

# *Datu egiturak*

*Array-ak  
Bektoreak eta  
Zerrenda  
Estekatuak*

# Aurkibidea

- Zer dira?
- Zergatik dira baliagarriak?
- Egitura sinpleak

- **Array-ak** 

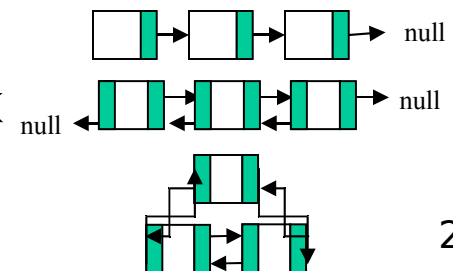
- Deklarazioa, sorkuntza eta hasieraketa
    - txertaketa, ezabaketa eta elementuen bilaketa

- **Bektoreak** 

- Deklarazioa, 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 ohizko eragiketak, datu baten:
  - Txertaketa
  - Ezabaketa
  - Bilaketadatu egitura batean izaten da
- Datu egituren arteko desberdintasun nagusienak, datuak **antolatze**ko modua eta eragiketetan ezartzen diren **murritzapenak** dira (adib: txertatutako azken elementua atzitu daiteke soilik).

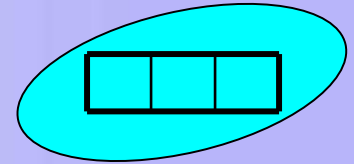
# ***Datu egiturak***

## **Zergatik dira baliagarriak?**

- **Berrerabilpen:** Programatu den kodea egiten diren programa berrietan erabili daiteke.
- **Abstrakzioa:** funtzionalitatearen deklarazioa (interfazea) edozein inplementazioaren aske dago
  - Ikuspuntu honekin, datuen egitura aldatu daiteke, datu egitura honen bezeroek ohartu gabe.
- **Erreferentzi biltegia:** Egitura hauek erreferentziak gordetzen dituzte eta ez datuen barne-kopiak.

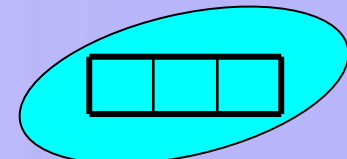
# Array-ak (Matrizeak)

## Zer dira?



- Unitate bat osatzen duten, datu mota berdineko elementu 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 elementuek zein **datu motatakoak** diren definitu

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

- Bi eratara egin daiteke
- Erazagutu ondoren, oraindik ez da memoria erreserbatu/esleitu datuak gordetzeko

*Besterik ezeko balioak:*

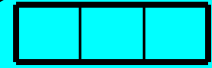
```
int, short, long = 0  
float, double = 0.0  
booleanoak = false  
String = null  
Object = null
```

- **Sorkuntza**: Array-aren memoria-tokia erreserbatzen da

- **new** erabiltzen da eta array-ren **tamaina** zehaztu behar da
- array bat sortu ondoren beren elementuek defektuzko balioak hartzen dituzte array-a hasieratu arte.

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

# Array-ak (Matrizeak)



## Deklarazioa, Sorkuntza, Hasieraketa

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

- Banaka

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

- Begizta baten bidez

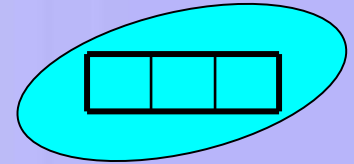
```
for(int i = 0; i < arrayName.length; i++){  
    arrayName[i] = balioa;  
}
```

- Esleipen zuzenen bidez

```
arrayName = {elem1, elem2, elem3, ...};
```

# Adibideak: Array-ak (Matrizeak)

Erazagupena, Sorkuntza, Hasieraketa



## Oinarrizko datu-moten array-ak

```
int a[]; //Deklaratu
a = new int[3] //Sortu
a[0]=1; //Hasieratu
a[1]=2;
a[2]=3;
```

```
int a[] = new int[3] //Deklatu eta Sortu
a[0]=1; //Hasieratu
a[1]=2;
a[2]=3;
```

```
int a[] = new int[3] // Deklatu eta Sortu
for(int i=0; i<a.length;i++){ //hasieratu
    a[i]=i+1;
}
```

