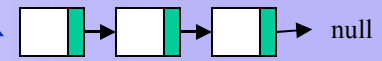


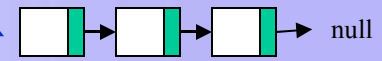
Zerrenda estekatuak



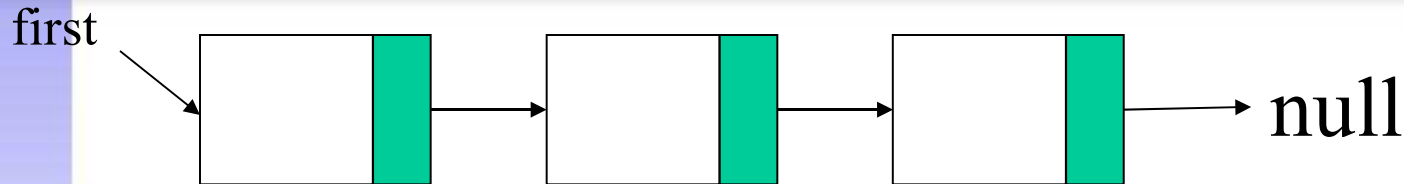
LinkedList: Zer dira?

- Elementuak memoria **posizio jarraian gordetzen ez diren** datu egitura bat da
- Datu **mugimendu handiak ez egiteko** erabiltzen dira, baina elementu bakoitzak **osagarritzko erreferentzia bat** gorde behar du
- Elementu bakoitza nodo objektu batean gordetzen da hurrengo edukiarekin:
 - gorde nahi dugun **objektua**
 - **Erreferentzia** bat zerrendaren hurrengo elementuari.
- Hasieran erreferentzia bat gordetzen da zerrendaren lehenengo elementuari (**first**)
- Beste **atributu eta metodo** berriak definitu daitezke funtzionalitate osagarriarekin

Zerrenda estekatuak



LinkedList: Zer dira?



```
public class LinkedList<T>{  
    Node<T> first;  
}
```

```
class LinkedList
```

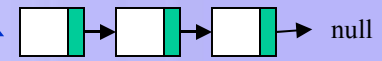
- Atributua: Node top

```
public class Node<T> {  
    T data;  
    Node<T> next;  
    public Node(T pElem){  
        data=pElem;  
    }  
}
```

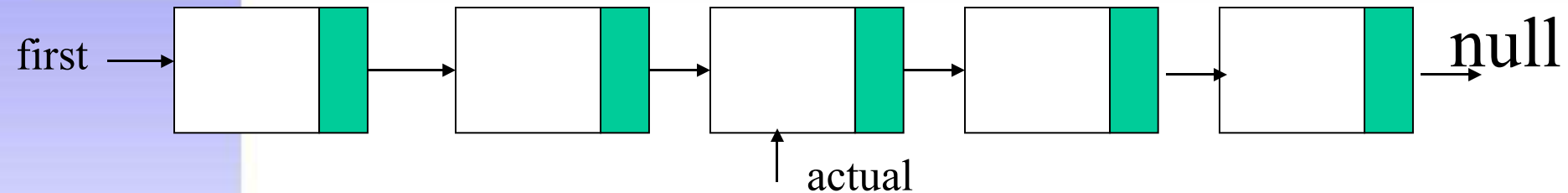
```
class Node
```

- Atributua: Object data
- Atributua: Node next
- Eraikitzailea

Zerrenda estekatuak



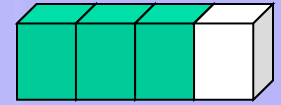
LinkedList: Klase osoa



```
class LinkedList
```

```
•Atributua: Node first  
•Atributua: Node actual ;  
•Metodoak:  
LinkedList () /* eraikitzailea  
void insertFirst()  
void insert ()  
void remove (T elem)  
void remove ()  
boolean find (T elem)  
void goFirst()  
void goNext()  
T getActual ()  
boolean hasNextElement ()  
void emptyList()  
boolean isEmpty ()
```

Zerrenda estekatuak



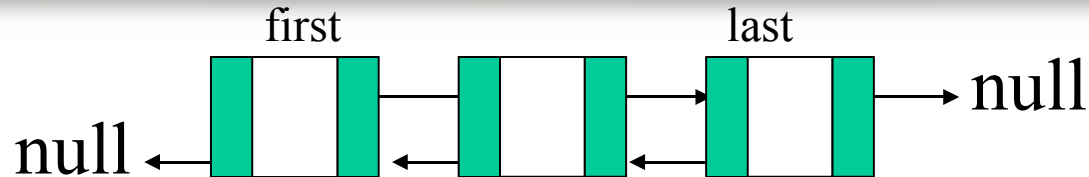
LinkedList: metodoak

Mugimendu metodoak	esanahia
void goFirst ()	Actual zerrendaren lehenengo elementura erreferenziatzen du
void goNext ()	Actual erreferenziatzen du, actual-en hurrengo nodora
boolean find (T elem)	actual erreferenziatzen du elem agertzen den lehendabiziko nodora, eta true izultzen du. Agertzen ez bada, actual=null eta false itzultzen du.

