

---

# Listak

---

[Creative Commons](#) lizentziapean publikatua

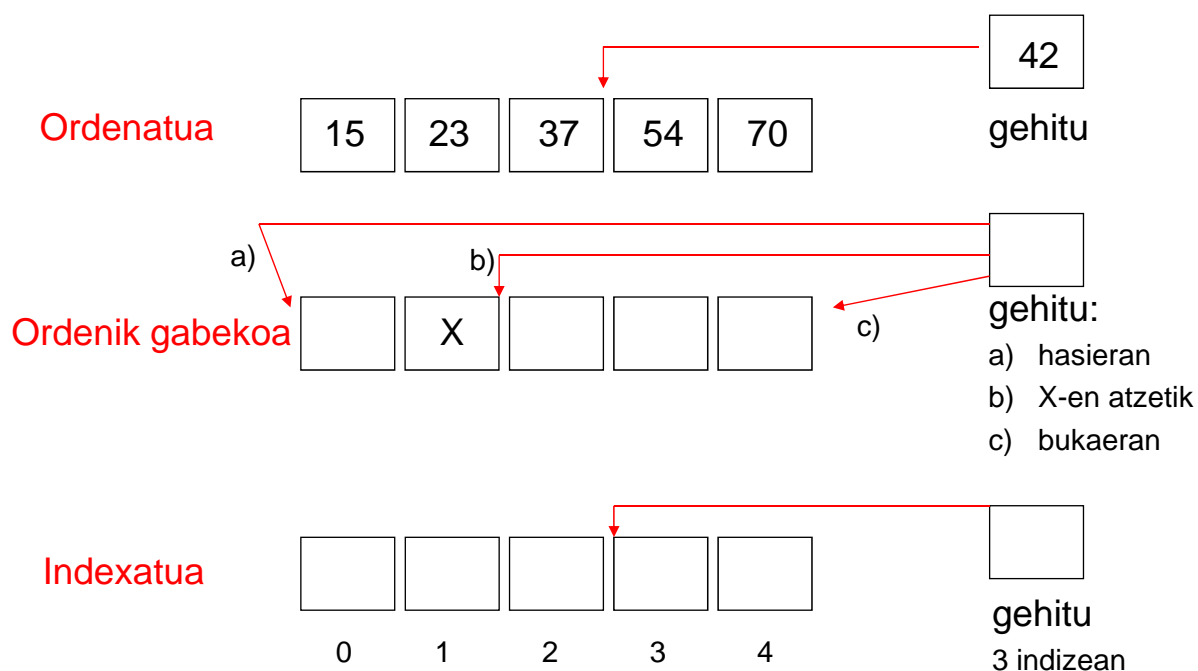


---

## Lista mota desberdinak

- Lista edo zerrenda bat elementuen bilduma lineala da. Beraien ezaugarrien arabera, 3 mota nagusitan sailkatuko ditugu:
  - **Lista ordenatuak**
    - Elementuak ordena-erlazio baten arabera daude ordenatuta
  - **Lista ez-ordenatuak**
    - Elementuak ez daude ezaugarrien arabera ordenatuta, sartu diren ordenarekiko baizik
  - **Lista indexatuak**
    - Elementuak zenbakizko indize baten bidez atzitu daitezke

## Lista mota desberdinak



## Lista mota guztiek dauzkaten eragiketak

Eragiketa	Deskribapena
T removeFirst()	Listako lehen elementua ezabatzen du
T removeLast()	Listako azken elementua ezabatzen du
T remove(T elem)	Listako elementu jakin bat ezabatzen du
T first()	Listako lehen elementuaren atzipena
T last()	Listako azken elementuaren atzipena
boolean contains(T elem)	Listak ea elementu jakin bat duen ala ez
boolean isEmpty()	Ea lista hutsa den
int size()	Listaren elementu-kopurua ematen du

## Lista ordenatuek bakarrik daukaten eragiketa

Eragiketa	Deskribapena
<code>void add(T elem)</code>	Elementu berria gehituko du listan (ordenatuta dagokion tokian)

