

```
import java.util.NoSuchElementException;
```

```
public class Stacky<E> {
```

```
    private Node top; //mientras no se les asigne un valor estan en null
```

```
    private int size;
```

```
    public Stacky() {
```

```
        size = 0;
```

```
    }
```

```
    /**
```

```
     * this class keeps track of each element information
```

```
     * @author java2novice
```

```
     *
```

```
    */
```

```
    private class Node {
```

```
        E element;
```

```
        Node next; //estos nodos valen null
```

```
        public Node(E element) {
```

```
            this.element = element;
```

```
            this.next = null;
```

```
        }
```

```
    }
```

```
    /**
```

```
     * returns the size of the linked list
```

```
     * @return
```

```
    */
```

```
    public int size() { return size; }
```

```
    /**
```

```
     * return whether the list is empty or not
```

```
     * @return
```

```
    */
```

```
    public boolean isEmpty() { return size == 0; }
```

```
    /**
```

```
     * adds element at the starting of the linked list
```

```
     * @param element
```

```
    */
```

```
    public void push(E element) {
```

```
        System.out.println("Añadiendo nodo *****");
```

```
        Node tmp = new Node(element);
```

```
        System.out.printf("TEMP: %s, %s\n", tmp.element, tmp);
```

```
        System.out.printf("TOP ANTES: %s\n", top);
```

```
        if(top == null) {
```

```
            top=tmp;
```

```
        }else {
```

```
            Node n = top;
```

```
            top= tmp;
```

```
            top.next = n;
```

```
        }
```

```

        System.out.printf("TOP DESUES: %s\n", top);
        size++;
        System.out.println("adding: " + element);
    }

```

```

/**
 * this method walks forward through the linked list
 */

```

```

public void iterateForward(){

    System.out.println("iterating forward..");
    Node tmp = top;
    while(tmp != null){
        System.out.println(tmp.element);
        tmp = tmp.next;
    }
}

```

```

/**
 * this method removes element from the start of the linked list
 * @return
 */

```

```

public E pop(E element) {

    if (size == 0) throw new NoSuchElementException();
    Node tmp = top;
    top = top.next;
    size--;
    System.out.println("POP*****");
    return tmp.element;
}

```

```

public static void main(String a[]){
    Stacky<Integer> dll = new Stacky<Integer>();
    dll.push(10);
    dll.push(34);
    dll.push(56);
    dll.iterateForward();
    dll.pop(34);
    dll.iterateForward();
}
}

```

