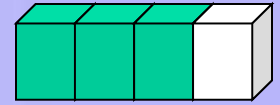


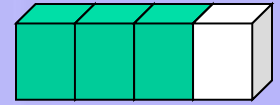
Bektoreak

Zer dira?



- **Object** datu motako elementu bilduma bat da.
- Handitu/Txikitu daiteke **dinamikoki** elementu kopuruen arabera
- elementuak **indize** baten bidez atzitu daitezke, baina **ez [] bidez**
- Tamaina
 - **capacity()** metodoak, bektoreak eduki dezaken elementu kopurua itzultzen du.
 - **size()** metodoak momentu horretan dauzkan elementu kopurua itzultzen du.
 - **capacityIncrement** aldagaiak, bektoreak haunditu behar denean zenbat unitatetan haziko da adierazten du

Bektoreak



Zer dira? Metodo eraikitzaileak

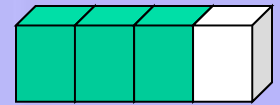
Klas e bat da

Ele men tu m ul tz o bat k udea tz eko er abi ltz en da
java .ut il l ibur ute gia n es kur a gar ri dag o

<i>Eraikitzaileak</i>	<i>Esanahia</i>
<code>Vector<T>()</code>	Bektore huts bat sortzen du
<code>Vector<T>(int capacity)</code>	<i>capacity</i> kopuruzko bektore bat sortzen du
<code>Vector<T>(int capacity, int capacityIncrement)</code>	<i>capacity</i> kopuzko bektore bat sortzen du eta bere igoera kopurua <i>capacityIncrement</i> elementuetan ipintzen du

Bektoreak

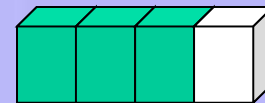
Atzipen metodoak



<i>Atzipen metodoak</i>	<i>Esanahia</i>
T firstElement()	Lehenengo elementua itzultzen du
T lastElement()	Azkeneko elementua itzultzen du
T elementAt(int index)	Index posizioan dagoen elementua itzultzen du
Enumeration<T> elements()	Elementuen zerrenda itzultzen du
boolean contains(T elem)	elem elementua bektorean dagoen konprobatzen du
int indexOf(T elem)	elem elementua bektorean lehenengo aldiz agertzen den posizioa itzultzen du
int indexOf(T elem, int index)	elem elementua bektorean index posiziotik aurrera lehenengo aldiz agertzen den posizioa itzultzen du
int lastIndexOf(T elem)	elem elementua bektorean agertzen den azken aldiaren posizioa itzultzen du
int lastIndexOf(T elem, int index)	elem elementua bektorean index posiziotik aurrera azken aldiz agertzen den posizioa itzultzen du

Bektoreak

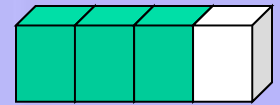
Txertatu, ezabatu eta aldatzeko metodoak



Elementuak Txertatu, ezabatu eta aldatzeko metodoak	Esanahia
<code>void addElement(T elem)</code>	<i>elem</i> objektua txertzen da bektore bukaeran
<code>void insertElementAt(T elem, int index)</code>	<i>elem</i> objektua txertzen da <i>index</i> posizioan
<code>void setElementAt(Telem, int index)</code>	<i>index</i> posizioan dagoen elementua, <i>elem</i> objektuarengatik aldatzen da
<code>boolean removeElement(T elem)</code>	<i>elem</i> objektuaren lehen agerpena ezabatzen da
<code>void removeElementAt(int index)</code>	<i>index</i> posiziozko elementua ezabatzen da
<code>void removeAllElements()</code>	elementu guztiak ezabatzen dira

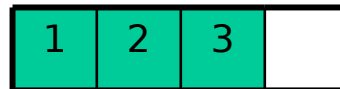
Bektoreak

Tamaina metodoak



Metodoak	Esanahia
<code>int capacity()</code>	bektoreak eduki dezaken elementu kopurua itzultzen du
<code>int size()</code>	momentu horretan dauzkan elementu kopurua itzultzen du
<code>void setSize(int newSize)</code>	Tamaina berri bat ipintzen du
<code>void trimToSize()</code>	Bektorearen tamaina egokitzen da momentuan dauden elementuei
<code>boolean isEmpty()</code>	Bektorea elementuak dauzka konprobatzen du

Vector v

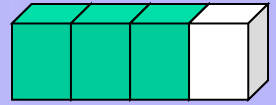


v.size() = 3

v.capacity() = 4

Abibideak: Bektoreak

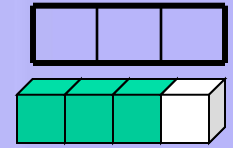
Erazagupena, Sorkuntza, Hasieraketa,
Atzipena



```
import java.util.Vector;
public class BektoreakProba2{
    public static void main(String args[]){
        Vector<Integer> v; //Bektorea erazagutzen da
        v= new Vector<Integer>(); //Bektorea sortzen dugu
        Integer obj1= new Integer(1); //Integer motako bi objektu
        Integer obj2= new Integer(2); //sortzen ditugu bektorean
        v.addElement(obj1); //gordetzeko
        v.addElement(obj2);
        Integer i1 = v.elementAt(0);
        int osoa1 = i1.intValue();
        Integer i2 = v.elementAt(1);
        int osoa2 = i2.intValue();

        System.out.println(osoa1+" eta "+ osoa2+" = "+(osoa1+osoa2));
    }
}
```

Array-ak vs Bektoreak



	ARRAY	BEKTORE
<i>Gor det zen di ren datu mot a</i>	Deklaratzerakoan ipintzen dena (int a[])	T motako <u>objektuak</u> (Vector<T> v;)
<i>Tam ain a</i>	length atributuaren bidez (a.length)	size() capacity() metodoen bidez (v.size())
<i>Tam ain a</i>	Sortu denean definitutako tamaina finkoa (a= new a[3])	Behar den neurrian hazi daiteke dinamikoki (v.addElement())
<i>At zip en a</i>	[] eragilearen bitartez (a[0]=2)	Metodoen bidez: addElement(Object obj), removeElement(Object obj), insertElementAt(Object obj, int index)
<i>De finituta dago</i>	Java lengoaian	Liburutegi klase batean