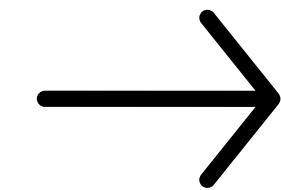


AEA2

Vectors i matrius



Prat

DAM

AEA2

Nom de l'AEA: Vectors i matrius	Hores totals AEA: 39 hores				
Descripció: L'alumnat aprenrà a declarar, inicialitzar i operar amb vectors i matrius,					
Resultat d'Aprendentatge					
RA2: Escriu i prova programes senzills, reconeixent i aplicant els fonaments de la programació orientada a objectes. RA3: Escriu i depura codi, analitzant i utilitzant les estructures de control del llenguatge. RA5: Realitza operacions d'entrada i sortida d'informació, utilitzant procediments específics del llenguatge i llibreries de classes. RA6: Escriu programes que manipulin informació seleccionant i utilitzant tipus avançats de dades.					
Organitzadors previs: Per desenvolupar aquesta AEA necessitem haver assolits els criteris d'avaluació de les AEA anteriors					
Seqüència de l'activitat d'ensenyament-aprenentatge	Organització aula	Temporització	Metodologia	Instruments d'avaluació	Localització
AA1 - Inicialització de parells 10%	Individual	5	ABP	Rúbrica	Aula
AA2 - Mitjana aritmètica 15%	Individual	10	ABP	Rúbrica	Aula
AA3 - Excel·lent 15%	Individual	5	ABP	Rúbrica	Aula
AA4 - Algú ha suspès? 15%	Individual	5	ABP	Rúbrica	Aula

AEA2

Nom de l'AEA: Vectors i matrius			Hores totals AEA: 39 hores		
Seqüència de l'activitat d'ensenyament-aprenentatge	Organització aula	Temporització	Metodologia	Instruments d'avaluació	Localització
AA5 - Trobar màxim i mínim 15%	Individual	5	ABP	Rúbrica	Aula
AA6 - Canviar valors matriu 15%	Individual	5	Aprenentatge cooperatiu	Rúbrica	Aula
AA7 - Major i menor element d'una matriu 15%	Individual	4	Aprenentatge cooperatiu	Rúbrica	Aula

Tipus de dades compostes

Tipus de dades compostes

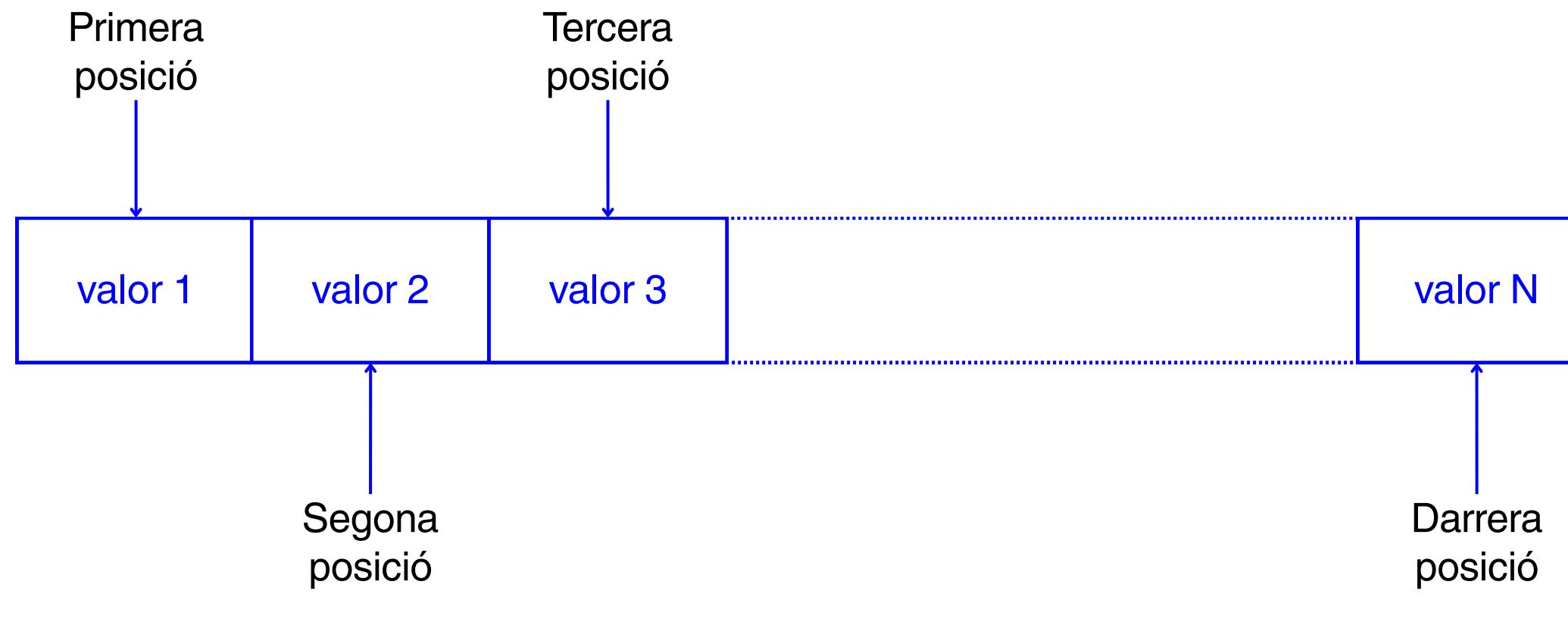
Un tipus de dada compost és aquell que permet emmagatzemar més d'un valor dins d'una única variable.

En el cas de l'array, aquest permet emmagatzemar una quantitat predeterminada de valors pertanyents al mateix tipus de dades.

Arrays o vectors

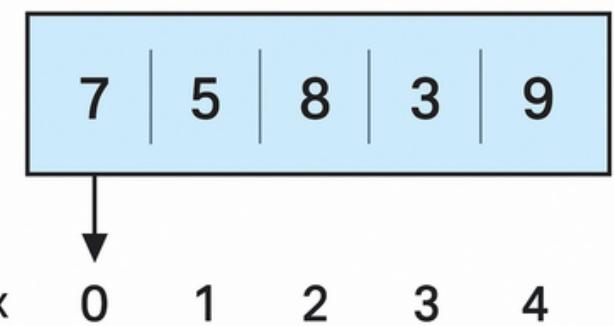
Arrays

Un vector (o array) és una estructura de dades que permet emmagatzemar un conjunt d'elements del mateix tipus en una sola variable. Cada element té una posició dins del vector, anomenada índex, que comença en 0.