## Lista ez-ordenatuek bakarrik daukaten eragiketak

Eragiketa	Deskribapena
<code>void addToFront(T elem)</code>	Elementua gehituko du listaren hasieran
<code>void addToRear(T elem)</code>	Elementua gehituko du listaren bukaeran
<code>void addAfter(T elem, T target)</code>	<i>elem</i> elementua gehituko du beste elementu jakin baten atzean ( <i>target</i> ), elementu hau listan badagoelarik

# Lista indexatuen eragiketa bereziak

Eragiketa	Deskribapena
<code>void add(int i, T elem)</code>	Elementu berria gehituko du adierazitako indizean
<code>void set(int i, T elem)</code>	Zehaztutako indizean dagoen elementua esleituko du
<code>T get(int i)</code>	Emandako indizearen elementuaren atzipena bueltatzen du
<code>int indexOf(T elem)</code>	Elementu horren indizea bueltatuko du
<code>T remove(int i)</code>	Emandako indizeari dagokion elementua listatik kenduko da eta bere atzipena bueltatuko da

Pentsatu nola implementatu daitekeen lista ez-ordenatu bat lista indexatua erabiliz

# Lista indexatuaren implementazioa Array baten bidez

```

public class ListaArray<T> implements ListaIndexatuaDMA<T> {
    private int maxSize;    // array-aren tamaina
    private T[] lista;
    private int luzera;    // libre dagoen lehen elementuaren indizea eta
                        // listaren elementu-kopurua
    //-----
    public ListaArray(int maxLuzera)
    {
        maxSize = maxLuzera;
        lista = (T[])(new Object[maxSize]);
        luzera = 0;
    }
    //-----
    public T first ()    // listako lehenengoa
    {
        return lista[0];
    }
    //-----
    public T last ()    // listako azkena
    {
        return lista[luzera - 1];
    }
}

```

## Lista indexatuaren inplementazioa Array baten bidez

```

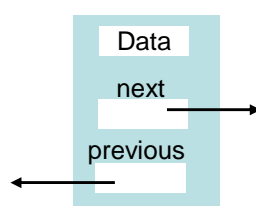
public T removeLast () { // listako azkena kentzen du
    T lag = lista[--luzera]; // listako azkena
    lista[luzera] = null; // erreferentzia ezabatu
    return lag;
}
//-----
public boolean isEmpty () { // Ea lista hutsik dagoen ala ez
    return (luzera == 0);
}

```

- Falta diren eragiketen inplementazioa ariketa moduan uzten da.

## Listaren inplementazioa egitura estekatuaren bidez

- Aukera bat esteka bateko adabegiak erabiltzea da (LinearNode<T> bezalakoak). Hau ariketa moduan uzten dugu.
- Beste aukera bat bi estekako adabegiak erabiltzea da: bata hurrengoari eta bestea aurrekoari



## DoubleNode<T> klasea

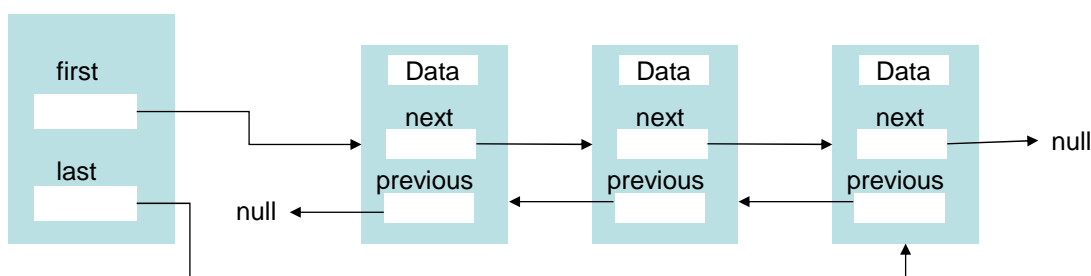
```
class DoubleNode<T> {

    private DoubleNode<T> next;
    private DoubleNode<T> previous;
    private T element;

    public DoubleNode(){ }
    public DoubleNode(T elem){}
    public DoubleNode<T> getNext() { }
    public void setNext(DoubleNode<T> node) { }
    public DoubleNode<T> getPrevious() { }
    public void setPrevious(DoubleNode<T> node) { }
    public T getElement() {}
    public void setElement(T elem) { }
}
```

