

LISTADO DE EJERCICIOS. JAVA BÁSICO (5). Arrays con métodos.

Con respecto a los arrays:

- a. Un array es un objeto, por lo que se manejan mediante referencias a objetos.
- b. Un array es un objeto contenedor que contiene un número fijo de valores de un mismo tipo. `int[] array;`
- c. La longitud de un array se fija cuando se crea. La propiedad `length` almacena la longitud del array. `int[] array = new int[10];`
- d. Los elementos de un array se identifican mediante índices. Un índice varía desde 0 hasta `length - 1`
- e. Los arrays se recorren mediante bucles. El bucle `for` es el más habitual: `for (int i = 0; i < matriz.length; i++) System.out.println(matriz[i]);`
- f. Un bucle especial para el recorrido de arrays es el `for each`. Se utiliza sólo para la lectura, no para modificación.
- g. Un vector es un array unidimensional `int[] vector = {9, 4, 7, 1};`
- h. Una matriz es un array bidimensional `int[][] array = {{4, 7, 8}, {9, 0, 1}};` Una matriz está compuesta de filas y columnas. `array[0]` es la primera fila
- i. Podemos considerar una matriz como un vector de vectores. Cada fila es un nuevo vector, cuyo tamaño no ha de coincidir. `int[][] matriz = {{1, 2}, {3, 4, 5}};`
`int[] fila0 = matriz[0];`

- 2. Con respecto a un array, indica verdadero o falso:
 - a. La longitud de un array puede modificarse después de su creación.
 - b. El primer índice siempre es el 0.
 - c. En un array bidimensional, el tamaño de cada una de sus filas siempre es el mismo.
 - d. El tipo de los elementos siempre es el mismo.
- 3. Crea una clase `DelReves` que implemente el método `mostrarDelReves(int [])`. Se le pasará como argumento un vector de enteros inicializado mediante llaves. Mostrará dicho vector del revés.
- 4. Crea una clase `CreaVector` que solicite al usuario el tamaño de un nuevo vector y que le pida los enteros necesarios para inicializarlo. Utiliza los métodos `pedirDimension()` y `pedirDatos(int[])`
- 5. Crea una clase `MayorDelVector` que cree un vector con enteros aleatorios y muestre el valor máximo. Para ello, implementa los métodos `pedirDimension()`, `inicializarVectorAleatoriamente(int[], i, max)` y `hallarMaximo(int[])`.
- 6. Crea una clase `MenorDelVector` que cree un vector con enteros aleatorios y muestre el valor mínimo. Para ello, implementa los métodos `pedirDimension()`, `inicializarVectorAleatoriamente(int[], min, max)` y `hallarMinimo(int[])`.
- 7. Crea una clase `MediaDelVector` que cree un vector con enteros aleatorios y muestre la media del vector. Para ello, implementa los métodos `pedirDimension()`, `inicializarVectorAleatoriamente(int[], min, max)` y `hallarMedia(int[])`.
- 8. Define la clase `Notas` que almacene en un array 15 notas. Las notas se inicializarán de forma aleatoria con enteros entre 0 y 10. Se visualizarán las notas resultantes.
- 9. Define la clase `ModificaVector`. Realizará las siguientes opciones:

- a. Creará un vector con la dimensión indicada por el usuario.
- b. Inicializará con valores consecutivos (1, 2, 3...).
- c. Después se solicitará al usuario un elemento del array (entre 1 y longitud) y su nuevo valor para modificarlo.
- d. Este proceso de modificación de un elemento se repetirá tantas veces como quiera el usuario (¿quiere modificar otro(s/n)?)
- e. Al acabar, se visualizarán el array ya modificado.

Vamos a crear un array con valores consecutivos. Dame la longitud del array (entero positivo):

3

YA LO HE CREADO. Ahora vamos a modificar el array.

Dame un elemento del array (entre 1 y 3):

3

Dame un valor para introducir en la posición 3 del array:

33

¿Quieres modificar otro (s/n)?

s

Dame un elemento del array (entre 1 y 3):

1

Dame un valor para introducir en la posición 1 del array:

11

¿Quieres modificar otro (s/n)?

n

a[0] = 11

a[1] = 2

a[2] = 33

10. Define la clase `AsignandoArrays` tal y como muestra la captura. Indica en un comentario lo que almacenaría la referencia `a1` tras modificarse `a2`. Añade también los siguientes comentarios en las líneas correspondientes:

- a. se referencia al mismo objeto
- b. modificación de los elementos
- c. declaración de un array
- d. se muestran los elementos del objeto array
- e. declaración, construcción e inicialización de un array

```

/**
 * Muestra el manejo de arrays
 * @author Lourdes Magarín
 * @version 1.0
 */public class AsignandoArrays {
    /**
     * Declara, construye, inicializa y muestra arrays.
     */
    public static void main (String[] args) {
        int [] a1 = { 3, 5, 7, 11, 13 } ;
        int [] a2;
        int i;

        a2 = a1;

        for(i = 0; i < a2.length; i++)
            a2 [i] ++;

        for ( i = 0; i < a1.length; i++)
            System.out.println("a1[" + i + "] = " + a1[i]);
    }
}