Exemple

`int[] nombres`

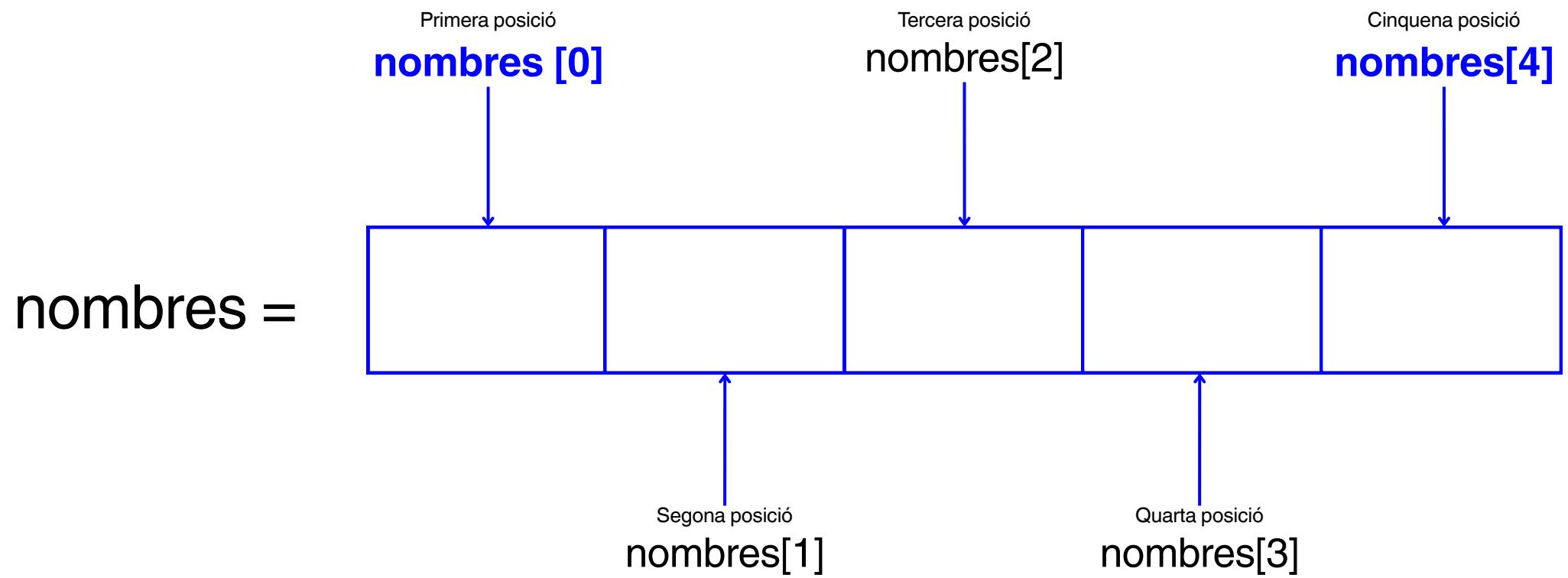


Arrays

Exemple

```
int[] nombres = new int[5];
```

Tipus de dada de l'array



Creació d'arrays



Inicialització a un valor concret

Li assignem un valor concret al vector

```
int[] numeros = {1, 2, 3, 4, 5};  
String[] noms = {"Anna", "Joan", "Marc"};
```

Inicialització a un valor per defecte

Creem el vector però no li assignem cap valor inicial, ho farem un cop el programa ja hagi començat.

```
int[] edats;           // Declaració  
edats = new int[10]; // Creació amb 10 posicions
```

=

```
double[] notes = new double[5];
```

important: Cal afegir la paraula **new** per a crear un vector sense incialitzar

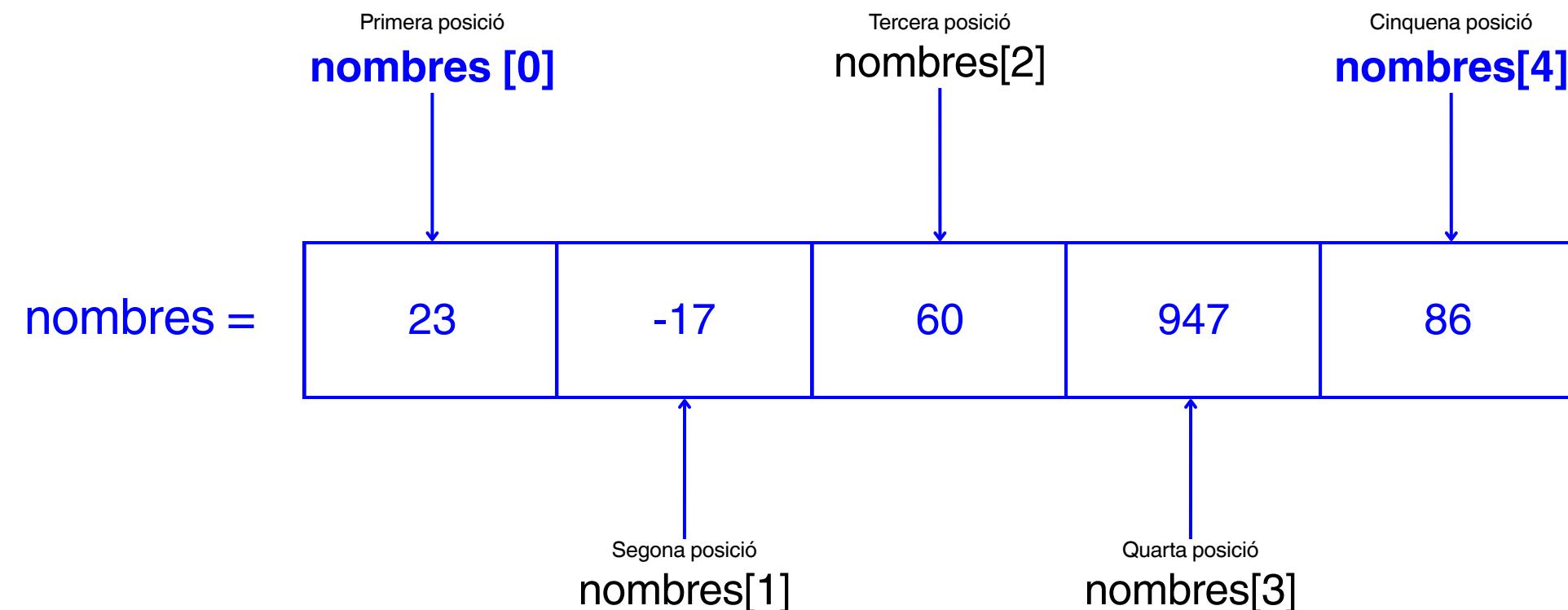
Accés als elements de l'array

Els elements del vector es poden llegir o modificar amb l'índex:

```
int primer = numeros[0]; // Accés al primer element  
numeros[2] = 10; // Canvia el tercer element a 10
```

Exemple

```
int[] nombres = {23,-17,60,947,86};
```



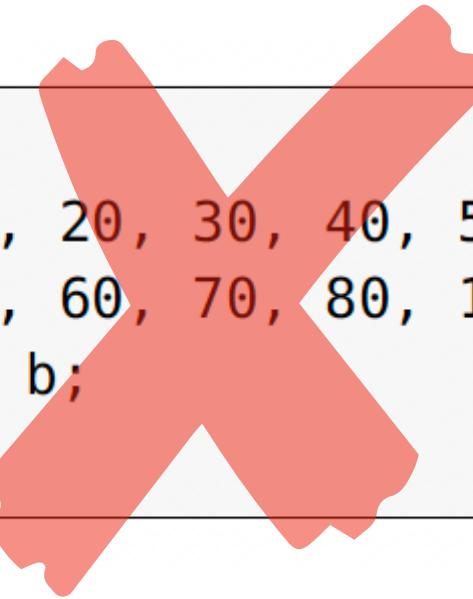
L'índex va de 0 a longitud - 1. Si intentes accedir a un índex fora del rang, Java llença un `ArrayIndexOutOfBoundsException`

- `nombres [0] = 23`
- `nombres [1] = -17`
- `nombres [2] = 60`
- `nombres [3] = 947`
- `nombres [4] = 86`

Manipulació d'arrays

Els arrays són una mica especials a l'hora de ser manipulats, ja que no disposen de cap operació. Per exemple, **NO ÉS POSSIBLE FER EL SEGÜENT:**

