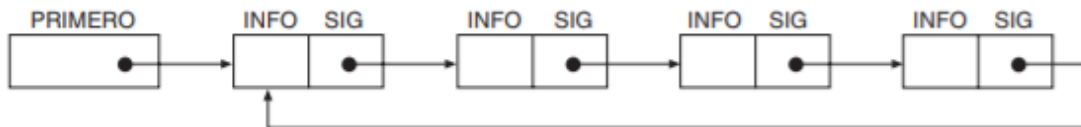


Listas

Listas Enlazadas Circulares

En las listas lineales simples o en las dobles siempre hay un primer nodo (cabeza) y un último nodo (cola). Una lista circular, por propia naturaleza, no tiene ni principio ni fin. Sin embargo, resulta útil establecer un nodo a partir del cual se acceda a la lista y así poder acceder a sus nodos.

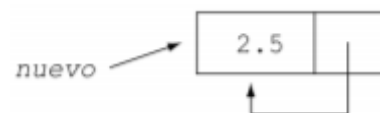


Las operaciones que se realizan sobre una lista circular son similares a las operaciones sobre listas lineales, teniendo en cuenta que no hay primero ni último nodo, aunque sí un nodo de acceso a la lista. Estas operaciones permiten construir el TAD Lista Circular y su funcionalidad es la siguiente:

- Inicialización o creación.
- Inserción de elementos en una lista circular.
- Eliminación de elementos de una lista circular.
- Búsqueda de elementos de una lista circular.
- Recorrido de cada uno de los nodos de una lista circular.
- Verificación de lista vacía.

La construcción de una lista circular se puede hacer con enlace simple o enlace doble. Se declara la clase Nodo, con el campo dato y enlace, y la clase Lista Circular con el puntero de acceso a la lista, junto a los métodos que implementan las operaciones. Los elementos de la lista pueden ser de cualquier tipo, se puede abstraer su tipo en otra clase, por ejemplo Elemento; con el fin de simplificar, se supone un tipo conocido.

El constructor de la clase Nodo varía respecto al de las listas no circulares, ya que el campo referencia enlace, en vez de quedar a null, se inicializa para que apunte al mismo nodo, de tal forma que queda como lista circular de un solo nodo.



Las listas circulares presentan las siguientes ventajas respecto de las listas enlazadas simples:

- Cada nodo de una lista circular es accesible desde cualquier otro nodo de ella. Es decir, dado un nodo se puede recorrer toda la lista completa. En una lista enlazada de forma simple sólo es posible recorrerla por completo si se parte de su primer nodo.
- Las operaciones de concatenación y división de listas son más eficaces con listas circulares.

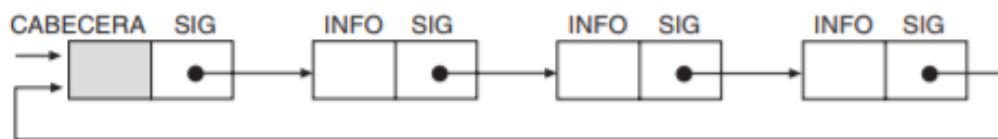
Los inconvenientes, por el contrario, son:

- Se pueden producir lazos o bucles infinitos. Una forma de evitar estos bucles infinitos es disponer de un nodo especial que se encuentre permanentemente asociado a la existencia de la lista circular. Este nodo se denomina cabecera de la lista.

El nodo cabecera puede diferenciarse de los otros nodos en una de las dos formas siguientes:

- Puede tener un valor especial en su campo INFO que no es válido como datos de otros elementos.
- Puede tener un indicador o bandera (flag) que señale cuando es nodo cabecera.

El campo de la información del nodo cabecera no se utiliza, lo que se señala con el sombreado de dicho campo.



Bibliografía

Luis Joyanes Aguilar. (2003). Fundamentos de programación – Algoritmos, Estructuras de datos y Objetos., de Mc Graw Hill

Luis Joyanes Aguilar, Ignacio Zahonero Martínez (2008). Estructuras de datos en Java, de Mc Graw Hill