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

## Objektuzko array-ak (Erreferentziazko datu motak)

```
NireKlase a[]; //Deklaratu
a = new NireKlase [3] //Sortu
a[0]=new NireKlase (param1);
a[1]=new NireKlase (param2);
a[2]=new NireKlase (param3);
```

```
NireKlase a[] = new NireKlase [3]
//Hasieratu
a[0]=new NireKlase (param1);
a[1]=new NireKlase (param2);
a[2]=new NireKlase (param3);
```

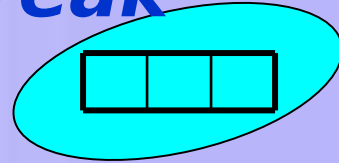
```
NireKlase a[] = new NireKlase [3]
//Hasieratu
for(int i=0; i<a.length;i++){
    a[i]=new NireKlase (param-i);
}
```

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



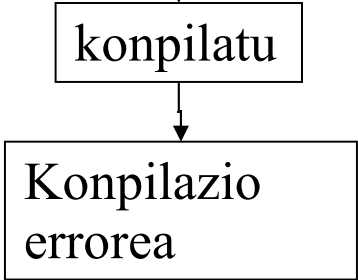
# Array-ak (Matrizeak): Ohizko erroreak

## Deklarazioa, Sorkuntza, Hasieraketa

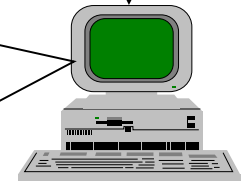


```
public class MatrizeAdibidea {  
    public static void main(String args[]) {  
        double nireTaula[];  
        System.out.println(nireTaula[0]);  
    }  
}
```

GAIZKI



variable nireTaula may not have been initialized



Matrizea **deklaratuta** dagoenean soilik, ezin daiteke beren elementuak atzitu. Programak ez du konpilatuko, eta **errore** bat agertuko da.

# Arrays (Matrices) Ohizko erroreak

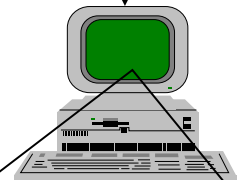
## Deklarazioa, Sorkuntza, Hasieraketa



```
public class MatrizeAdibidea2{
    public static void main(String args[]){
        int nireOsokoMatrizea[] = new int[10];
        float nireErrealekoMatrizea[] = new float[10];
        boolean nireBooleanokoMatrizea[] = new boolean[10];
        char nireKaraktereMatrizea[] = new char[10];
        String nireStringMatrizea[] = new String[10];
        Object nireObjektuMatrizea[] = new Object[10];
        System.out.println("Defektuzko osoa: " + nireOsokoMatrizea[0]);
        System.out.println("Defektuzko erreala: " + nireErrealekoMatrizea[0]);
        System.out.println("Defektuzko booleanoa: " + nireBooleanokoMatrizea[0]);
        System.out.println("Defektuzko karaktere: " + nireKaraktereMatrizea[0]);
        System.out.println("Defektuzko String-a: " + nireStringMatrizea[0]);
        System.out.println("Defektuzko Objektua: " + nireObjektuMatrizea[0]);
    }
}
```

konpilatu

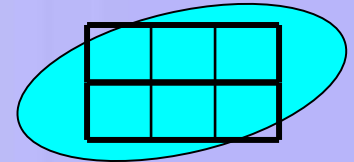
Exekutatu



Defektuzko osoa : 0  
Defektuzko erreala : 0.0  
Defektuzko booleanoa : false  
Defektuzko karaktere :  
Defektuzko String-a : null  
Defektuzko Objektua : null

Matrizea **deklaratuta eta sortuta** dagoenean baina **ez hasieratuta**, beren elementuek eskuragarri daude baina **bestarik ezeko balioa** edukiko dute

# Dimentsio anitzeko array-ak



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

	0	1	2
0	A	B	C
1	D	E	F
2	G	H	I

$A[0][2]='C'$

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

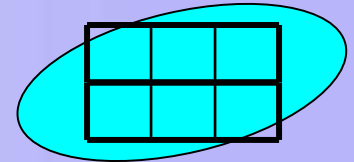
		0	1	2
1	2	r	s	t
0	0	a	b	c
1	1	d	e	f
2	2	g	h	i

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

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

# Dimentsio anitzeko array-ak

## Adibideak



### Deklaratu eta sortu batera

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

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

### Deklaratu eta sortu bi pausotan

```
int [][] nireMatrizea ; // array-a deklaratu  
nireMatrizea = new int[numFilas][]; // Erreferentziako array-a sortu
```

### Beste adibideak

```
// 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