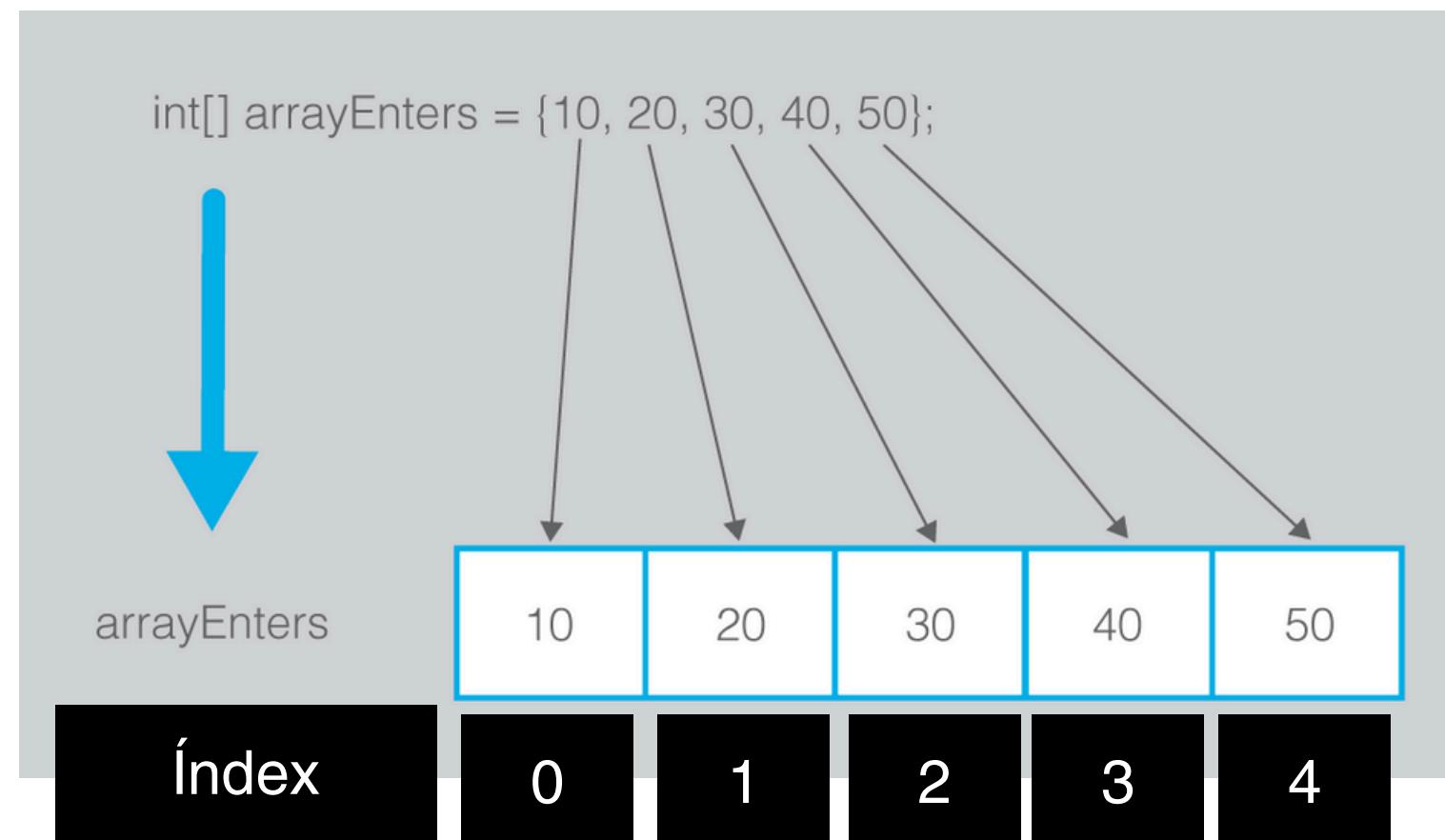
```
...
int[] a = {10, 20, 30, 40, 50};
int[] b = {50, 60, 70, 80, 100};
int[] c = a + b;
...
```



Sempre que es vulgui fer alguna **operació amb les dades emmagatzemades dins d'arrays cal manipular-les de manera individual**, posició per posició. En aquest aspecte, **cada posició d'un array té exactament el mateix comportament que una variable tal com les heu estudiades fins ara**.

Manipulació d'arrays

Per tal de distingir entre les **diferents posicions** d'un array, cadascuna té **assignat un índex**, un valor enter que n'indica l'ordre dins de l'estructura.



Exercici de classe - Registre notes

Digueu la mida i els index de cada array.

⚠️ **Els índex comencen sempre en 0.** És a dir, la mida del vector sempre serà mida -1

