

# Práctica 6

## Programación Orientada a Objetos: Desarrollo

### Listas enlazadas

#### ACTIVIDADES

Estudiar y analizar el siguiente código que aparece en el recuadro.

Programa en java:

```
class Link {  
  
    public int iData; // dato  
  
    public double dData; // dato  
  
    public Link next; // referencia al siguiente link  
  
    public Link(int id, double dd) { // constructor  
  
        iData = id; // initialize dato  
  
        dData = dd; // ('next' es automaticamente // puesto null)  
  
    }  
  
    public void displayLink() { // display a nosotros mismo  
  
        System.out.print("{ " + iData + " , " + dData + "}");  
  
    }  
}
```

```
}
```

```
class LinkList {
```

```
    private Link first; // referencia al first link de la lista
```

```
    public void LinkList() { // constructor
```

```
        first = null; // no hay items en list aun
```

```
    }
```

```
    public boolean isEmpty() { // true si lista esta empty
```

```
        return (first==null);
```

```
    }
```

```
    ... // otros metodos van aqui
```

```
}
```

```
// insertar al principio de la list
```

```
public void insertFirst(int id, double dd) {
```

```
    // hace un nuevo link
```

```
    Link newLink = new Link(id, dd);
```

```
    newLink.next = first; // newLink --> old first
```

```
    first = newLink; // first --> newLink
```

```
}
```

```
public Link deleteFirst() { // delete first item // (assumes list not empty)

    Link temp = first; // save reference to link

    first = first.next; // delete it: first-->old next

    return temp; // return deleted link

}
```

```
public void displayList() {

    System.out.print("List (first-->last): ");

    Link current = first; // start at beginning of list

    while(current != null) // until end of list,

    {

        current.displayLink(); // print data

        current = current.next; // move to next link

    }

    System.out.println("");

}
```

```
class Link {

    public int iData; // data item (key)

    public double dData; // data item

    public Link next; // next link in list

    public Link(int id, double dd) { // constructor
```

```

        iData = id; // initialize data

        dData = dd; // ('next' is automatically // set to null)

    }

    public void displayLink() { // display ourself

        System.out.print "{" + iData + ", " + dData + "} ";

    }

} // end class Link


class LinkedList {

    private Link first; // ref to first link on list


    public LinkedList() { // constructor

        first = null; // no items on list yet

    }


    public boolean isEmpty() { // true if list is empty

        return (first==null);

    }


    // insert at start of list

    public void insertFirst(int id, double dd) {

        // make new link

```

```

        Link newLink = new Link(id, dd);

        newLink.next = first; // newLink --> old first

        first = newLink; // first --> newLink
    }

    public Link deleteFirst() { // delete first item (assumes list not empty)

        Link temp = first; // save reference to link

        first = first.next; // delete it: first-->old
        next

        return temp; // return deleted link
    }

    public void displayList() {

        System.out.print("List (first-->last): ");

        Link current = first; // start at beginning of list

        while(current != null)

            // until end of list,

            {

                current.displayLink(); // print data

                current = current.next; // move to next link

            }

        System.out.println("");
    }

```

```
} // end class LinkList
```

### Clase principal

```
class LinkListApp {  
    public static void main(String[] args) {  
        LinkList theList = new LinkList(); // make new list  
  
        theList.insertFirst(22,2.99); // insert four items  
  
        theList.insertFirst(44,4.99);  
  
        theList.insertFirst(66,6.99);  
  
        theList.insertFirst(88,8.99);  
  
        theList.displayList(); // display list  
  
        while( !theList.isEmpty() ) // until it's empty,  
        {  
            Link aLink = theList.deleteFirst(); // delete link  
  
            System.out.print("Deleted "); // display it  
  
            aLink.displayLink();  
  
            System.out.println("");  
        }  
  
        theList.displayList(); // display list  
    } // end main()  
}
```

```
} // end class LinkListApp
```

Estudiar a detalle el código anterior. Comentar los programas ejecutados para una explicación clara de cada uno de dichos programas.

**Entregar las respuestas de su cuestionario en una presentación en power point.**

## **CUESTIONARIO**

- 1 Identificar las clases que implementan un programa de listas enlazadas de estudiantes.
- 2 Identificar los métodos usados para implementar el programa de listas enlazadas de estudiantes.
- 3 Solamente creará los métodos: Insertar al principio de la lista, Buscar por valor, eliminar elemento e imprimir los elementos de la lista.