Datu-egiturak

2.0.1. gaia: array eta matrize-egiturak

***Juanan Pereira**ren eta **Jon Iturriotz**en*
gardenkietan oinarritutako gardenkiak dira hauek,
euren *Creative Commons* lizentzia bera daukana.

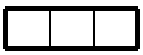
Aurkibidea

Zer dira?

Zergatik dira baliagarriak?

Egitura sinpleak

Array-ak



Erazagupena, sorkuntza eta hasieraketa

Txertaketa, ezabaketa eta elementuen bilaketa

Bektoreak

Erazagupena, sorkuntza eta hasieraketa

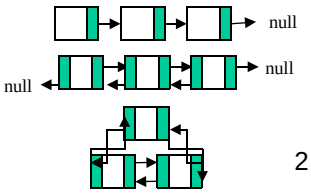
Txertaketa, ezabaketa eta elementuen bilaketa

Zerrenda estekatuak

Esteka sinpleko zerrendak

Esteka bikoitzeko zerrendak

Zerrenda zirkularrak



Datu-egiturak
Zer dira?

Datu-egitura bat oinarritzen da:
-datu batzuen **errepresentazioan**
-Datu horiengan onartutako **eragiketa** multzo batean

Egiten diren ohiko eragiketak:
Txertaketa
Ezabaketa
Bilaketa

Datu-egituren arteko desberdintasun nagusienak:
-datuak **antolatzeko** modua
-eragiketetan ezartzen diren **murriztapenak**
(adib: txertatutako azken elementua soilik atzitu daiteke).

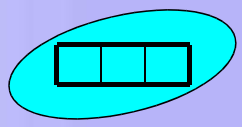
Datu-egiturak
Zergatik dira baliagarriak?

Berrerabilpena: Programatu den kodea egiten diren programa berrietan erabil daiteke.

Abstrakzioa: funtzionalitatearen deklarazioa (interfazea) ez dago inplementazioaren mende
Ikuspuntu honekin, datuen egitura alda daiteke, datu-egitura honen erabiltzaileak ohartu gabe.

Array-ak (Matrizeak)

Zer dira?



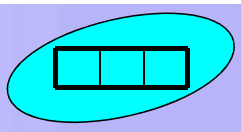
Unitate bat osatzen duten datu mota bereko elementuen bilduma

indexazio [ ] eragilea erabiliz atzitu daiteke array-aren edozein elementu.

length atributuak array-ak daukan elementu kopurua itzultzen du

Array-ak (Matrizeak)

Erazagupena, Sorkuntza, Hasieraketa



Erazagupena: Array-ari **izendatzaile** bat ipini eta beren elementuen **datu mota** definitu

- Bi eratara egin daiteke:

```
mota arrayIzena[];
mota []arrayIzena;
```

- Erazagutu ondoren, oraindik ez da memoria erreserbatu/esleitu datuak gordetzeko

Sorkuntza: Array-aren memoria-tokia erreserbatzen da

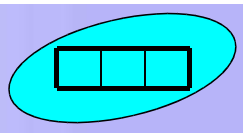
- new erabiltzen da eta array-ren **tamaina** zehaztu behar da:

```
arrayIzena = new mota[tamaina];
```

- array bat sortu ondoren eta beste modu batean hasieratu bitartean, array-aren elementuek dituzten balio lehenetsiak hartzen dituzte.

Balio lehenetsiak:
int, short, long = 0
float, double = 0.0
boolearrak = false
String = null
Object = null

Array-ak (Matrizeak)



Erazagupena, sorkuntza, hasieraketa

Hasieraketa: Array-aren elementuei **balioa ematen zaie**. Era desberdinetan egin daiteke:

Banaka:

```
arrayName[0] = element0;
arrayName[1] = element1;
...
```

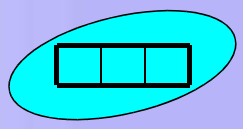
Begizta baten bidez:

```
for(int i = 0; i < arrayName.length; i++){
 arrayName[i] = balioa;
}
```

Esleipen zuzenen bidez:

```
DatuMota arrayName [] = {elem1, elem2, elem3, ...};
```


Adibideak: Array-ak (Matrizeak)



Erazagupena, sorkuntza, hasieraketa

Oinarrizko datu moten array-ak

```
int a[];            //Erazagutu
a = new int[3]  //Sortu

//Hasieratu
a[0]=1;
a[1]=2;
a[2]=3;
```

```
//Erazagutu & sortu
int a[] = new int[3]

//Hasieratu
a[0]=1;
a[1]=2;
a[2]=3;
```

```
//Erazagutu eta sortu
int a[] = new int[3]

//hasieratu
for(int i=0; i<a.length;i++){
    a[i]=i+1;
}
```

```
int a[] = {1, 2, 3};  //Erazagutu, sortu eta hasieratu
```