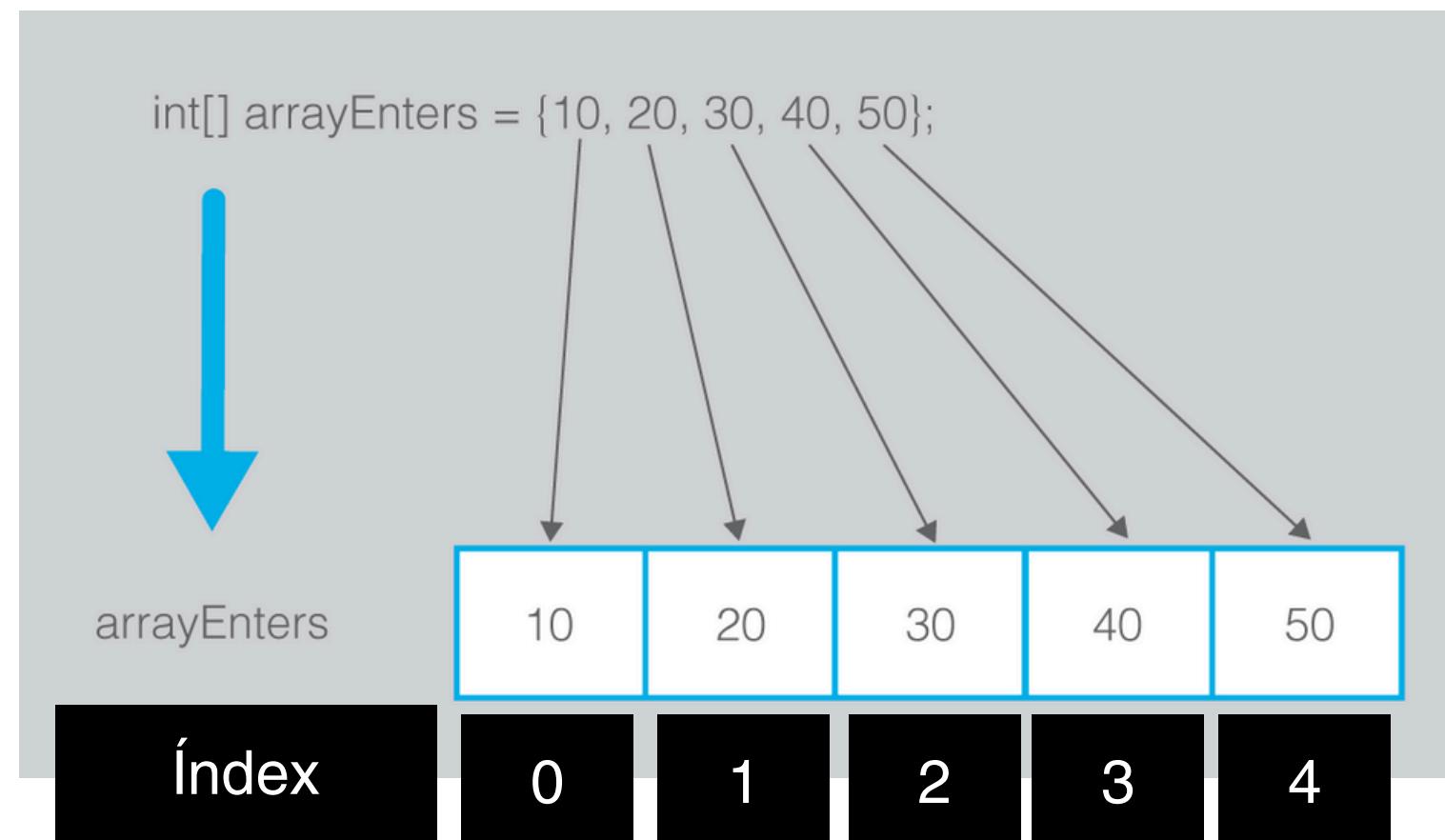
Lletres[?] =
A D R

Array

Decimals[?] =
1.1 4.7 2.3 3.1 5.6 9.1 2.5

Manipulació d'arrays

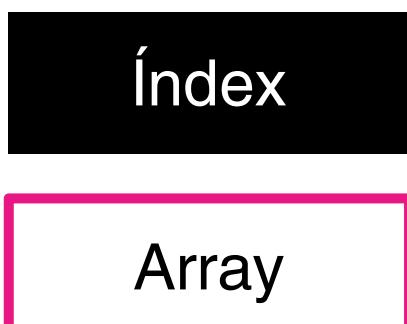
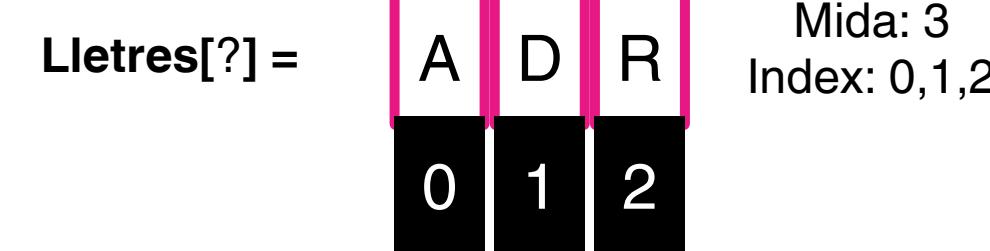
Per tal de distingir entre les **diferents posicions** d'un array, cadascuna **té assignat un índex**, un valor enter que n'indica l'ordre dins de l'estructura.



Exercici de classe - Registre notes

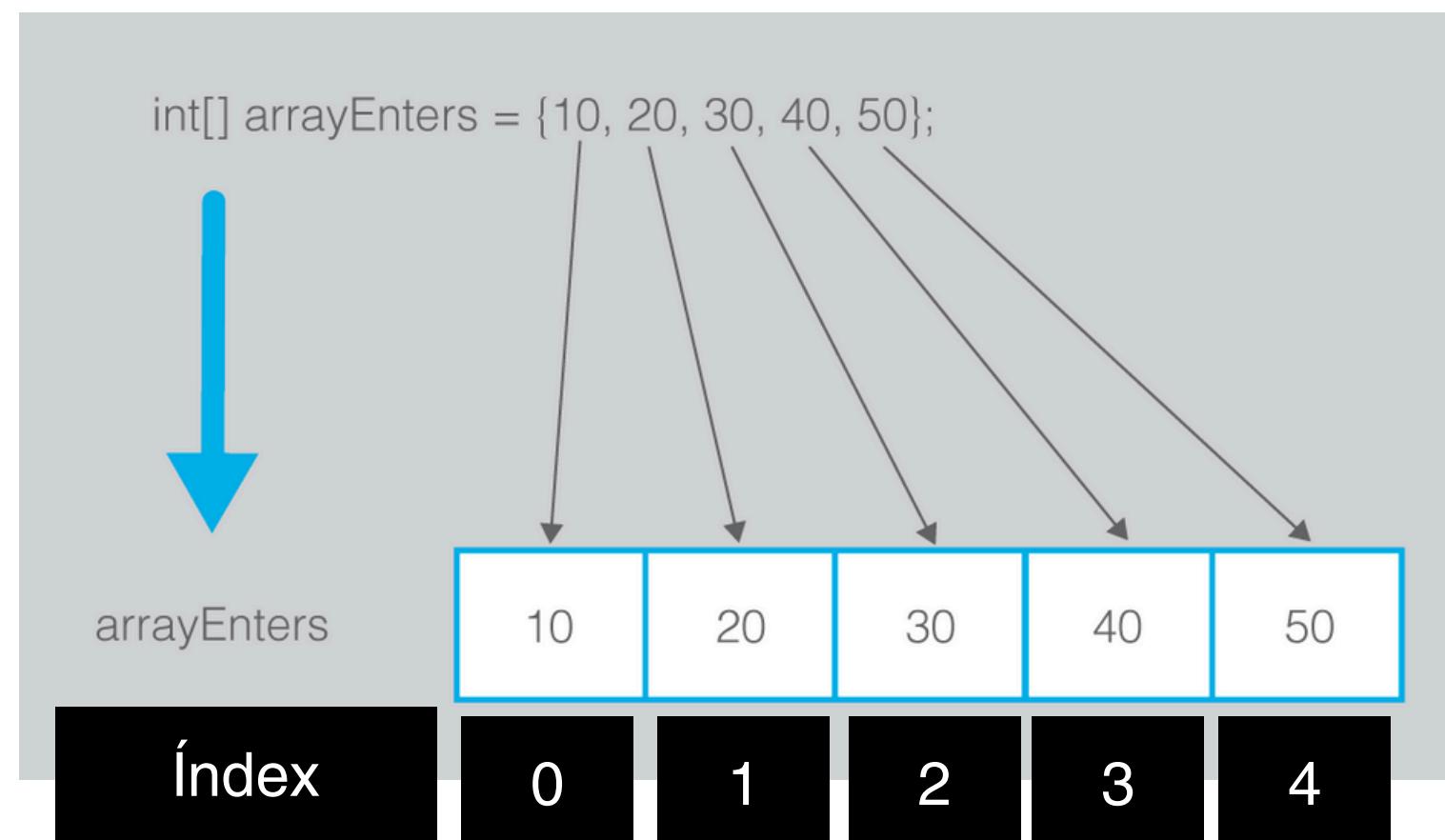
Digueu la mida i els index de cada array.

⚠️ **Els índex comencen sempre en 0.** És a dir, la mida del vector sempre serà mida -1



Manipulació d'arrays

Per tal de distingir entre les **diferents posicions** d'un array, cadascuna té **assignat un índex**, un valor enter que n'indica l'ordre dins de l'estructura.



Exercici de classe - Registre notes

Digueu la mida i els index de cada array.

⚠️ **Els índex comencen sempre en 0.** És a dir, la mida del vector sempre serà mida -1

Lletres[?] =

A	D	R
0	1	2

Mida: 3
Index: 0,1,2

Índex

Array

Decimals[?] =

1.1	4.7	2.3	3.1	5.6	9.1	2.5
0	1	2	3	4	5	6

Mida: 7
Index: 0,1,2,3,4,5,6

Manipulació d'arrays

Exemple

Quins són els valors que estem sumant?

```
...
int[] arrayA = {10, 20, 30, 40, 50};
int[] arrayB = {50, 60, 70, 80, 100};
int[] arrayC = new int[5];
//La variable de tipus enter "r" valdrà 140, 40 + 100.
int r = arrayA[3] + arrayB[4];    Suma de posicions d'un array
...

```

Manipulació d'arrays

Exemple

Quins són els valors que estem sumant?

arrayA[5] =

10	20	30	40	50
0	1	2	3	4

arrayB[5] =

50	60	70	80	100
0	1	2	3	4

```
...
int[] arrayA = {10, 20, 30, 40, 50};
int[] arrayB = {50, 60, 70, 80, 100};
int[] arrayC = new int[5];
//La variable de tipus enter "r" valdrà 140, 40 + 100.
int r = arrayA[3] + arrayB[4];    Suma de posicions d'un array
```

Manipulació d'arrays

Exemple

Quins són els valors que estem sumant?

arrayA[5] = 10 20 30 40 50

0 1 2 3 4

arrayB[5] = 50 60 70 80 100

0 1 2 3 4

```
...
int[] arrayA = {10, 20, 30, 40, 50};
int[] arrayB = {50, 60, 70, 80, 100};
int[] arrayC = new int[5];
//La variable de tipus enter "r" valdrà 140, 40 + 100.
int r = arrayA[3] + arrayB[4]; Suma de posicions d'un array
```

