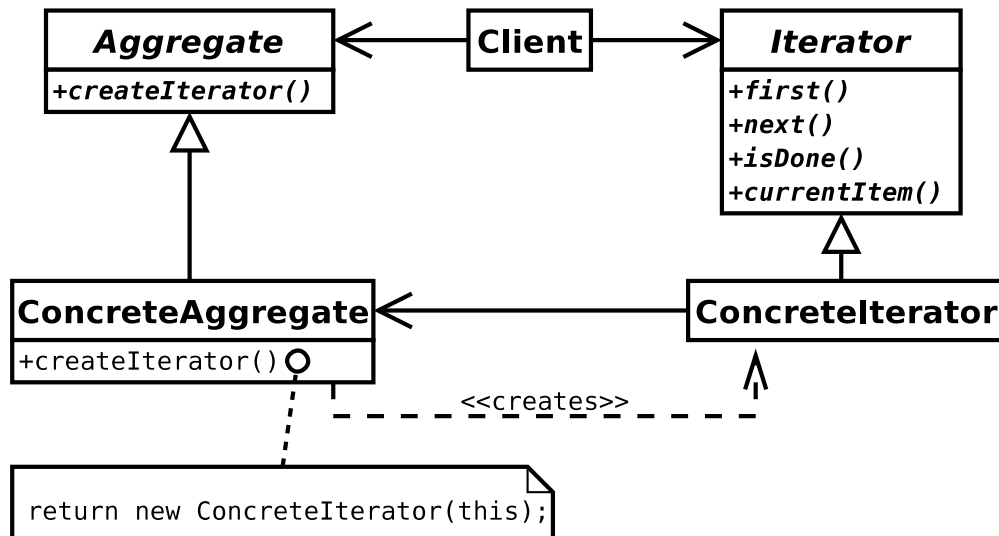
- No tiene por qué limitarse a recorrer la estructura (por ejemplo, filtrado)
- Problema: Los clientes de la estructura de datos deben conocer la estructura concreta para crear un iterador
- Solución: Generalizar los distintos iteradores en una superclase y dotar a las estructuras de datos de un *método de fabricación* que cree un iterador concreto



■ Aplicabilidad

- Acceso al contenido de una estructura sin exponer su representación interna
- Distintos tipos de recorrido sobre la estructura
- Interfaz uniforme para recorrer diferentes tipos de estructuras

■ *Estructura*



■ *Participantes*

- Iterador (Iterator)
 - Define la interfaz para recorrer el agregado de elementos
- Iterador Concreto (ConcreteIterator)
 - Implementa la interfaz propuesta por el Iterador
 - Mantiene la posición actual en el recorrido de la estructura
- Agregado (Aggregate)
 - Define la interfaz para el método de fabricación de iteradores
- Agregado Concreto (ConcreteAggregate)
 - Implementa la estructura de datos
 - Implementa el método de fabricación de iteradores que crea un iterador específico para su estructura

■ *Colaboraciones*

- Un iterador concreto mantiene la posición actual dentro de la estructura de datos e interactúa con ésta para calcular el siguiente elemento en el recorrido

■ *Consecuencias*

- Distintos tipos de recorrido
- Simplificación de la interfaz del agregado
- Más de un recorrido simultáneo

■ *Implementación*

- Control de la iteración
 - iterador externo vs. iterador interno
- Definición del recorrido
- Robustez del iterador ante cambios en la estructura
- Operaciones adicionales en el iterador
- Iteradores pueden necesitar acceso privilegiado
- Iteradores para Composición
- Iteradores nulos