**Estructuras enlazadas**

**Idea en general de la estructura**

Hasta ahora, todas las estructuras que vimos se apoyaban de los arreglos; estructura fija, uniforme, se puede mover por los índices, etc.

¿Qué es una **estructura enlazada**? Más de una variable simple que tiene la particularidad de estar formada por **nodos**.

¿Qué es un **nodo**? Está formado por dos partes. La primera, donde yo guardo el dato (alguna parte que me permita guardar la información), la segunda es el tipo Nodo. Finalmente, nodo es un objeto. Lo que guardamos generalmente de un objeto es la dirección del objeto. Aquí pasa lo mismo. Este segundo elemento lo que tiene es la dirección de **otro nodo**. Entonces, una parte guarda el dato/información, la segunda guarda la dirección de otro nodo.

Entonces, cuando yo hago una colección de **nodos** que se conectan entre si menos el último nodo (no apunta a nadie; ya llegué al final) a esto se le llama una **estructura enlazada**.

Cuando una variable con memoria a la que nadie se le referencia, entonces el recolector de basura de Java se lo lleva.

Al final, para crear el objeto o base de una estructura enlazada, primero se tiene que construir un método.

**Construcción de un nodo**

¿Qué pasa con el inicio? Debo tener una variable que no se olvidé nunca que es el primero. **Pierdo la referencia del primer nodo, pierdo todo**.

Vamos a tener que pensar en crear un iterador que me permita moverme a través de una estructura enlazada sin que yo desde afuera sepa dónde está cada objeto.

**UML**

|  |
| --- |
| Nodo |
| * dato T * direccion Nodo |
| + Nodo()  + Nodo(T)  + get  + set  + toString(): String |