Objektuen erreferentziaz osatutako array-ak

```
NireKlase a[];   //Erazagutu
a = new NireKlase [3]  //Sortu

//Hasieratu
a[0]=new NireKlase (param1);
a[1]=new NireKlase (param2);
a[2]=new NireKlase (param3);
```

```
//Erazagutu eta sortu
NireKlase a[] = new NireKlase [3]

//Hasieratu:
a[0]=new NireKlase (param1);
a[1]=new NireKlase (param2);
a[2]=new NireKlase (param3);
```

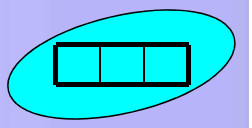
```
//Erazagutu eta sortu
NireKlase a[] = new NireKlase [3]

//Hasieratu:
for(int i=0; i<a.length;i++){
    a[i]=new NireKlase (param-i);
}
```

```
NireKlase a [] = {new NireKlase (param1), new NireKlase (param2), new NireKlase (param3)};
```

Array-ak (Matrizeak): Ohiko erroreak

Erazagupena, sorkuntza, hasieraketa



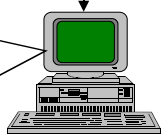
```
public class MatrizeAdibidea{
    public static void main(String args[]){
      double nireTaula[];
      System.out.println(nireTaula[0]);
    }
}
```

GAIZKI

konpilatu

Konpilazio errorea

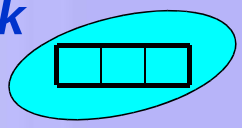
variable nireTaula may not have been initialized



Matrizea erazagututa soilik badago, ezin daitezke bere elementuak atzitu. Konpilatzerakoan, **errore** bat agertuko da.

Arrayak (Matrizeak) Ohiko erroreak

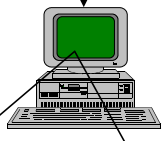
Erazagupena, sorkuntza, hasieraketa



```
public class MatrizeAdibidea2{
    public static void main(String args[]){
      int nireOsokoMatrizea[] = new int[10];
      float nireErrealMatrizea[]= new float[10];
      boolean nireBoolearMatrizea[] = new boolean[10];
      char nireKaraktereMatrizea[] = new char[10];
      String nireStringMatrizea[] = new String[10];
      Object nireObjektuMatrizea[] = new Object[10];
      System.out.println("Osokoen balio lehenetsia: " + nireOsokoMatrizea[0]);
      System.out.println("Errealen balio lehenetsia:" + nireErrealMatrizea[0]);
      System.out.println("Boolearren balio lehenetsia: " + nireBoolearMatrizea[0]);
      System.out.println("Karaktereen balio lehenetsia: " + nireKaraktereMatrizea[0]);
      System.out.println("String-en balio lehenetsia: " + nireStringMatrizea[0]);
      System.out.println("Objektuen balio lehenetsia: " + nireObjektuMatrizea[0]);
    }
}
```

konpilatu

Exekutatu

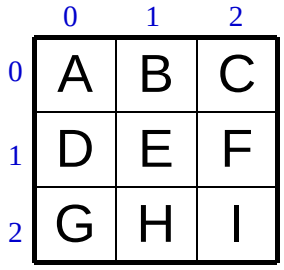


```
Osokoen balio lehenetsia : 0
Errealen balio lehenetsia: 0.0
Boolearren balio lehenetsia: false
Karaktereen balio lehenetsia:
String-en balio lehenetsia: null
Objektuen balio lehenetsia: null
```

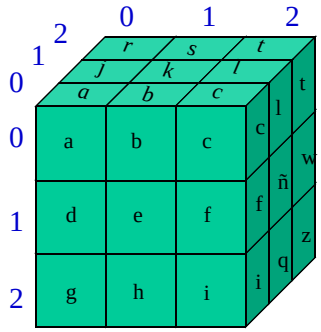
Matrizea **erazagututa eta sortuta** dagoenean baina **ez hasieratuta**, bere elementuak eskuragarri daude baina **balio lehenetsia** izango dute

Dimentsio anitzeko array-ak

Dimentsio bat baino gehiago duten array-ak dira. Elementuak atzitzeko indize bat baino gehiago behar dira.



A[0][2]='C'



A[0][2][1]='l'

```
char a[][];                 //Erazagutu
a = new char[3][3]          //Sortu
a[0][0]='A';                //Hasieratu
...
```

```
char a[][][];               //Erazagutu
a = new char[3][3][3]       //Sortu
a[0][0][0]='g';             //Hasieratu
...
```


Dimentsio anitzeko array-ak
Adibideak

Deklaratu eta sortu batera:

```
//Erazagutu eta Sortu
String [][]nireMatrizea = new String[3][4]
```

null	null	null	null
null	null	null	null
null	null	null	null

Erazagutu eta sortu bi pausotan:

```
int [][] nireMatrizea ;                    // array-a erazagutu
nireMatrizea = new int[errenkKop][]; // Erreferentziazko array-a sortu
```

Adibide gehiago:

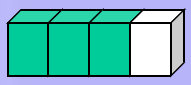
```
// 3x3 matrizea 0 balioarekin
int [][] a= new int[3][3];
```

0	0	0
0	0	0
0	0	0

```
int [][] b= {{1, 2, 3},
             {4, 5, 6}};
```

1	2	3
4	5	6

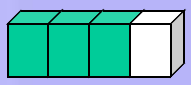
Bektoreak



Zer dira?