Manipulació d'arrays



Exercici de classe - Manipulació d'arrays

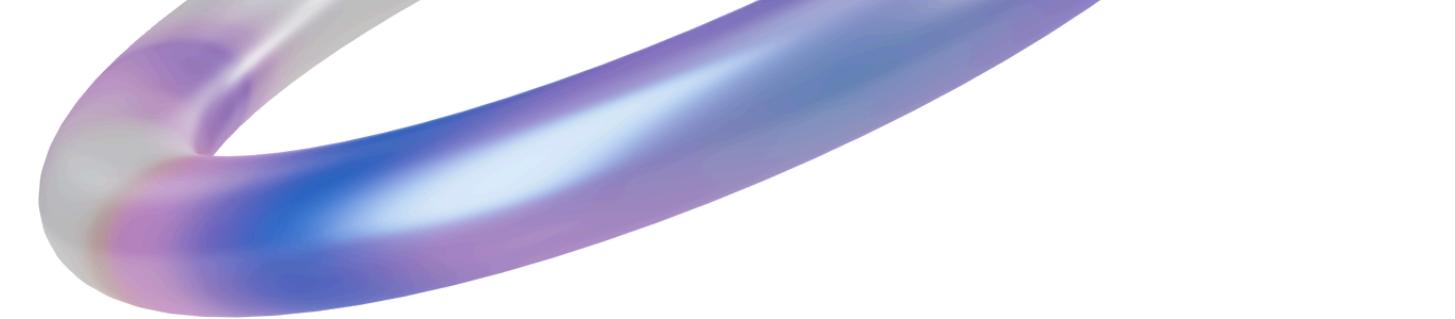
És correcte aquest codi?

```
...
int[] arrayA = {10, 20, 30, 40, 50};

arrayA[5] = 60;

int r = arrayA[2] + arrayA[4] + arrayA[6];
...
```

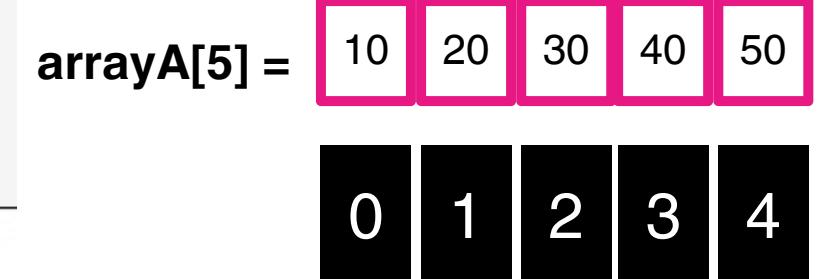
Manipulació d'arrays



Exercici de classe - Manipulació d'arrays

És correcte aquest codi?

```
...
// "arrayA" té mida 5.
// Els índexs vàlids van de 0 a (mida - 1), que és 4.
int[] arrayA = {10, 20, 30, 40, 50};
// S'assigna un valor a una posició incorrecta (índex 5).
arrayA[5] = 60;
// L'índex 6 és invàlid, no es pot consultar el valor.
int r = arrayA[2] + arrayA[4] + arrayA[6];
...
```



Longitud d'un vector



Poden extreure de forma simple la mida del vector. Per fer-ho cal:

```
nom vector  
int mida = numeros.length;  
System.out.println("La mida del vector és: " + mida);
```

Recórrer un vector

Recórrer un vector vol dir passar per tots els seus elements un per un, ja sigui per llegir-los, imprimir-los o modificar-los.

```
import java.util.Vector;

public class ExempleVector {
    public static void main(String[] args) {
        // Crear un vector de noms
        Vector<String> noms = new Vector<>();
        noms.add("Anna");
        noms.add("Bernat");
        noms.add("Carla");

        // Recórrer amb un for indexat
        for (int i = 0; i < noms.size(); i++) {
            System.out.println("Nom " + i + ": " + noms.get(i));
        }

        System.out.println("----");

        // Recórrer amb un for-each
        for (String nom : noms) {
            System.out.println("Nom: " + nom);
        }
    }
}
```

Recórrer un vector



While

```
// Recórrer amb un while
int i = 0; // Comencem pel primer índex
while (i < noms.size()) {
    System.out.println("Nom " + i + ": " + noms.get(i));
    i++; // Passar al següent element
}
```



For

```
// Recórrer amb un for indexat
for (int i = 0; i < noms.size(); i++) {
    System.out.println("Nom " + i + ": " + noms.get(i));
}
```

Mateix procediment, però amb
diferent loop

For-each

```
// Recórrer amb un for-each
for (String nom : noms) {
    System.out.println("Nom: " + nom);
}
```

El bucle for-each és útil per llegir
tots els elements, però no per
modificar-los directament.

Manipulació d'arrays

Exemple

Mireu el codi, enteneu què fa, copieu-lo i executeu-lo.

```
import java.util.Scanner;
public class exemple2 {

    public static void main (String[] args) {

        int[] arrayEnters = {1,2,3,4,5};
        int mida = arrayEnters.length;
        for (int i = 0; i < mida; i++) {
            int suma = arrayEnters[i] + 10;
            System.out.println("El resultat de sumar 10 + " + arrayEnters[i] + " = " + suma );
        }
    }
}
```

Manipulació d'arrays

Exercici a classe

💡 Quan ho tingueu, feu un programa que demani a l'usuari els valors del vector (MIDA = 5) i que, en comptes de sumar tots els valors, sumi només si l'índex és senar.

```
import java.util.Scanner;
public class exemple2 {

    public static void main (String[] args) {

        int[] arrayEnters = {1,2,3,4,5};
        int mida = arrayEnters.length;
        for (int i = 0; i < mida; i++) {
            int suma = arrayEnters[i] + 10;
            System.out.println("El resultat de sumar 10 + " + arrayEnters[i] + " = " + suma );
        }
    }
}
```

Manipulació d'arrays

Exemple

Mireu el codi, enteneu què fa, copieu-lo i executeu-lo.

```
import java.util.Scanner;
public class exemple4 {

    public static void main (String[] args) {
        int[] arrayEnters = new int[5];
        System.out.println("Escriu 5 nombres enters");

        Scanner s = new Scanner(System.in);

        int j = 0;
        //LOOP DE LECTURA DELS VALORS
        while (j < arrayEnters.length) {
            int num = s.nextInt();
            arrayEnters[j] = num;
            System.out.println("Estic escrivint: " + arrayEnters[j] + " i hauria d'escriure " + num);
            ++j;
        }

        for (int i = 0; i < arrayEnters.length; ++i){
            System.out.println(arrayEnters[i]);

        }
    }
}
```

Manipulació d'arrays

Exercici a classe

💡 Quan ho tingueu, feu un programa que:

- feu que el vector tingui 8 valors
- feu que sigui un vector de CHAR
- demani a l'usuari els valors del vector
- En cas que el CHAR sigui una vocal, canvieu el valor a “0”

```
import java.util.Scanner;
public class exemple4 {

    public static void main (String[] args) {
        int[] arrayEnters = new int[5];
        System.out.println("Escriu 5 nombres enters");

        Scanner s = new Scanner(System.in);

        int j = 0;
        //LOOP DE LECTURA DELS VALORS
        while (j < arrayEnters.length) {
            int num = s.nextInt();
            arrayEnters[j] = num;
            System.out.println("Estic escrivint: " + arrayEnters[j] + " i hauria d'escriure " + num);
            ++j;
        }

        for (int i = 0; i < arrayEnters.length; ++i){
            System.out.println(arrayEnters[i]);

        }
    }
}
```

Exercici per entregar



Inicialització de parells

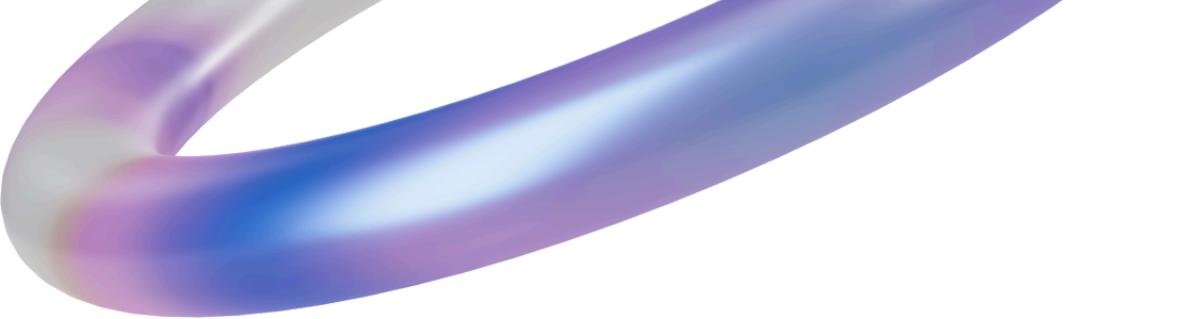
> Feu un programa que inicialitzi proceduralment un array amb els 100 primers nombres parells (Nota: el 0 és parell).