```

11. Define la clase MatrizDecimales que cree una matriz de decimales. Pide las dimensiones (filas y columnas). Se inicializa con valores aleatorios entre 1 y 100 y visualiza el array.
12. Define la clase MatrizDecimales2 basada en la clase anterior. Se ha de inicializar con valores aleatorios (solicita al usuario el mínimo y el máximo de los valores). Ha de localizar los valores máximo, mínimo y media de la matriz. Que muestre tanto los valores como la posición que ocupan en la matriz. Utiliza los métodos pedirFila(), pedirColumna(), pedirMínimo(), pedirMaximo(), crearMatriz(), mostrarMatriz(double[][]), hallarMedia(double[][]);
13. Define la clase SumaMatrices que sume dos matrices de enteros. El usuario introducirá por teclado las dimensiones de las matrices (m x n). Las matrices se inicializarán de forma aleatoria con enteros entre 0 y 10. Se visualizarán sumandos y suma. Completa este esqueleto:

```

import utiles.Teclado;

public class SumaMatrices {
public static void main(String[] args){
    int filas = pedirDimension("Dame las filas: ");
    int columnas = pedirDimension("Dame las columnas: ");

    int [][] sumando1= crearMatriz(filas, columnas);
    mostrar(sumando1, "MATRIZ 1: ");
    int [][] sumando2 = crearMatriz (filas, columnas);
    mostrar(sumando2, "MATRIZ 2: ");
}
}

```

```

    int [][] suma = sumar (sumando1, sumando2);
    mostrar(suma, "MATRIZ SUMA: ");
}

```

14. Define la clase TraspuestaMatriz que invierta una matriz de enteros. Se piden las dimensiones y toma valores aleatorios entre 1 y 10.

Matriz:

1	2	10	9	7	7	9
7	5	4	4	1	1	6
8	6	7	8	10	6	1
5	7	8	9	6	2	2

Resultado:

1	7	8	5
2	5	6	7
10	4	7	8
9	4	8	9
7	1	10	6
7	1	6	2
9	6	1	2

15. Define la clase MultiplicaMatrices que multiplique dos matrices de enteros. El usuario introducirá por teclado las dimensiones de las matrices (m x n, n x o). Las matrices se inicializarán de forma aleatoria con enteros entre 1 y 10. Se visualizarán factores y resultado.
16. Crea una clase CreaMatriz que solicite al usuario el tamaño de una nueva matriz y que le pida los enteros necesarios para inicializarlo. Utiliza los métodos pedirFila(), pedirColumna() y pedirDatos(int[])
17. Crea una clase MostrarMatriz que solicite al usuario el tamaño de una nueva matriz y que le pida los enteros necesarios para inicializarla. Utiliza los métodos pedirFila(), pedirColumna(), pedirDatos(int[]) y mostrarMatriz();
18. Crea una clase MínimoMatriz que solicite al usuario el tamaño de una nueva matriz y que la inicialice con valores aleatorios (solicita al usuario el mínimo y el máximo de los valores). Hallará el valor mínimo de una matriz, dando exactamente su posición fila-columna. Utiliza los métodos pedirFila(), pedirColumna(), pedirMínimo(), pedirMaximo, mostrarMatriz(), mostrarMínimo() ;
19. Crea la clase NotasAlumnos. Declara y crea un tipo de variable que sea capaz de contener la nota que han obtenido 4 alumnos en los diferentes módulos en los que están matriculados. Sabemos que el primer alumno está matriculado en 4 módulos; el segundo en 7, el tercero en 3 módulos y el cuarto en dos módulos. Genera aleatoriamente las notas de los alumnos y muéstralos. Utiliza métodos de la forma más eficiente y clara.
20. Crea la clase AlumnosMatriculados. Queremos guardar en la memoria del ordenador el número de alumnos matriculados en cada grupo del ciclo de "Desarrollo de Aplicaciones Web" (por módulos). Existen dos grupos y se imparten seis módulos distintos en cada grupo. Utiliza métodos en la medida de lo posible. Sabemos que:
- En el primer grupo hay:
 - 30 alumnos matriculados en Sistemas informáticos
 - 27 alumnos matriculados en Bases de datos

- iii. 25 alumnos matriculados en Programación
 - iv. 21 alumnos matriculados en Lenguaje de marcas
 - v. 19 alumnos matriculados en Entornos de desarrollo
 - vi. 16 alumnos matriculados en FOL
 - b. En el segundo grupo hay:
 - i. 15 alumnos matriculados en Desarrollo Web en entorno cliente
 - ii. 14 alumnos matriculados en Desarrollo Web en entorno servidor
 - iii. 15 alumnos matriculados en Despliegue de aplicaciones Web
 - iv. 15 alumnos matriculados en Diseño de interfaces Web
 - v. 15 alumnos matriculados en Desarrollo Web en entorno cliente
21. Define la clase Secuencia que busque la secuencia en orden creciente (\leq) más larga dentro de un vector de enteros inicializado aleatoriamente (`Math.random()`). Se mostrará tanto la posición de la primera componente de la secuencia como el tamaño de la misma.
22. Define la clase SieteYMedia que permita al usuario jugar a las siete y media. Para ello, has de almacenar las 40 cartas en un array bidimensional. Utiliza el método `Math.random()` para seleccionar una de ellas. Recuerda que no pueden seleccionarse dos veces. El jugador decide si seguir arriesgando o no mediante una pregunta.