```
import java.util.*;

class ListNode
{
    public String   element;
    public ListNode nextAddress;

     // Constructors

   public ListNode( String theElement )
    {
        element= theElement;
        nextAddress=null;
    }

    public ListNode( String theElement, ListNode n )
    {
        element = theElement;
        nextAddress    = n;
    }




    public static void main (String [] a)
    {

     ListNode n1=new ListNode("Nina");

     ListNode n2=new ListNode("Kalle",n1);
     ListNode n3=new ListNode("Olle",n2);
     ListNode n4=new ListNode("Mio",n3);

     ListNode n =n4;
     while(n!=null)
     {
      System.out.println(n.element);
        n=n.nextAddress;
     }


    }

}


   import java.util.*;

/**
 *  * Första noden i listan innehåller inget objekt
```

```
 */

    public class SimpleLinkedList{

      private ListNode head;

       public SimpleLinkedList( ) {
        head = new ListNode( null );
      }

    /**
    Skapar en ny ListNode objekt med en String och lägger den
i listan
     */
       public void insert (String theobj)
      {
         ListNode nynode = new ListNode(theobj);

         ListNode temp = head;
         while(temp.nextAddress != null){
            temp=temp.nextAddress;
         }

         temp.nextAddress=nynode;
      }


   /*
   Ta bort noden som innehåller respektive objekt
     */
       public void remove(String theobj) {
         ListNode temp = head;

         while(temp.nextAddress != null){
            if(temp.nextAddress.element.equals(theobj)){
               temp.nextAddress =
temp.nextAddress.nextAddress;
               break;
            }
            temp = temp.nextAddress;
         }
      }

    /**
    Skriver ut innehållet i listan
     */
       public void print() {
         ListNode node = head.nextAddress;
         while(node != null){
            System.out.println(node.element);
            node = node.nextAddress;
```

```
        }
      }

       public void insert2(String theobj ) {
         ListNode nynode = new ListNode(theobj,
head.nextAddress);
        head.nextAddress = nynode;
      }

    /**
    Returnerar noden som innehåller det sökta objektet
     */
       public ListNode find(String theobj) {
         ListNode node = head.nextAddress;
         while(!node.element.equals(theobj)){
            node = node.nextAddress;
         }
         return node;
      }

     /**

        Sorterat i alfabetisk ordning
        */
       public void insertSort( String theobj)
      {
         ListNode n =new ListNode(theobj);
         ListNode curr = head.nextAddress;
        ListNode prev=null;

         while( (curr!=null) &&
theobj.compareTo(curr.element)>0 ){
            prev=curr;
            curr = curr.nextAddress;


         }

      // när nya elment skall läggas till i början av listan
         if( prev==null){
            n.nextAddress=head.nextAddress;
            head.nextAddress=n;

         }


         // när nya elementet skall vara sist i listan

         else if( curr==null)
         {
            prev.nextAddress=n;
```

```
        }

        // när den nya elementet skall vara mellan
        else
        {
            n.nextAddress=curr;
            prev.nextAddress=n;

        }


    }


     public static void main ( String [] arg)
    {

        SimpleLinkedList klassLista=new SimpleLinkedList();
        klassLista.insertSort("A");
        klassLista.insertSort("C");
        klassLista.insertSort("B");

        klassLista.print();


    }


}
```