Sortida:

Array amb els 100 primers nombres parells:

```
0 2 4 6 8 10 12 14 16 18  
20 22 24 26 28 30 32 34 36 38  
40 42 44 46 48 50 52 54 56 58  
60 62 64 66 68 70 72 74 76 78  
80 82 84 86 88 90 92 94 96 98  
100 102 104 106 108 110 112 114 116 118  
120 122 124 126 128 130 132 134 136 138  
140 142 144 146 148 150 152 154 156 158  
160 162 164 166 168 170 172 174 176 178  
180 182 184 186 188 190 192 194 196 198
```

Exercici per entregar



Mitjana aritmètica

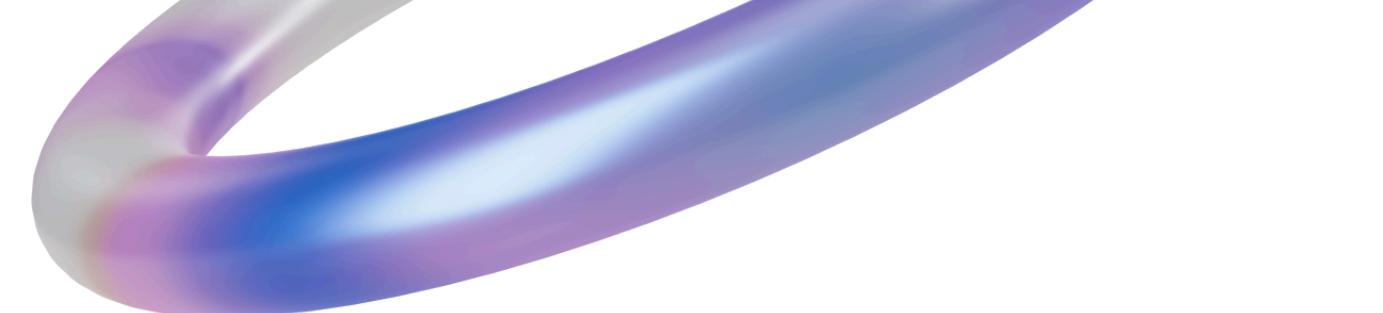
- > Feu un programa en Java que:
1. Demani a l'usuari quants valors vol introduir.
 2. Creï un vector amb la mida indicada.
 3. Llegeixi per teclat els valors reals (double) i els vagi desant dins del vector.
 4. Calculi la mitjana dels valors introduïts.
 5. Mostri per pantalla la mitjana obtinguda.

Sortida:

```
Quants valors vols introduir? 5
Introduceix el valor 1: 6.5
Introduceix el valor 2: 7.0
Introduceix el valor 3: 8.5
Introduceix el valor 4: 5.0
Introduceix el valor 5: 9.0

La mitjana dels valors és: 7.2
```

Exercici per entregar



Excel·lent

- > Feu un programa en Java que:
1. Demani a l'usuari quantes notes vol introduir.
 2. Creï un vector amb la mida indicada.
 3. Llegeixi per teclat les notes (valors enters o reals entre 0 i 10).
 4. Comprovi si algun alumne ha tret la nota màxima (10).
 5. Mostri per pantalla un missatge indicant si hi ha algun 10 o no.

Sortida:

```
Quantes notes vols introduir? 4
Introduceix la nota 1: 5
Introduceix la nota 2: 7
Introduceix la nota 3: 10
Introduceix la nota 4: 8

Sí, hi ha almenys un alumne amb un 10.
```

```
Quantes notes vols introduir? 3
Introduceix la nota 1: 4
Introduceix la nota 2: 6
Introduceix la nota 3: 9

No hi ha cap alumne amb un 10.
```

Exercici per entregar



Algú ha suspès?

- > Feu un programa en Java que:
1. Demani a l'usuari quantes notes vol introduir.
 2. Creï un vector amb la mida indicada.
 3. Llegeixi per teclat les notes (valors enters o reals entre 0 i 10).
 4. Comprovi si algun alumne ha suspès.
 - a. Utilitzeu la sentència *break*
 5. Si algú ha suspès, mostri per pantalla.

Sortida:

```
Quantes notes vols introduir? 5
Introduceix la nota 1: 7
Introduceix la nota 2: 4
Almenys un alumne ha suspès.
```

```
Quantes notes vols introduir? 3
Introduceix la nota 1: 6
Introduceix la nota 2: 7.5
Introduceix la nota 3: 10
Cap alumne ha suspès.
```

Exercicis per repassar



Exercicis que no s'entreguen

1. Feu un programa que tingui un vector de 5 paraules. Després, ha d'imprimir-lo.
2. Feu un programa que tingui un vector de 5 paraules inicials. Després, cal crear un altre vector amb paraules diferents.
Finalment, cal imprimir els dos vectos de paraules.
3. Feu un programa que tingui un vector de 12 chars. Feu que l'usuari inicialitzi el vector. Quan estigui tot el vector ple, reviseu que no hi hagi cap vocal, en cas que n'hi hagi, canvieu-la per “\$”. En cas que hi hagi una número, és a dir, entre el 0 i el 9, cal que es canviï el valor a “*”

Còpia d'arrays



Quan programem sentències de l'estil...

arrayA = arrayB

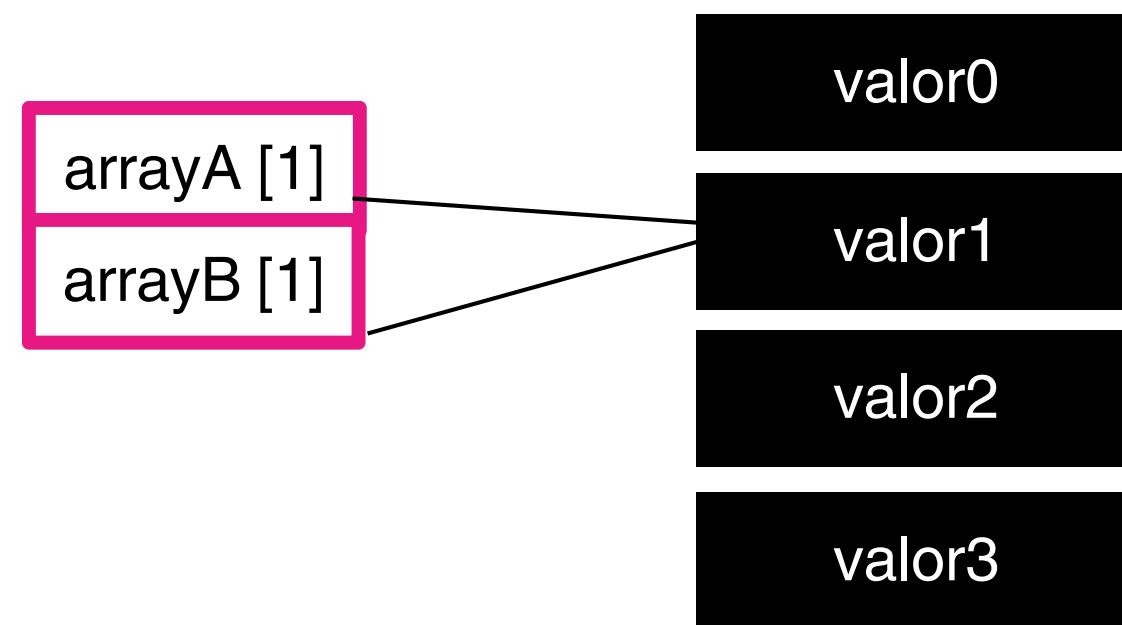
hem de vigilar molt i tenir molt clar què estem fent.