Txertaketa, ezabaketa eta aldaketa metodoak	esanahia
void insert (T elem)	elem objektua txertatzen du actual elementuaren jarraian eta actual nodo berriari erreferenziatzen du. Actual null bada ez da ezer eiten.
void insertFirst (T elem)	Elem objektua zerrendaren hasieran txertatzen da.
void remove (T elem)	Elem objektuaren lehendabiziko agerpena ezabatzen da zerrendatik eta actual erreferenziatzen du zerrendaren lehendabiziko elementura
void remove ()	Actual erreferenziatzen duen elementua ezabatzen da eta actual hurrengora erreferenziatuko du.
T getActual ()	Actual erreferenziatzen duen nodoaren edukia itzultzen du.
boolean hasNextElement ()	True, actual azken elementua ez bada, eta false beste kasuan
boolean isEmpty ()	True, zerrenda hutsa bada, eta false beste kasuan
void emptyList ()	Zerrendaren elementu guztiak ezabatzen ditu

Esteka bikoitzeko zerrendak

DoubleLinkedList: Zer dira?



```
public class DoubleLinkedList<T>{  
    Node<T> first;  
    Node<T> last;  
}
```

```
public class Node<T>{  
    T data;  
    Node<T> next;  
    Node<T> prev;  
    public Node (Object pElem){  
        data=pElem;  
    }  
}
```

Klase **DoubleLinkedList**

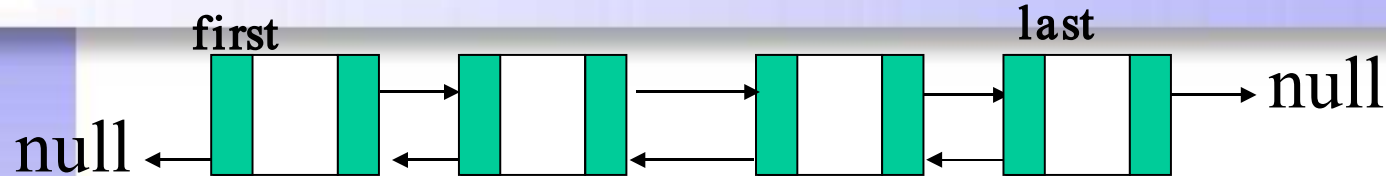
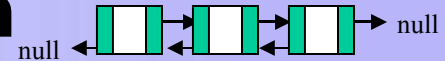
- Atributua: Node first
- Atributua: Node last

Klase Node

- Atributua: Object data
- Atributua: Node next
- Atributua: Node prev
- Eraikitzailea

Esteka bikoitzeko zerrendak

DoubleLinkedList: klase osoa



class DoubleLinkedList

•Atributuak:

Node <T> first

Node <T> last

Node <T> actual

•Metodoak:

DoubleLinkedList () /* eraikitzailea

void **insert** (T elem);

void **insert First** (T elem); void **insert Last** (T elem)

void **remove** (T elem); void **remove** (); void

emptyList()

boolean **find** (T elem);

void **go First**();

void **go Last**();

boolean **hasNextElement**();

boolean **hasPreviousElement**();

void **go Next**();

void **go Prev**();

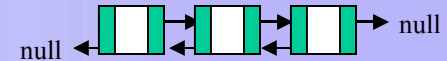
T **getActual**();

boolean **isEmpty**();

```
public
DoubleLinkedList () {
    first=null;
    last=null;
    actual=null;
}
```

Esteka bikoitzeko zerrendak

DoubleLinkedList: metodoak

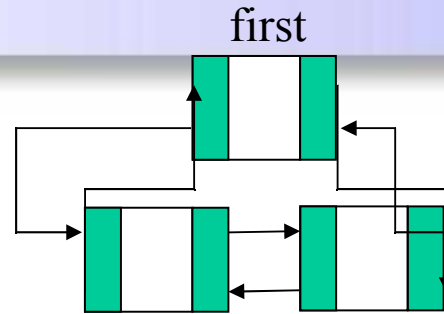
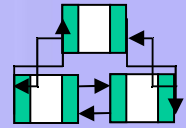


Mugimendu metodoak	esanahia
void goFirst() / goLast()	actual zerrendaren lehenengo/azkeneko elementura erreferenziatzen du
void goNext() / goPrev()	Actual erreferenziatzen du, actual-en hurrengo / aurreko nodora
boolean find (T elem)	actual erreferenziatzen du elem agertzen den lehendabiziko nodora, eta true izultzen du. Agertzen ez bada, actual=null eta false itzultzen du.

