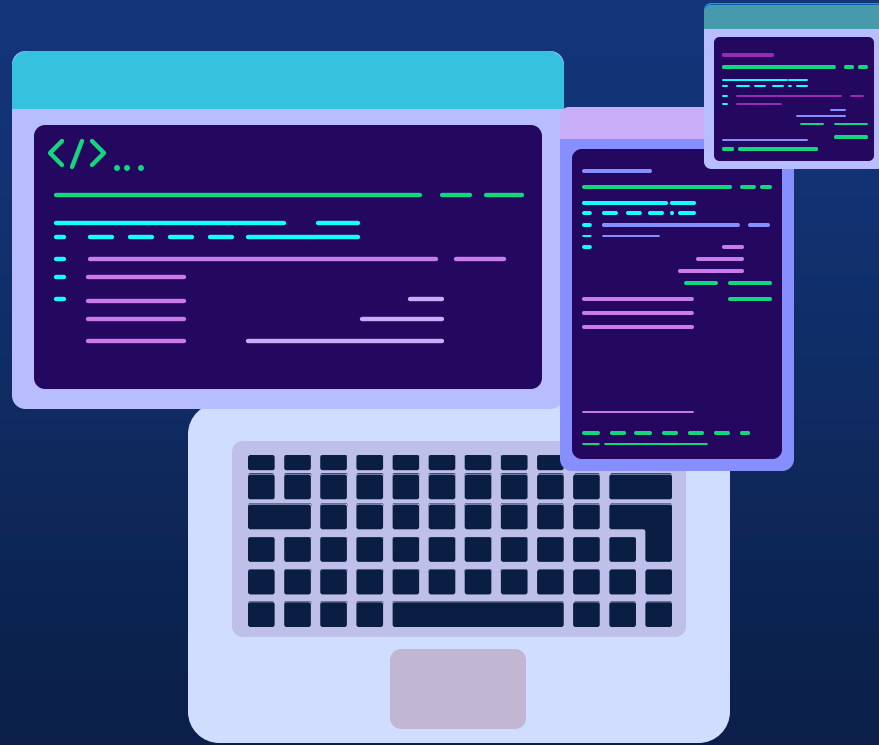


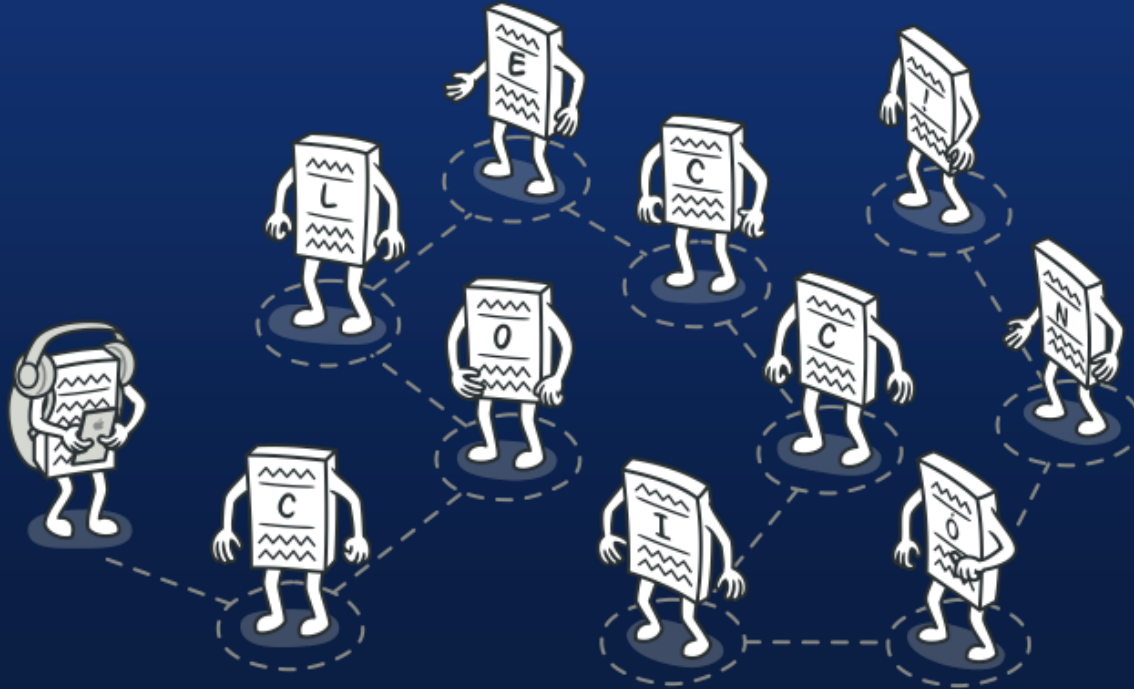
Patrón de diseño: Iterador

Leonardo Martin
Bacelis Bautista



Iterador

Iterator es un patrón de diseño de comportamiento que te permite recorrer elementos de una colección sin exponer su representación subyacente (lista, pila, árbol, etc.)





Problema

Las colecciones son de los tipos de datos más utilizados en programación. Sin embargo, una colección tan solo es un contenedor para un grupo de objetos.