1. Aquesta sentència és correcta 
2. **No es crea una còpia**, és a dir, tenim arrayA i arrayB que accedeixen a les mateixes dades a la memòria del pc.

Còpia d'arrays

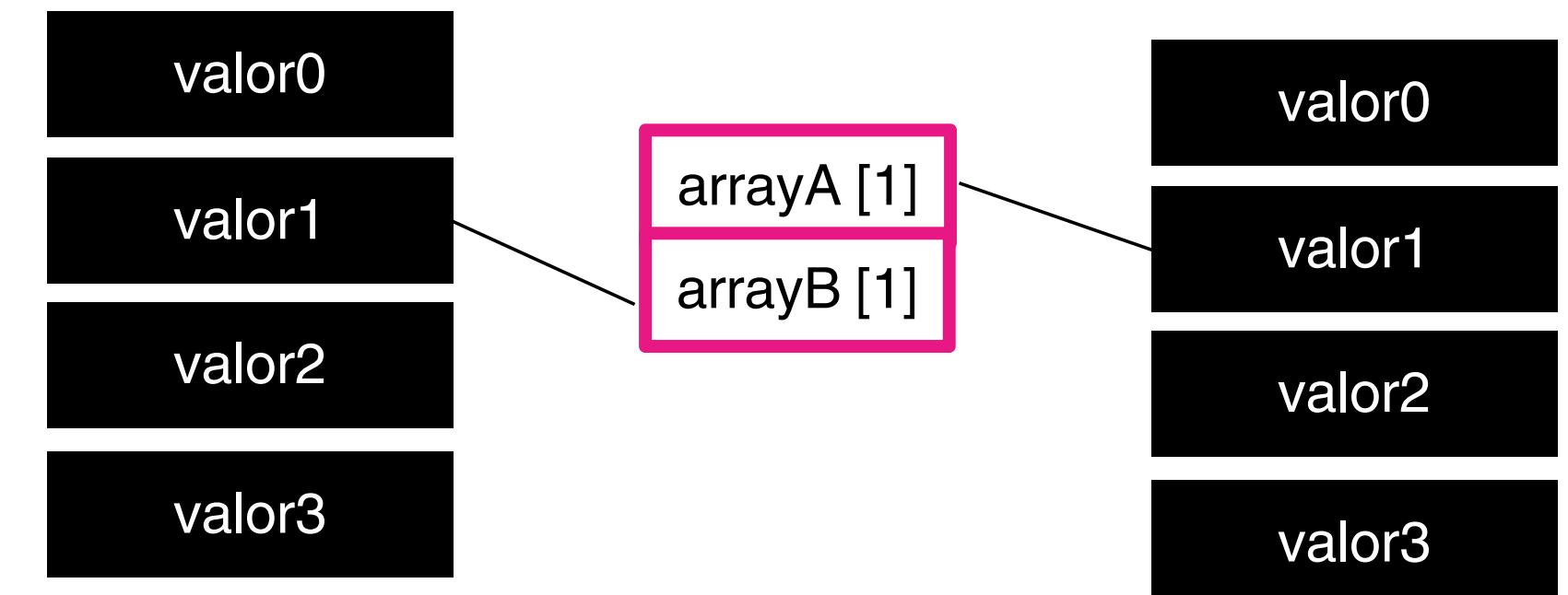
Exemple

arrayA = arrayB



Memòria PC

arrayA != arrayB



Memòria PC

Memòria PC

Còpia d'arrays

Exemple

Quant hauria de valer arrayB[2]? i arrayA[2]?

```
...
int[] arrayA = {10, 20, 30, 40, 50};
int[] arrayB = {60, 70, 80, 90, 100};

arrayA = arrayB;

arrayB[2] = 60;
```

Còpia d'arrays

Exemple

Quant hauria de valer arrayB[4]? i arrayA[4]?

```
...
int[] arrayA = {10, 20, 30, 40, 50};
int[] arrayB = {60, 70, 80, 90, 100};

arrayA = arrayB;

arrayB[2] = 60;

System.out.println(arrayA[2]);

arrayA[4] = 70;
```

Ordenació d'arrays

Ordenar un array vol dir col·locar els valors de dins de l'array en ordre per a que, després, sigui més fàcil buscar elements de qualsevol tipus,

Exemple

```
import java.util.Arrays;

public class OrdenarVector {
    public static void main(String[] args) {
        // Vector desordenat
        int[] numeros = {42, 7, 19, 3, 25, 10};

        System.out.println("Vector original: " + Arrays.toString(numeros))

        // Ordenació de menor a major
        Arrays.sort(numeros);

        System.out.println("Vector ordenat: " + Arrays.toString(numeros));
    }
}
```

```
import java.util.Arrays;
```

```
Arrays.sort(numeros); // Ordena el vector de menor a mayor
System.out.println(Arrays.toString(numeros));
```

Sortida

Vector original: [42, 7, 19, 3, 25, 10]
Vector ordenat: [3, 7, 10, 19, 25, 42]

Exercici per entregar



Trobar màxim i mínim

> Escriviu un programa que busqui el nombre més gran i el més petit dins el vector.

El vector és: {12, 3, 45, -3, 65, -5, 88, 99, 2, 23, 54, 33, 27, 53, 96, 0}

El vector és: {12, 3, 45, -3, 65, -5, 88, 99, 2, 23, 54, 33, 27, 53, 96, 0}

El nombre més petit del vector és: -5

El nombre més gran del vector és: 99

Exercicis per repassar



Exercicis que no s'entreguen

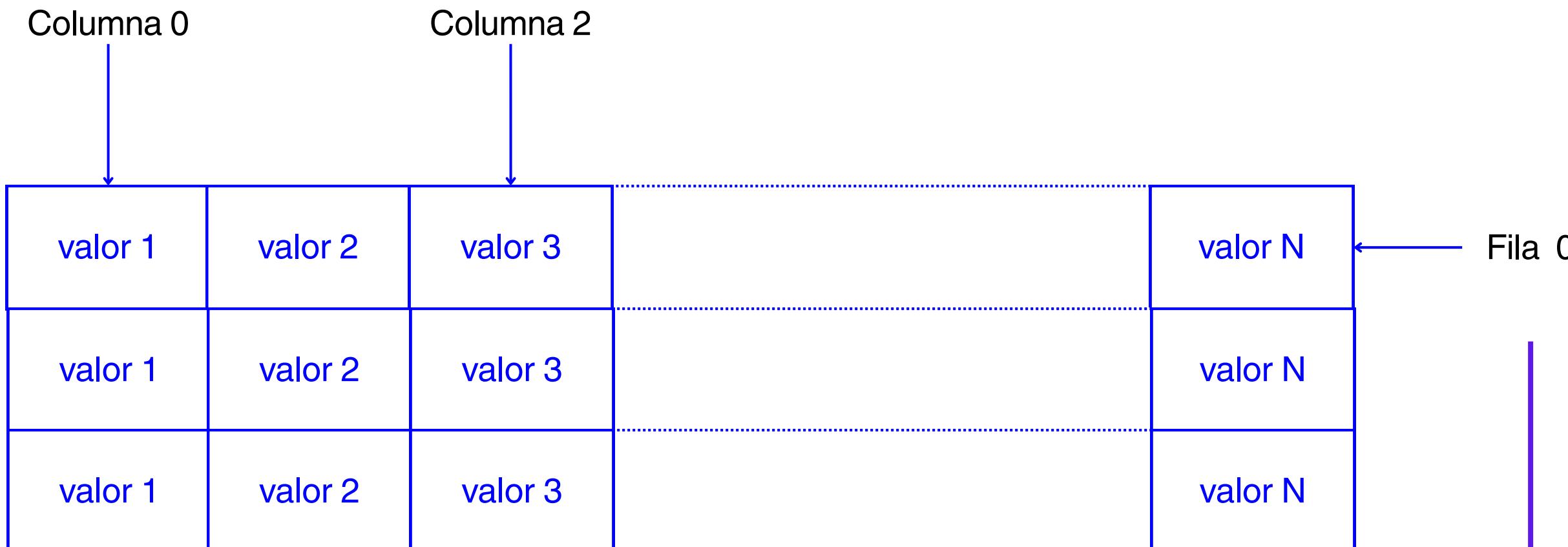
1. Escriviu un programa que escrigui el vector totalment al revés de com l'hem entrat. És a dir, si el vector és: {12, 3, 0}, volem que aparegui : {0, 3, 12}. El vector inicial ens el proporicona l'usuari.
2. Hem escrit un programa que busca el nombre més gran i el més petit dins el vector. Ara endrecem aquest vector amb l'algoritme bubble sort i tornem a buscar el valor mínim i màxim. El vector és: {12, 3, 45, -3, 65, -5, 88, 99, 2, 23, 54, 33, 27, 53, 96, 0}

