RECORRIDO DE MATRICES

Recorrer los elementos de una matriz para asignar valores o ver su contenido.

Al igual que con las matrices simples o unidimensionales, también podemos emplear bucles for para asignar o ver los datos de una matriz multidimensional:

Asignar datos a una matriz multidimensional:

```
int lim_0 = notas[ 0 ].length ;

int lim_1 = notas[ 1 ].length ;


for ( int  k = 0;  k < lim_0 ; k++ ) {

  for( int  l = 0 ; l < lim_1 ;  l++ ){

     notas  [ k  ][ l  ] = EntDatos.entero( );
```

```
  }
}
```

Ver datos de una matriz multidimensional:

```
                int lim_0 = notas[ 0 ].length ;
                int lim_1 = notas[ 1 ].length ;
for ( int k = 0 ;  k < lim_0 ; k++ ) {
  for( int  l = 0 ;  l <lim_1 ;  l++ ){
    System.out.println ( notas [ k  ] [ l  ] ) ;
  }
}
```

Como vemos, hemos declarado dos variables lim_0 y lim_1, para saber el tamaño de cada matriz unidimensional y establecer límites a los bucles. las variables «k» y «l» representan las filas y las columnas de la matriz bidimensional. Para asignar datos a la matriz hemos utilizado la clase EntDatos y su método entero( ) para introducir números enteros por teclado. Esta clase debe de estar en el mismo paquete, donde utilicemos la llamada a la misma, o en su defecto (no aconsejable) en la clase donde se situe el programa principal (contiene el método main( )) desde donde efectuemos la llamada al método de la clase EntDatos.

Podemos recorrer una matriz de varias formas. En cualquier caso, al ser un recorrido de dos dimensiones, necesitaremos dos índices:

Por filas y columnas

```
public class RecorrerMatrizPorFilasYColumnas {
public static void main(String[] args) {
int[][] matriz = new int[50][100]; // Matriz de números enteros que supondremos llena.
                                   // 50 filas y 100 columnas.
```

```
for (int i = 0; i < 50; i++)            // El primer índice recorre las filas.
for (int j = 0; j < 100; j++){  // El segundo índice recorre las columnas.
                                        // Procesamos cada elemento de la matriz.
System.out.println(matriz[i][j]);
                        }
        }
}
```

Por columnas y filas

```
public class RecorrerMatrizPorColumnasYFilas {
public static void main(String[] args) {
int[][] matriz = new int[50][100]; // Matriz de números enteros que supondremos llena.
                                                // 50 filas y 100 columnas.

for (int i = 0; i < 100; i++)           // El primer índice recorre las columnas.
for (int j = 0; j < 50; j++){   // El segundo índice recorre las filas.
                                        // Procesamos cada elemento de la matriz.
System.out.println(matriz[j][i]);     // ¡Índices cambiados de orden!
                        }
        }
}
```

Por filas y columnas al revés

```
public class RecorrerMatrizPorFilasYColumnasAlReves {
```

```
public static void main(String[] args) {

int[][] matriz = new int[50][100]; // Matriz de números enteros que supondremos llena.

                                                    // 50 filas y 100 columnas.

for (int i = 49; i > 0; i--)                  // El primer índice recorre las filas.

for (int j = 99; j > 0; j--){      // El segundo índice recorre las columnas.

                                    // Procesamos cada elemento de la matriz.

System.out.println(matriz[i][j]);

                        }

        }

}
```

O sin cambiar los índices:

```
public class RecorrerMatrizPorFilasYColumnasAlReves2 {

public static void main(String[] args) {

int[][] matriz = new int[50][100]; // Matriz de números enteros que supondremos llena.

                                                    // 50 filas y 100 columnas.

for (int i = 0; i < 50; i++)                  // El primer índice recorre las filas.

for (int j = 0; j < 100; j++){  // El segundo índice recorre las columnas.

                                    // Procesamos cada elemento de la matriz.

System.out.println(matriz[49 - i][99 - j]);

                        }

        }

}
```