11

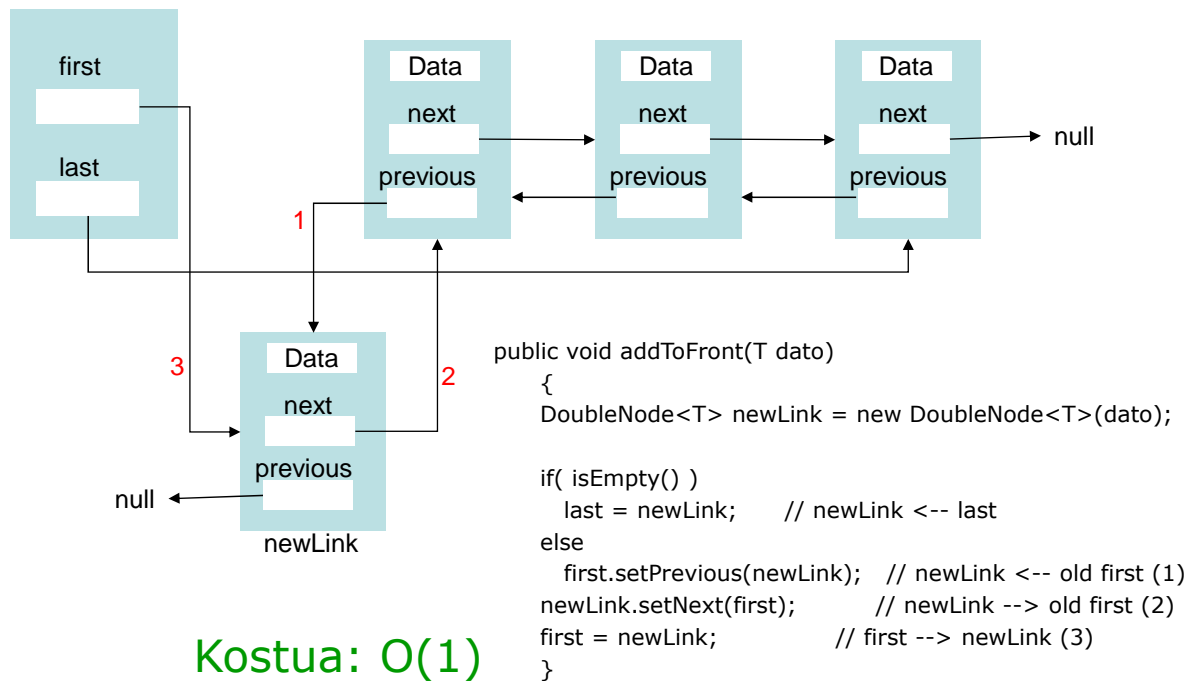
## Estekadura bikoitzeko listak (0)



```
class DoubleNode<T>
{
    private T data; // informazioa
    private DoubleNode<T> next; // hurrengo adabegia
    private DoubleNode<T> previous; // aurreko adabegia
    .....
}
```

