

BOLETIN A

ARRAYS DE UNA DIMENSION (VECTORES)

0. SOLUCIONADO. Diseñar algoritmo correspondiente a un programa para:

- a. Crear un **vector** de 10 elementos de tipo numérico entero y de nombre 'vector'.

```
final int NUMELEMENTOS = 10;  
int[] vector = new int[NUMELEMENTOS];
```

- b. Cargar el vector con los 10 primeros números impares

```
for (int i = 0, impares=1; i < vector.length; i++, impares+=2) {  
    vector[i] = impares;  
}
```

- c. Mostrar el contenido del vector en pantalla

```
for (int j = 0; j < vector.length; j++) {  
    System.out.print(vector[j] + "-");  
}
```

0. SOLUCIONADO. Diseñar algoritmo correspondiente a un programa que:

- a. Cree un **vector** de 6 elementos de tipo cadena de caracteres y de nombre 'alumnos'.
b. Carga el vector con los nombres y apellidos de los compañeros de tu fila.
c. Muestre el contenido del vector en pantalla (usa bucle for-each)

```
final int SIZE = 6, posicion=1;  
String[] alumnos = new String[SIZE];
```

```
alumnos[0]="Pepe";  
alumnos[1]="Juan";  
alumnos[2]="Sara";  
alumnos[3]="Luis";  
alumnos[4]="Ana Galvez";  
alumnos[5]="Sara Baras";
```

```
for (String nombre: alumnos) {  
    System.out.println(nombre);  
}
```

1. Diseñar algoritmo correspondiente a un programa que:

- a. Cree un vector de 20 elementos de tipo numérico entero y de nombre 'pares'.
b. Cargue el vector con los 20 primeros números pares de mayor a menor
c. Muestre el contenido del vector en pantalla (usa bucle for-each)

```
final int SIZE = 20;  
int[] pares = new int[SIZE];  
for (int i = 0, par=40; i < pares.length; i++, par-= 2) {  
    pares[i] = par;  
}  
for (int num: pares) {  
    System.out.print(num + "-");  
}
```

EJERCICIOS TEMA 3 ESTRUCTURAS DE ALMACENAMIENTO

2. Diseñar algoritmo correspondiente a un programa que:

- Cree una **vector** de 6 elementos de tipo numérico entero y de nombre 'edades'.
- Cargue la tabla con valores recogidos por teclado, representando a las edades de tus compañeros de fila
- Contabiliza la cantidad de menores y mayores de edad almacenados en el proceso de inicialización
- Muestre por pantalla cuantos has contabilizado.

```
final int SIZE = 6;
int i, mayores=0, menores;
int[] edades = new int[SIZE];
Scanner sc = new Scanner(System.in);

for (i = 0; i < edades.length; i++) {
    System.out.print("Dame una edad: ");
    edades[i] = Integer.parseInt(sc.next());
    if (edades[i] >= 18) mayores++;
}

System.out.println("La cantidad de mayores de edad: " + mayores);
menores = edades.length - mayores;
System.out.println("La cantidad de menores de edad: " + menores);
```

3. . Diseñar algoritmo correspondiente a un programa que:

- Cree una vector de 30 elementos de tipo numérico entero y de nombre 'vector'.
- Cargue la tabla con elementos aleatorios generados mediante:
`import java.util.Random;`
`Random aleatorio = new Random();`
`int numero = aleatorio.nextInt(999); //genera números entre 0 y 999`
- Calcule la suma de los 'N' primeros elementos de la tabla. El valor de N no debe superar el tamaño del vector.
- Muestre por pantalla el contenido de la tabla y al final la suma.

```
Scanner tec = new Scanner(System.in);
int[] vector = new int[30];
Random aleatorio = new Random();
int sumar = 0;
for (int i = 0; i < vector.length; i++) {
    vector[i] = aleatorio.nextInt(999);
}
System.out.println("Calcule la suma de los 'N' primeros elementos(Introduzca N)");
int num = Integer.parseInt(tec.next());
if (num <= vector.length) {
    for (int cont = 0; cont < num; cont++) {
        sumar = sumar + vector[cont];
        System.out.print(vector[cont] + " ");
    }
    System.out.println("\nLa suma de los " + num + " primeros numeros es " + sumar);
} else {
    System.out.println("N debe ser menor al tamaño del vector");
}
```

EJERCICIOS TEMA 3 ESTRUCTURAS DE ALMACENAMIENTO

4. Diseñar algoritmo en el que dado el siguiente vector de nombre 'origen':

1	4	10	32	60	91	34	56	73	88	86
---	---	----	----	----	----	----	----	----	----	----

- a. Lo recorra metiendo en un segundo **vector** de nombre 'destino' todos aquellos elementos impares mayores de X introducido por teclado. ¿Qué tamaño debería tener el vector 'destino'?
- b. Muestre por pantalla el contenido de los dos vectores

```
int origen [] = {1, 4, 10, 32, 60, 91, 34, 56, 73, 88, 86};
int num=0, i, x=0;

System.out.println(" Que valor deben superar los números");
int max = Integer.parseInt(sc.next());
for (i=0; i<origen.length; i++){
    if ((origen[i] > max)&&(origen[i]%2 != 0)){
        num++;
    }
}

int destino[]=new int[num]; //SE CREA EL VECTOR DEL TAMAÑO NECESARIO. ESTO SERIA
//EFICIENCIA ESPACIAL PORQUE OCUPA EL ESPACIO EN MEMORIA JUSTO

for (i=0; i< origen.length ;i++){
    if ((origen[i] > max)&&(origen[i]%2 != 0)){
        destino[x]= origen[i];
        x++;
    }
}

for (i=0; i< origen.length ;i++){
    System.out.print(origen[i]+" ");
}

System.out.println(" ");
for (i=0; i< destino.length ;i++){
    System.out.print(destino[i]+" ");
}
```