Txertaketa, ezabaketa eta aldaketa metodoak	esanahia
void insert (T elem)	elem objektua txertatzen du actual elementuaren jarraian eta actual nodo berriari erreferenziatzen du. Actual null bada ez da ezer egiten.
void insertFirst (T elem) / insertLast (T elem)	Elem objektua zerrendaren hasieran/bukaeran txertatzen du
void remove (T elem)	Elem objektuaren lehendabiziko agerpena ezabatzen da zerrendatik eta actual erreferenziatzen du zerrendaren lehendabiziko elementura
void remove ()	Actual erreferenziatzen den elementua ezabatzen da eta actual hurrengora erreferenziatuko du.
T getActual ()	Actual erreferenziatzen duen nodoaren edukia itzultzen du.
boolean hasNextElement () / hasPrevElement ()	True, actual azken/lehendabiziko elementua ez bada, eta false beste kasuan
boolean isEmpty ()	True, zerrenda hutsa bada, eta false beste kasuan
void emptyList ()	Zerrendaren elementu guztiak ezabatzen ditu

Zerrenda Zirkularrak

DoubleEndedLinkedList: Zer dira?



```
public class DoubleEndedLinkedList<T>{  
    Node<T> first;  
}
```

```
public class Node<T>{  
    T data;  
    Node<T> prev;  
    Node<T> next;  
    public Node(T pElem){  
        data=pElem;  
    }  
}
```

```
class DoubleEndedLinkedList
```

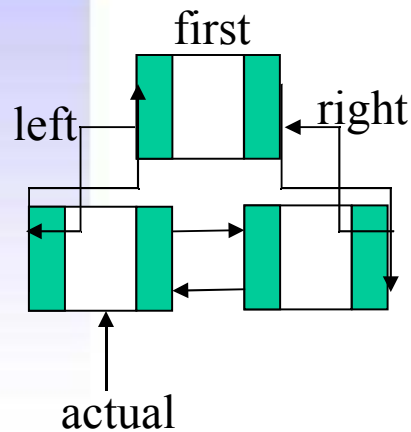
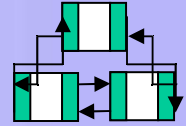
- Atributua: Node first

Klase Node

- Atributua: Object element
- Atributua: Node prev
- Atributua: Node next
- Eraikitzailea

Lista Zirkularrak

DoubleEndedLinkedList: Klase osoa



Class DoubleEndedLinkedList<T>

- Atributua: Node first
- Atributua Node **actual**;
- Metodoak:
 - void insert(T elem);
 - T remove(T elem);
 - boolean find(Object elem);
 - T getActual();
 - void goFirst();
 - void goRight();
 - void goLeft();
 - boolean isEmpty();

Bektoreak vs. Lista Zirkularrak

	BEKTOREA	LinkedList
<i>Gordetzen diren datu mota</i>	T motako elementuak (Vector<T> v;)	<ul style="list-style-type: none"> • Node klasean definitutako datu mota • Erreferentziak gordetzen dira hurrengo elementura
<i>Tamaina</i>	Metodoen bidez Luzatu daiteke dinamikoki	<ul style="list-style-type: none"> • Metodoen bidez, • Handitu daiteke dinamikoki • Leku osagarri behar du hurrengo elementuei erreferentziak gordetzeko
<i>Atzipena, Txertaketa, Ezabaketa, Aldaketak</i>	Metoden bidez: addElement(T obj), removeElement(T obj), insertElementAt(T obj, int index),	Node klase osagarri erabiltzen da. Metodoen bidez: addElement(T obj), removeElement(T obj), emptyList()

Ondorioak

- **Array-ak eta bektoreak** eraginkorragoak dira elementuen **atzipena ordenarik gabe** (ausazko ordenan) egiten denean
- **Zerrenda estekatuak** eraginkorragoak dira posizio desberdinetan elementuen **txertaketa eta ezabaketa ugari** egin behar direnean