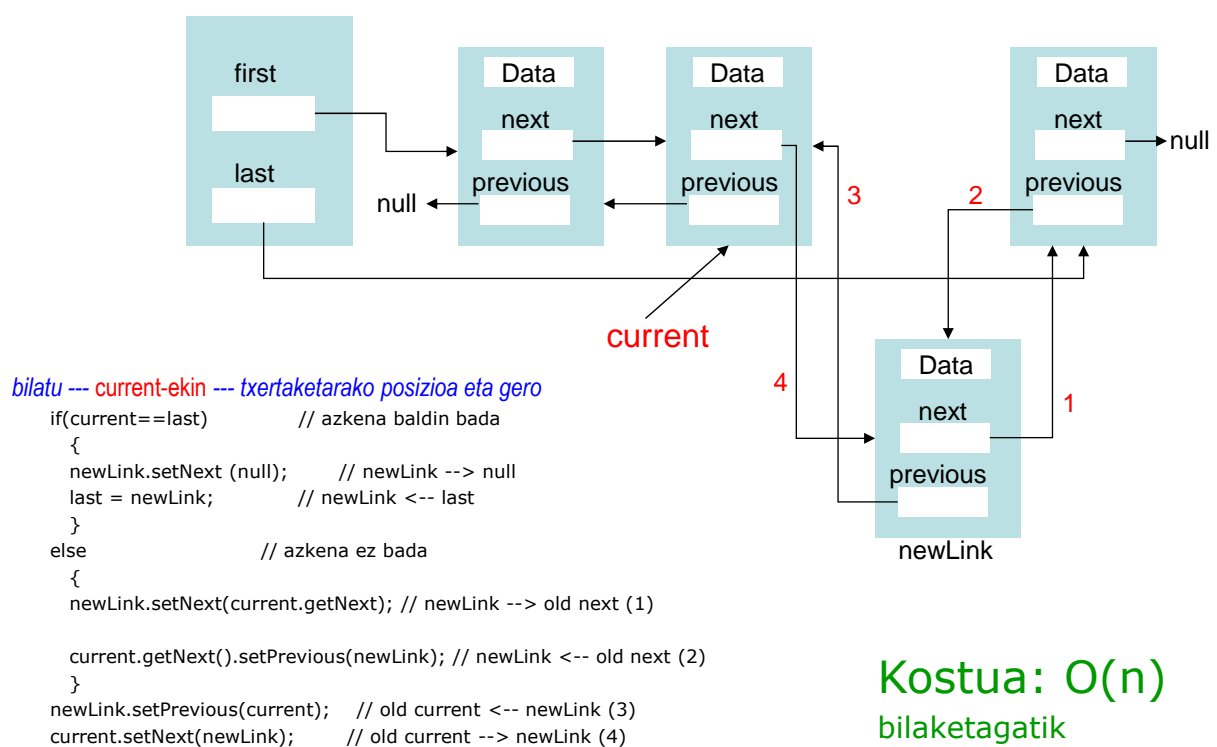
12

# Hasieran txertatu (1)



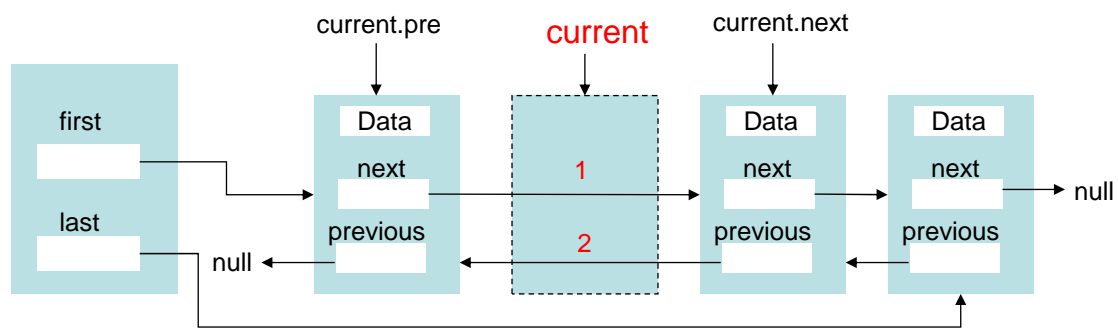
13

# Txertaketa (2)



14

## Ezabaketa (3)



Bilatu --- current-ekin --- ezabatu behar den adabegia eta gero

Kostua:  $O(n)$   
bilaketagatik

```
if(current==first)           // lehenengoa baldin bada
    first = current.getNext(); // first --> old next
else                         // lehenengoa ez bada
    current.getPrevious().setNext(current.getNext()); // old previous --> old next (1)

if(current==last)           // azkena bada
    last = current.getPrevious(); // old previous <-- last
else                         // azkena ez bada
    current.getNext().setPrevious(current.getPrevious()); // old previous <-- old next (2)
```

## Irakurketa

- Javaren API Collections-ek dauzka lista indexatuaren inplementazioak
  - ArrayList
  - LinkedList
- [Lewis, Chase 2010]
  - 6. kapitulua