OTRA VERSION de mejor eficiencia temporal

```
Scanner sc = new Scanner(System.in);
int origen[] = {1, 4, 10, 32, 60, 91, 34, 56, 73, 88, 86};
int destino[] = new int[origen.length];
int num = 0, i, x = 0;

System.out.println(" Que valor deben superar los números");
int max = Integer.parseInt(sc.next());

//SOLO SE RECORRE UNA VEZ EL ARRAY ESTO SERIA EFICIENCIA TEMPORAL,
//SE TARDA MENOS TIEMPO EN PROCESAR LOS DATOS,
//PERO OCUPA MAS ESPACIO EN MEMORIA PORQUE HABRIAN HUECOS VACIOS
for (i = 0; i < origen.length; i++) {
    if ((origen[i] > max) && (origen[i] % 2 != 0)) {
        destino[x] = origen[i];
        x++;
    }
}

for (i = 0; i < origen.length; i++) {
    System.out.print(origen[i] + " ");
}

System.out.println(" ");
for (i = 0; i < destino.length; i++) {
    System.out.print(destino[i] + " ");
}
```

}

CLASE ARRAYS

5. Diseñar algoritmo correspondiente a un programa que:

- Declare un vector para almacenar los nombres de estas 7 personas:
- Muestra esos nombre ordenados alfabéticamente por pantalla
- Pide un nombre por teclado, búscalo e indica en qué posición lo has encontrado
- Declara e inicializa otro vector con el nombre "Pepe" 7 veces.
- Compara ambos vectores e indica si son iguales

```
String[] nombres = {"Zaira", "Luis", "Alberto", "Ana", "Andres", "Javier", "Laura"};
String[] nombres2 = new String[nombres.length];
String nombre = "Luis";
int posicion;
boolean iguales;

Arrays.sort(nombres);
System.out.println(Arrays.toString(nombres));

// imprescindible ordenar para usar binarySearch
Arrays.sort(nombres);
posicion = Arrays.binarySearch(nombres, nombre);
if (posicion < 0) {
    System.out.println("El nombre no existe");
} else {
    System.out.println("Esta en posición " + posicion);
}

Arrays.fill(nombres2, "Pepe");
System.out.println(Arrays.toString(nombres2));

iguales = Arrays.equals(nombres, nombres2);
if (iguales == true) {
    System.out.println("SON iguales");
} else {
    System.out.println("NO son iguales");
}
```

ARRAYS DE DOS DIMENSIONES (MATRICES)

USO DE System.out.PRINTF, ESCRITURA CON FORMATO

cada variable representada por → %[alineacion][huecos][letra_segun_tipo_datos]

[alineacion] → es opcional, simbolo - indica si se alinea a la izquierda, si no aparece se alinea a la derecha

Ejemplos → %d (variable de tipo entero)

%f (variable de tipo double)

%s (variable de tipo String)

int variable_int=123, variable2_int=2;

double variable_double=34.8765, variable2_double=1.53;

String variable_String="Pepe", variable2_String="Alberto";

System.out.printf("variable int que ocupa 12 huecos %-12d %12d \n", variable_int, variable2_int);

System.out.printf("variable double que ocupa 12 huecos %-12f %12f \n", variable_double, variable2_double);

System.out.printf("variable String que ocupa 12 huecos %-12s %12s \n", variable_String, variable2_String);

6. SOLUCIONADO. Diseñar algoritmo correspondiente a un programa que:

- Cree una **tabla de dos dimensiones** y tamaño 8x12 (8 filas x 12 columnas) cuyo nombre es tabla
- Cargue la tabla de tal forma que cada elemento contenga su número de columna
- Muestre el contenido de la matriz en pantalla

Output - Pruebas (run) ×

0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	3	4	5	6	7	8	9	10	11

BUILD SUCCESSFUL (total time: 0 seconds)

```
int[][] tabla = new int[8][12];
int i = 0, j = 0;

for (i = 0; i < tabla.length; i++) {
    for (j = 0; j < tabla[i].length; j++) {
        tabla[i][j] = j;
        System.out.printf( "%4d",tabla[i][j] );
    }
    System.out.println();
}
```

EJERCICIOS TEMA 3 ESTRUCTURAS DE ALMACENAMIENTO

7. Diseñar algoritmo correspondiente a un programa que:

- Cree un **array de dos dimensiones** y longitud 4x6 (4 filas x 6 columnas) cuyo nombre es producto
- Cargue la matriz de tal forma que cada elemento contenga la suma de su fila y su columna
- Muestre el contenido de la matriz en pantalla. Mostrando numero de cada fila y columna

```
run:
      0      1      2      3      4      5
-----
0->    0      1      2      3      4      5
1->    1      2      3      4      5      6
2->    2      3      4      5      6      7
3->    3      4      5      6      7      8
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
int[][] suma = new int[4][6];
int fila = 0, columna = 0;

System.out.print("      ");
for (columna = 0; columna < suma[fila].length; columna++) {
    System.out.printf("%5d", columna);
}
System.out.println("");
System.out.println("      -----");

for (fila = 0; fila < suma.length; fila++) {
    System.out.printf(fila + "-> ");
    for (columna = 0; columna < suma[fila].length; columna++) {
        suma[fila][columna] = fila + columna;
        System.out.printf("%5d", suma[fila][columna]);
    }
    System.out.println();
}
```