- **Object** datu motako elementuen bilduma bat da bektore bat.
- Bektorearen tamaina handitu eta txikitu daiteke **dinamikoki**
- **E**lementuak **indize** baten bidez atzi daitezke, baina **ez [ ] bidez**
- Bektorearen tamaina:
  - **capacity()** metodoak bektoreak har ditzakeen elementuen kopurua itzultzen du.
  - **size()** metodoak une horretan dauzkan elementuen kopurua itzultzen du.
  - **capacityIncrement** aldagaiak bektorea handitzean zenbat haziko den adierazten du.

Bektoreak



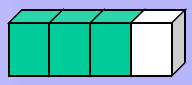
Zer dira? Zein eraikitzaile dituzte?

- Klase bat da (Vector klasea)
- Elementu multzo bat kudeatzeko erabiltzen da
- java.util paketean dago eskuragarri

Eraikitzaileak	*Esanahia*
Vector<T>()	Bektore huts bat sortzen du
Vector<T>(int capacity)	*capacity* kopuruko bektore huts bat sortzen du
Vector<T> (int capacity, int capacityIncrement)	*capacity* kopuruko bektore bat sortzen du etabere gehikuntza kopurua capacityIncrement aldagaian finkatzen da

Bektoreak

Atzipen metodoak

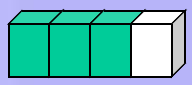


Atzipen-metodoak	*Esanahia*
T firstElement()	Lehenengo elementua itzultzen du
T lastElement()	Azkeneko elementua itzultzen du
T elementAt(int index)	Index posizioan dagoen elementua itzultzen du
Enumeration<T> elements()	Elementuen zerrenda itzultzen du

boolean contains(T elem)	elem elementua bektorean dagoen ala ez itzultzen 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 atzera lehenengo aldiz agertzen den posizioa itzultzen du

Bektoreak

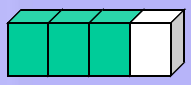
Txertatu, ezabatu eta aldatzeko metodoak



Elementuak txertatu, ezabatu eta aldatzeko metodoak	*Esanahia*
void add(T elem)	*elem* objektua txertatzen da bektore bukaeran
void insertElementAt(T elem, int index)	*elem* objektua txertatzen da index posizioan
void setElementAt(T elem, int index)	*index* posizioan dagoen elementua elem objektuarekin ordezkatzen da
boolean removeElement(T elem)	*elem* objektuaren lehen agerpena ezabatzen da
void removeElementAt(int index)	*index* posizioko elementua ezabatzen da
void removeAllElements()	elementu guztiak ezabatzen dira

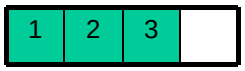
Bektoreak

Tamainari dagozkion metodoak



Metodoak	Esanahia
int capacity()	bektoreak har dezakeen elementu kopurua itzultzen du
int size()	une horretan dauzkan elementu kopurua itzultzen du
void setSize(int newSize)	Tamaina berri bat finkatzen du
void trimToSize()	Bektorearen edukiera = bektorearen elementu kopurua
boolean isEmpty()	Bektorea hutsik badago, true bueltatzen du; false, bestela.

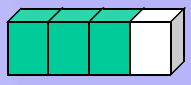
Vector <Integer> v



v.size() = 3
v.capacity() = 4

Abibideak: Bektoreak

Erazagupena, sorkuntza, hasieraketa, atzipena

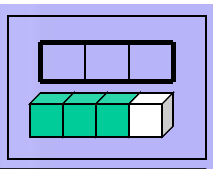


```
import java.util.Vector;
public class ProbaBektoreak{
  public static void main(String args[]){
    Vector<Integer> v;          //Bektorearen erazagupena
    v= new Vector<Integer>(); //Bektorea sortzen dugu
    Integer obj1= new Integer(1);
    Integer obj2= new Integer(2);
    v.add(obj1);
    v.add(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));
  }
}
```

//Integer motako bi objektu sortzen ditugu bektorean gordetzeko

ARIKETA: Vector klaseko v objektua sortuta eta hasieratuta dagoela suposatuz (Integer klaseko objektuez osatuta), kalkulatu bektoreko elementu txikienaren eta handienaren indizeak eta inprimatu.

Array-ak vs Bektoreak



	ARRAY	BEKTORE
Datu mota nola finkatu	Erazagutzean ipintzen dena (*int a[ ]*)	T datu motako objektuak (*Vector<T> v;*)
Tamaina	**length** atributuaren bidez (*a.length*)	size() eta capacity() metodoen bidez (v.size())
Dinamikoa / Estatikoa	Sortzean definitzen da tamaina, eta finkoa da (*a= new a[3]*)	Behar adina handi daiteke, **dinamikoki** (*v.add()*)
Txertaketa	[ ] eragilearen bidez (*a[0]=2*)	Metodoen bidez: *addElement(T obj) removeElement(T obj), insertElementAt(T obj, int index)*
Non dator?	java.lang paketean	java.util paketean