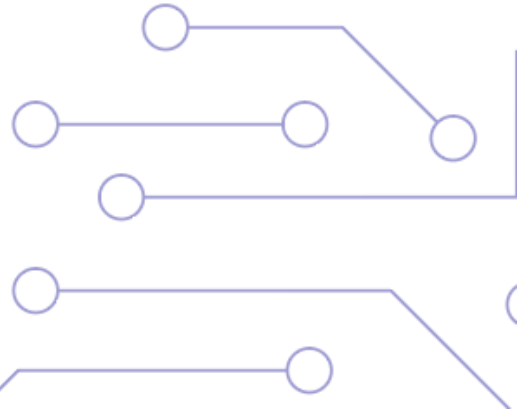
Programazioa

Datu-egitura estatikoak (Array-ak)

Datu-egiturak

- Datuak antolatu, kudeatu eta biltegiratzeko formatu bat eskaintzen dute
  - Gordetzen duten informazioa modu eraginkorrean sartzeko eta eraldatzeko aukera ematen dute
- Zehatzago esanda, datu-egitura bat datu-balioen bilduma bat da
  - Aldagai batek balio bakarra izan beharrean balio-multzo bat gordeko du
  - Balioez gain, barruan dituen datuen antolaketa eta haien arteko erlazioa definitzen ditu eta baita datu horiekin egin ahal izango diren eragiketen multzoa



Datu-egitura estatikoak: array-ak

- Array bat tamaina ezagun bat duen elementuen lista bat da
  - Indizeak erabilita zerrendako edozein elementu atzitu daiteke
- Array bat Java objektu bat da (berezko funtzioak ditu) eta gorde ditzakeen elementuak oinarrizko datu-motak edo Java objektuak izan daitezke
- Kortxeteekin [ ] identifikatzen dira
  - Adibidez, nireArray [ ] array-ak zenbaki osoak gordetzen ditu

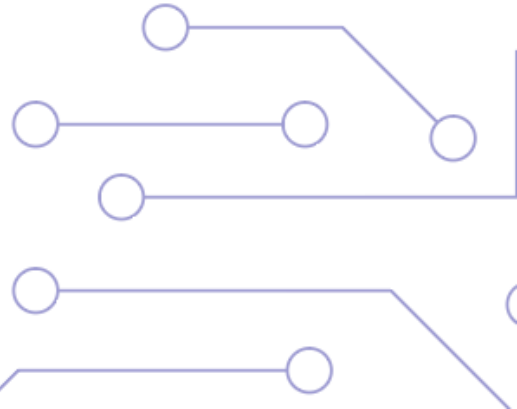
Adi: Array batek gordetzen dituen elementu guztiak mota berdinekoak izan behar dira

Array-ak: definizioa

- Array bat definitzeko barnean gordeko duen datu-mota edo objektua adierazi beharko da

```
int[] nireZenbakiak;
Double[] zenbakiObjektuak;
```

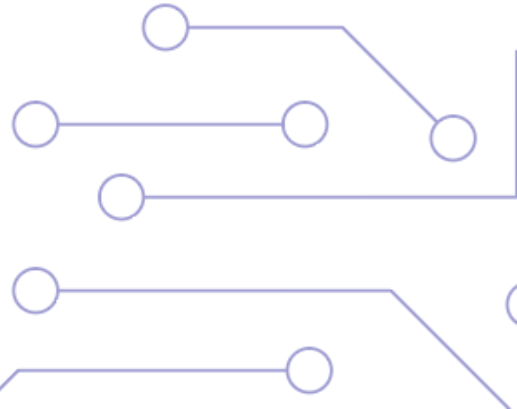
- Array bat sortzeko bi modu daude:
  - **new** hitz erreserbatua erabiltzea (tamaina adierazi behar da)
  - Zuzenean gordeko dituen balioen bilduma esleitzea



Array-ak: sorrera

- Array bat **new** hitz erreserbatua erabiliz sortzea
  - Tamaina adierazi behar da
  - Array-aren elementu bakoitzak lehenetsitako balioa izango du
  - 0 edo false oinarrizko datu-moten kasuan eta *null* (balio hutsa) String eta objektuen kasuan

```
int[] nireZenbakiak = new int[4];
String[] karaktere_kateak = new String[10];
```