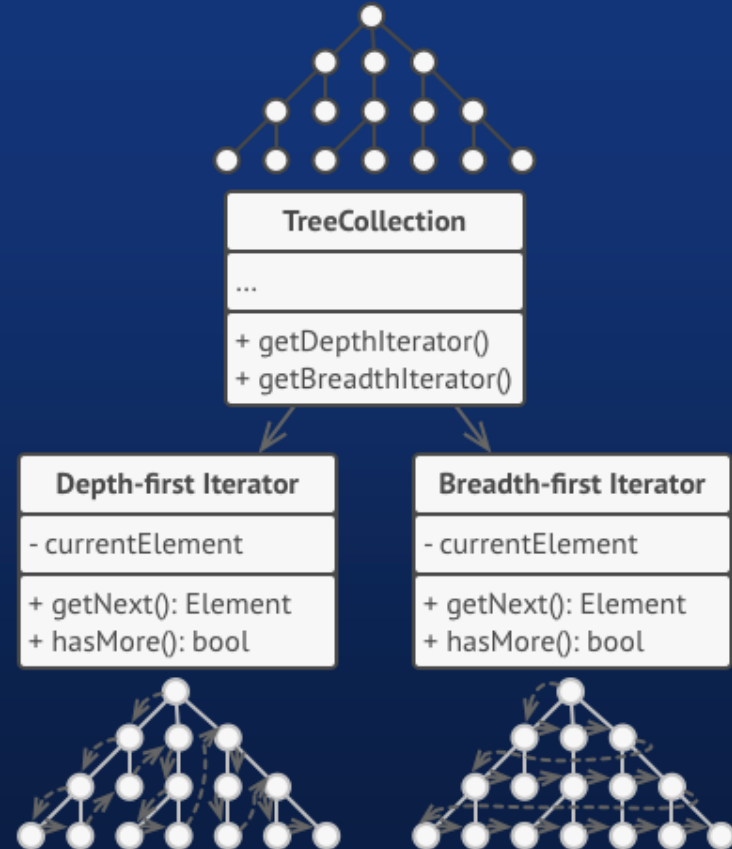
La mayoría de las colecciones almacena sus elementos en simples listas, pero algunas de ellas se basan en pilas, árboles, grafos y otras estructuras complejas de datos.



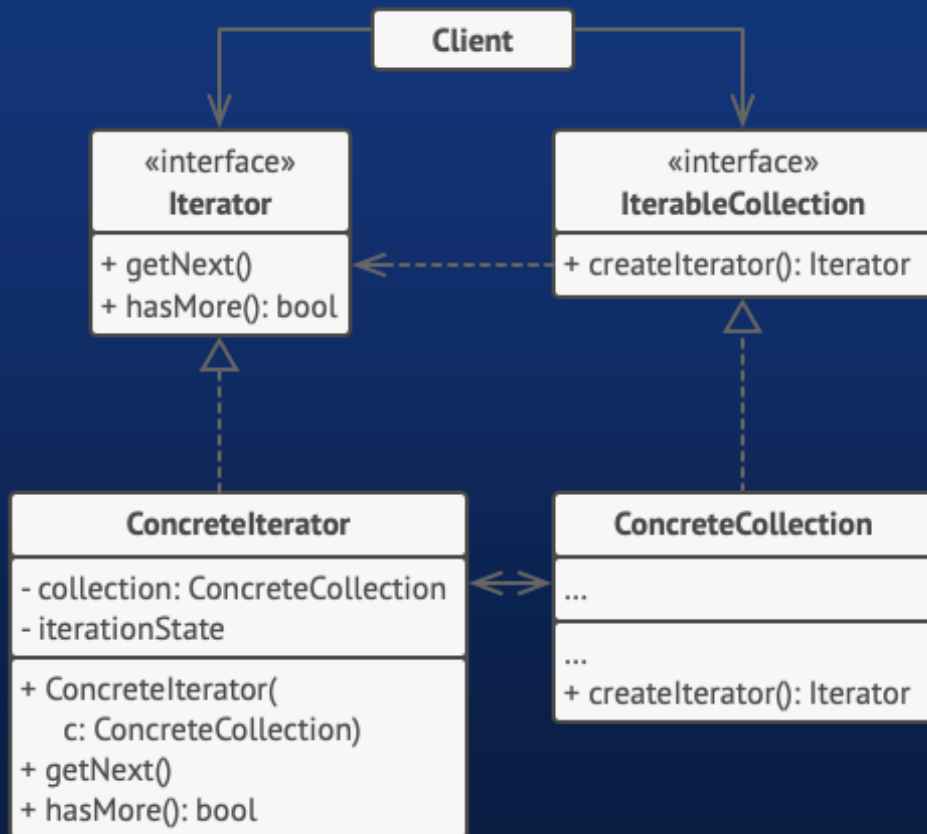


Solución

La idea central del patrón Iterator es extraer el comportamiento de recorrido de una colección y colocarlo en un objeto independiente llamado iterador.



Estructura





Pros

- + Se puede acceder a cada colección de manera uniforme.
- + Múltiples iteraciones son posibles al mismo tiempo
- + El algoritmo de iteración puede variar

Contras

- Menor eficiencia
- No se garantiza la robustez (inserciones, supresiones)
- "Oculta" la estructura de datos subyacente





Usos

- Muchos lenguajes de programación utilizan iteradores para recorrer colecciones (C++, C#, Java, Python, ...).
- For Loop también utiliza Iterator (implícitamente)
- Enumeradores y Generadores también son variantes de iteradores