Matrius

Matrius

Una matriu és com un vector de vectors.

Si un vector és una línia d'elements, una matriu és una taula de files i columnes on tots els elements són del mateix tipus (int, double, String...).



Exemple

1	2	3	4
5	6	7	8
9	10	11	12

Creació de matrius



Inicialització a un valor concret

Li assignem un valor concret al vector

```
int[][] matriu = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

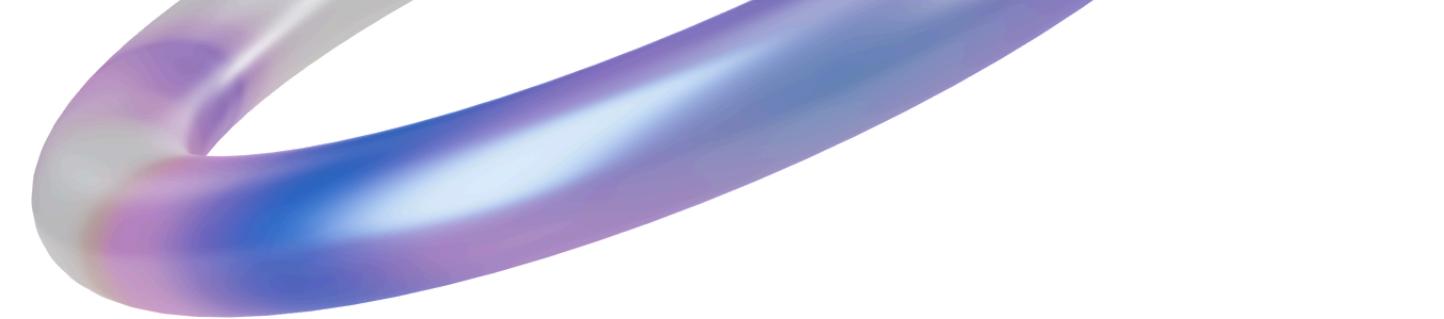
Inicialització a un valor per defecte

Creem el vector però no li assignem cap valor inicial, ho farem un cop el programa ja hagi començat.

```
int[][] matriu = new int[3][4]; // 3 files i 4 columnes
```

important: Cal afegir la paraula **new** per a crear un vector sense incialitzar

Accés als valors d'una matriu

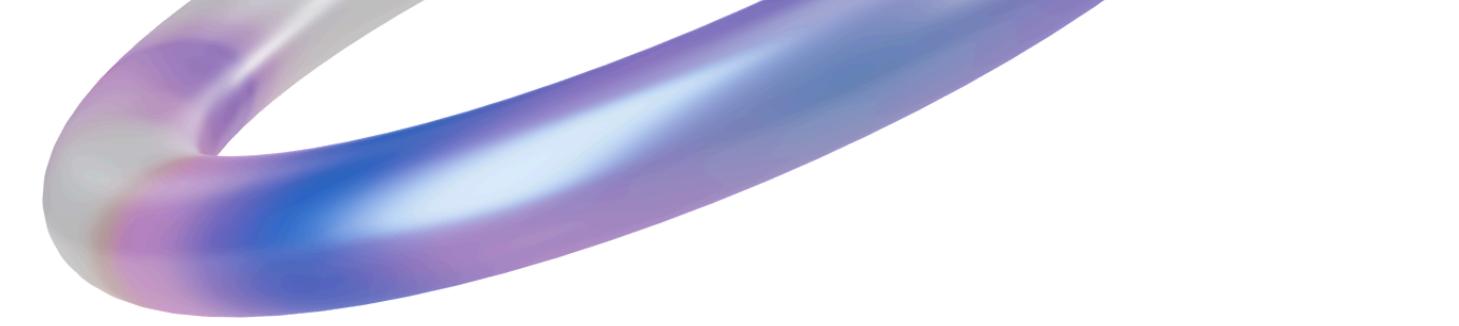


Per accedir a un element, cal indicar la fila i la columna:

Exemple

```
int valor = matriu[0][2]; // fila 0, columna 2 → retorna 3  
matriu[2][1] = 50; // fila 2, columna 1 → canvia el 10 per un 50
```

Recórrer una matriu



Ho farem gairebé sempre amb loops for, però es podria fer amb while.

Exemple

```
for (int i = 0; i < matriu.length; i++) {          // recórrer files
    for (int j = 0; j < matriu[i].length; j++) {    // recórrer columnes
        System.out.print(matriu[i][j] + " ");
    }
    System.out.println(); // salt de línia després de cada fila
}
```

Exemple complet

Exemple

```
public class ExempleMatrius {  
    public static void main(String[] args) {  
        int[][] matriu = {  
            {1, 2, 3},  
            {4, 5, 6},  
            {7, 8, 9}  
        };  
  
        System.out.println("Elements de la matriu:");  
        for (int i = 0; i < matriu.length; i++) {  
            for (int j = 0; j < matriu[i].length; j++) {  
                System.out.print(matriu[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Sortida

Elements de la matriu:

1 2 3

4 5 6

7 8 9

Exercici per entregar



Canviar valors matriu

- > Reutilitzeu la matriu que vam crear a l'exemple anterior, però ara feu que:
- Files parells i columnes parells —> el valor canvia a 1
 - Files parells i columnes senars —> el valor canvia a 2
 - Files senars i columnes parells —> el valor canvia a 3
 - Files senars i columnes senars —> el valor canvia a 4

Matriu d'entrada

1	2	3
4	5	6
7	8	9

Matriu de sortida

Matriu transformada:

1	2	1
3	4	3
1	2	1

Exercici per entregar



Major i menor element d'una matriu

> Fes un programa que:

- Llegeixi una matriu d'enters des del teclat.
- Mostri el nombre més gran i el més petit que conté.

La matriu és:

5 12 -3 7

9 0 22 15

4 -8 6 1

El nombre més petit de la matriu és: -8

El nombre més gran de la matriu és: 22

Exercicis per repassar



Exercicis que no s'entreguen

1. Reutilitzeu la matriu que vam crear a l'exemple anterior, però ara feu que en totes aquelles files i columnes parells el valor canviï a -1.
2. A partir d'una matriu de 3×3 d'enters:
 - a. Calcula la suma dels elements de cada fila.
 - b. Mostra el resultat fila per fila.



Gràcies

mmoreno@pratfp.com

Desenvolupament d'aplicacions multiplataforma

Prat