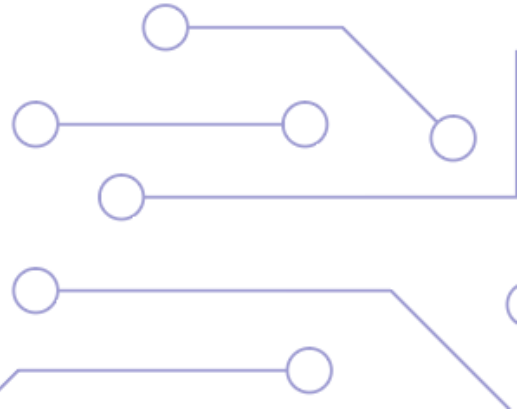


Kontuz: Array bati tamaina bat esleitzen zaionean ezingo zaio hau aldatu

Array-ak: sorrera

- Array bat gordeko dituen balioen bilduma esleituz sortzea
  - Ez da tamaina adierazten, sartzen zaizkion elementuen kopuruak zehazten du

```
int[] nireZenbakiak = {3,5,7,1,0,0};
String[] karaktere_kateak = {"txakurra", "", new String(), "etxe"};
```



Array-ak: atzipena

- Array baten balio bat atzitzeko bere posizioaren indizea erabili daiteke
  - ADI: Indizeak 0tik hasten dira
- Horrela atzituz posizio horretako balioa edo objektua jaso daiteke eta balio berri bat ere esleitu daiteke
- Tamaina baino handiagoa den posizio bat atzitzeak *IndexOutOfBoundException* errorea emango du

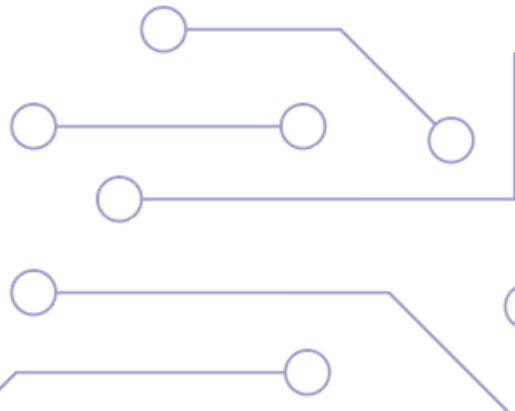
```
int[] nireZenbakiak = {3,5,7,1,0,0};
int zenb = nireZenbakiak[2];
System.out.println(zenb)

> 7
```

```
String[] kateak = new String[6];
kateak[0] = "txakurra";
kateak[1] = new String();
kateak[2] = "";
```

Array-ak: luzera funtzioa

- Array baten luzera jakiteko bere length atributua erabili daiteke
  - Irakurketarako bakarrik balio du, ezin da bertan idatzi!
- Array baten elementuak **0**-tik **length – 1**-era doaz
  - Array baten elementuak begizta baten bidez oso erraz aztertu daitezke

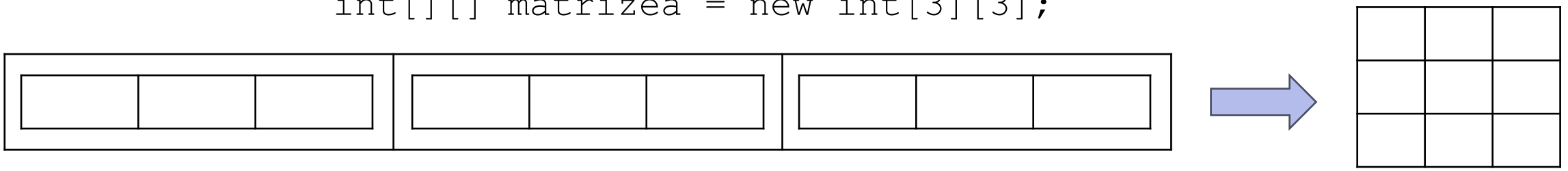
```
int[] nireZenbakiak = {3,5,7,1,0,0};
for (int i = 0; i < nireZenbakiak.length; i++)
{
    System.out.println(nireZenbakiak[i]);
}
```



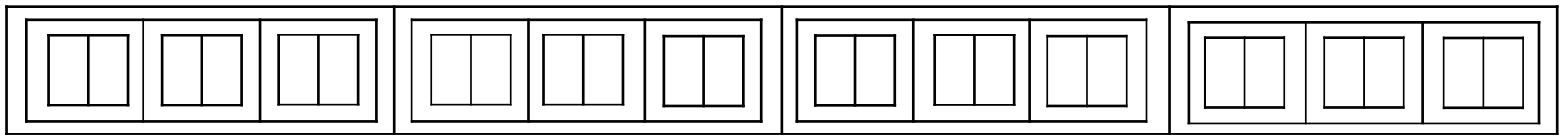
Dimentsio anitzeko array-ak

- Elementu bezala array-ak dituzten array-ak dira (matrizeak ere deituak)
  - Azken array-aren elementuak bai izango dira oinarrizko datu-motak edo objektuak
- Definizioa array baten antzeko moduan egiten da eta sortzeko osatzen duten array-en tamaina adierazi behar da

```
int[][] matrizea = new int[3][3];
```



