

Contenedores e iteradores

1. ¿Qué es un contenedor en C++?

Un contenedor es una estructura de datos que almacena y organiza otros objetos. Los contenedores proporcionan formas convenientes de almacenar colecciones de elementos, como arrays, listas, colas, pilas, conjuntos y mapas.

2. Tipos de contenedores en C++, incluyendo: vector, list, deque, array, map, etc. En qué casos usar cada uno?

Vector: Es útil cuando necesitas un array que pueda crecer o reducir su tamaño de manera eficiente.

List: Es útil cuando necesitas realizar muchas inserciones y eliminaciones en el medio de la lista

Deque: Es útil cuando necesitas insertar o eliminar elementos al principio o al final del contenedor de manera eficiente, pero también quieres la capacidad de acceso aleatorio a los elementos.

Array: Es útil cuando necesitas un array con un tamaño fijo conocido en tiempo de compilación y un acceso rápido a los elementos.

Map: Es útil cuando necesitas buscar elementos rápidamente por su clave y mantener un orden específico en las claves.

3. Propósito de los contenedores en C++

Los contenedores tienen diversos propósitos, algunos de los más importantes son:

- Almacenamiento de datos
- Abstracción de estructuras de datos
- Flexibilidad y eficiencia

4. Características comunes de los contenedores: tamaño, forma de almacenamiento (contigua o aleatoria), modificabilidad.

- **Tamaño:** Los contenedores pueden ser de tamaño variable (vectores, listas, deque) o de tamaño fijo (array).
- **Forma de almacenamiento:** Los contenedores pueden almacenar elementos de forma contigua en la memoria o de forma no contigua. Los contenedores que almacenan elementos de forma contigua tienen la ventaja de un acceso más rápido a los elementos mediante

punteros o iteradores, mientras que los contenedores no contiguos pueden permitir una inserción y eliminación más eficiente de elementos en ciertos casos.

- **Modificabilidad:** La capacidad de modificar los elementos almacenados en un contenedor varía según el tipo de contenedor. Algunos contenedores permiten modificar fácilmente los elementos almacenados mediante operaciones de inserción, eliminación y modificación. Otros tienen restricciones en la modificación de elementos para mantener ciertas propiedades.

5. ¿Qué es un iterador?

Es un objeto que permite recorrer secuencialmente los elementos de un contenedor o una secuencia. Los iteradores proporcionan una forma uniforme de acceder a los elementos de un contenedor sin exponer la estructura interna del contenedor.

6. Uso de los iteradores con los contenedores estándar en C++: recorrido, acceso a elementos, etc.

- **Recorrido de elementos:** Puedes recorrer los elementos de un contenedor utilizando un bucle for junto con un iterador.
- **Acceso a elementos:** Puedes acceder a los elementos de un contenedor utilizando un iterador (*it se utiliza para acceder y modificar cada elemento de la lista).
- **Inserción y eliminación de elementos:** Los iteradores también se pueden utilizar para insertar y eliminar elementos en contenedores que admiten estas operaciones, como vectores y listas (vec.erase(it) elimina el elemento apuntado por el iterador)

7. Funciones y métodos de los contenedores y iteradores en C++: push_back(), pop_back(), begin(), end(), find().

- **push_back():** Agrega un elemento al final de un contenedor secuencial
- **pop_back():** Elimina el último elemento de un contenedor secuencial.
- **begin():** Devuelve un iterador al primer elemento del contenedor.
- **end():** Devuelve un iterador al elemento después del último elemento del contenedor. Este iterador no apunta a ningún elemento válido y se utiliza como señal de finalización en bucles.
- **find():** Busca un elemento en un contenedor que admite la operación de búsqueda.