

Estructuras de datos

Clase práctica 3



Contenido

- Listas enlazadas

Material elaborado por: Julián Moreno

Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión

Listas enlazadas

Para trabajar con listas, Java ya tiene implementada la clase *List* de la librería *java.util.** la cual posee varias implementaciones, siendo de nuestro interés *LinkedList*.

Además, los métodos que nos interesan son prácticamente los mismos que los de la clase *Vector*:

| | |
|--------------------------|------------------------------------------------------------------|
| <code>addFirst(e)</code> | Ingresa el elemento <i>e</i> al inicio de la lista |
| <code>add(e)</code> | Ingresa el elemento <i>e</i> al final de la lista |
| <code>add(i,e)</code> | Ingresa el elemento <i>e</i> en la posición <i>i</i> de la lista |
| <code>clear()</code> | Borra todos los elementos |
| <code>get(i)</code> | Devuelve el elemento en la posición <i>i</i> de la lista |
| <code>getFirst()</code> | Devuelve el primer elemento de la lista |
| <code>getLast()</code> | Devuelve el último elemento de la lista |

Listas enlazadas

| | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>isEmpty()</code> | Devuelve verdadero si la lista está vacío |
| <code>indexOf(e)</code> | Devuelve la posición de la primera ocurrencia del elemento <i>e</i> dentro de la lista, o -1 si no está |
| <code>lastIndexOf(e)</code> | Devuelve la posición de la última ocurrencia del elemento <i>e</i> dentro de la lista, o -1 si no está |
| <code>remove(i)</code> | Borra el elemento en la posición <i>i</i> de la lista |
| <code>removeFirst()</code> | Borra el primer elemento de la lista |
| <code>removeLast()</code> | Borra el último elemento de la lista |
| <code>remove(e)</code> | Borra la primera ocurrencia del elemento <i>e</i> dentro de la lista. Devuelve <i>true</i> si el elemento se encuentra. |
| <code>set(i,e)</code> | Reemplaza el elemento en la posición <i>i</i> por <i>e</i> |
| <code>size()</code> | Devuelve la cantidad de elementos en la lista |

Listas enlazadas

Veamos un ejemplo sencillo:

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);
        List<long> numeros = new LinkedList<long>();
        long x;
        String opcion;
        do{
            System.out.println("Numero a ingresar en la lista: ");
            x = entrada.nextLong();
            numeros.add(x);
            System.out.println("Desea ingresar otro número? (s/n): ");
            opcion = entrada.next();
        } while (opcion.compareTo("s") == 0);
        System.out.println("Numeros ingresados\n" + numeros);

        //...
```

Listas enlazadas

```
//...
do{
    System.out.println("Numero a buscar: ");
    x = entrada.nextLong();
    if (numeros.indexOf(x) >= 0){
        System.out.println(x + " si esta en la lista");
    }
    else{
        System.out.println(x + " no esta en la lista");
    }
    System.out.println("Desea buscar otro número? (s/n): ");
    opcion = entrada.next();
} while (opcion.compareTo("s") == 0);

}
```

Ejercicios en CPP

La meta del día es realizar los ejercicios "Inventario dinámico" y "Despidos masivos"