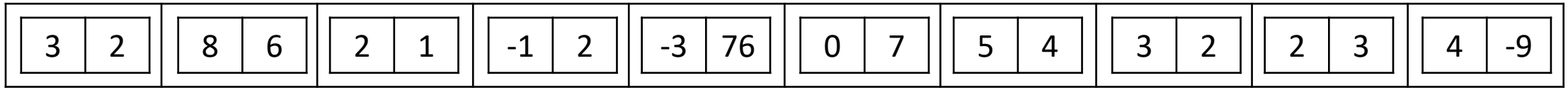
```
String[][][] nireKateak = new String[4][3][2];
```



Dimentsio anitzeko array-ak: atzipena

- Dimentsio bakoitzeko 0-tik hasten den indize bat adierazi behar da
  - Indizeak ezkerretik hasita dimentsio handienetik zehatzenera adierazten dira

```
int[][] matrizea = new int[10][2];
```

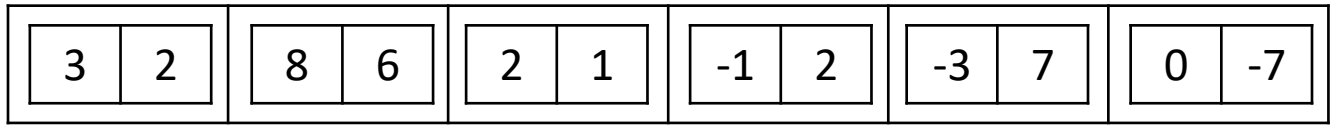



```
matrizea[3] -> {-1,2}
matrizea[0][0] -> 3
matrizea[6][1] -> 4
matrizea[1][5] -> Errorea
matrizea[matrizea.length][0] -> Errorea
matrizea[10][1] -> Errorea
matrizea[matrizea.length - 1] -> {4,-9}
matrizea[7][matrizea.length - 1] -> 2
```

Dimentsio anitzeko array-ak: atzipena

- Dimentsio bakoitzeko for bat behar da balioak inprimatzeko

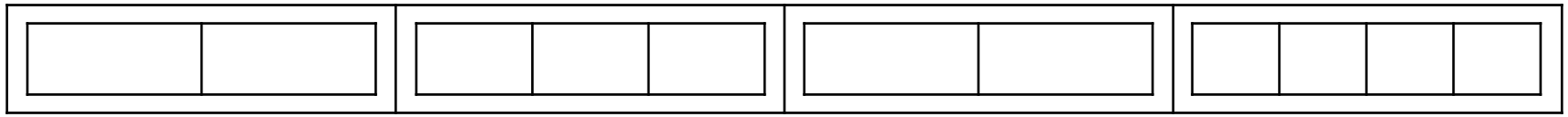
```
int[][] matrizea = {{3,2},{8,6},{2,1},{-1,2},{-3,7},{0,-7}};
for (int i = 0; i < matrizea.length; i++){
    for(int j = 0; j < matrizea[0].length; j++){
        if (j != matrizea[0].length - 1){
            System.out.print(matrizea[i][j] + ", ");
        }
        else{
            System.out.print(matrizea[i][j]);
        }
    }
    System.out.println();
}
```



Dimentsio anitzeko array-ak: irregularrak

- Dimentsio anitzeko array-etan beharrezkoa den tamaina bakarra orokorrena den (ezkerrerago dagoena) array-arena da
  - Honek tamaina desberdineko array-ak sortzea ahalbidetzen du

```
int[][] zenbakiak = new int[4][];
zenbakiak[0] = new int[2];
zenbakiak[1] = new int[3];
zenbakiak[2] = new int[2];
zenbakiak[3] = new int[4];
```



Dimentsio anitzeko array-ak: irregularrak

- Zer ateratzen du hurrengo programak?

```
int[ ][ ] mat;
int i, j;
mat = new int[10][];
for (i=0; i < mat.length; i++) {
    mat[i] = new int[i+1]; ints
}
for (i=0; i<mat.length; i++) {
    for (j=0; j<mat[i].length; j++) {
        mat[i][j] = i * j;
    }
}
for (i=0; i<mat.length; i++) {
    for (j=0; j<mat[i].length; j++) {
        System.out.print(mat[i][j] + " ");
    }
    System.out.println();
}
```