

Actividad6_1

Índice

Ejercicio6_1_1.....	1
Ejercicio6_1_2.....	1
Ejercicio6_1_3.....	2
Ejercicio6_1_4.....	2
Ejercicio6_1_5.....	2
Ejercicio6_1_6.....	2
Ejercicio6_1_7.....	2
Ejercicio6_1_8.....	3
Ejercicio6_1_9.....	3
Ejercicio6_1_10.....	3
Ejercicio6_1_11.....	4
Ejercicio6_1_12. Matriz irregular triangular.....	4
Ejercicio6_1_13.....	4

Ejercicio6_1_1.

Define tres arrays de 20 números enteros cada una:

- ❖ numero
- ❖ cuadrado
- ❖ cubo.

Carga el **array numero** con valores aleatorios entre 0 y 100.

- ❖ En el **array cuadrado** se deben almacenar los cuadrados de los valores que hay en el array numero.
- ❖ En el **array cubo** se deben almacenar los cubos de los valores que hay en numero.
- ❖ A continuación, muestra el **contenido de los tres arrays** dispuesto en tres columnas.

```

public static void inicializar (int[] numero, int[] cuadrado, int[] cubo )
{
    System.out.printf("%6s | %6s | %6s %n", "n", "n^2", "n^3");
    System.out.println("-----");
    for(int i = 0; i < numero.length; i++)
    {
        numero[i] = (int) (Math.random()* 101);
        cuadrado[i] = numero[i] * numero[i];
        cubo[i] = numero[i] * numero[i] * numero[i];
        System.out.printf("%6d | %6d | %6d %n", numero[i], cuadrado[i], cubo[i]);
    }
}

public static void main(String[] args)
{
    int[] numero = new int[5];
    int[] cuadrado = new int[5];
    int[] cubo = new int[5];

    inicializar(numero, cuadrado, cubo);
}
}

```

```

n |    n^2 |    n^3
-----
82 |   6724 | 551368
14 |    196 |   2744
72 |   5184 | 373248
43 |   1849 | 79507
20 |    400 |   8000
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio6_1_2.

Realiza un programa que pida **la temperatura media** que ha hecho en cada mes de un determinado año y que muestre a continuación un **diagrama de barras horizontales con esos datos**. Las barras del diagrama se pueden dibujar a base de asteriscos o cualquier otro carácter.

```

public static void grafica(int[] temperatura, String[] año)
{
    for(int i = 0; i < temperatura.length; i++)
    {
        System.out.printf("%10s |", año[i]);
        for(int j = 0; j <= temperatura[i]; j++)
        {
            System.out.print("=");
        }
        System.out.printf(" %d°C", temperatura[i]);
        System.out.println("");
    }
}

public static void main(String[] args)
{
    int[] temperatura = {6, 10, 15, 20, 24, 30, 35, 36, 31, 21, 12, 7};
    String[] año = {"enero", "febrero", "marzo", "abril", "mayo", "junio", "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"};

    grafica(temperatura, año);
}

```

```

    enero |===== 6°C
    febrero |===== 10°C
    marzo |===== 15°C
    abril |===== 20°C
    mayo |===== 24°C
    junio |===== 30°C
    julio |===== 35°C
    agosto |===== 36°C
    septiembre |===== 31°C
    octubre |===== 21°C
    noviembre |===== 12°C
    diciembre |===== 7°C
    BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio6_1_3.

Realiza un programa que pida 8 números enteros y que luego muestre esos números junto con la **palabra "par" o "impar"** según proceda.

```
public class Ejercicio6_1_3
{
    public static void mostrar(int[] numero)
    {
        for(int i = 0; i < numero.length; i++)
        {
            if(numero[i]%2 == 0)
                System.out.printf("%d par %n", numero[i]);
            else
                System.out.printf("%d impar %n", numero[i]);
        }
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        int tamaño = 8;
        int[] numero = new int[tamaño];

        System.out.println("Introduzca 8 números enteros, pulse INTRO después de cada número");
        for(int i = 0; i < tamaño; i++)
        {
            numero[i] = sc.nextInt();
        }
        System.out.println("");
        mostrar(numero);
    }
}
```

```
Introduzca 8 números enteros, pulse INTRO después de cada número
79
44
33
21
50
45
80
303

79 impar
44 par
33 impar
21 impar
50 par
45 impar
80 par
303 impar
BUILD SUCCESSFUL (total time: 17 seconds)
```

Ejercicio6_1_4.

Escribe un programa que pida 10 números por teclado y que luego muestre los números introducidos junto con las **palabras "máximo" y "mínimo"** al lado del máximo y del mínimo respectivamente.

```
public static void mostrar(int[] numero)
{
    int maximo = 0, minimo = 0;
    for(int i = 0; i < numero.length; i++)
    {
        if(maximo < numero[i])
            maximo = numero[i];
        else if(minimo > numero[i])
            minimo = numero[i];
    }
    for(int i = 0; i < numero.length; i++)
    {
        if(maximo == numero[i])
            System.out.println("" + maximo + " máximo");
        else if (minimo == numero[i])
            System.out.println("" + minimo + " mínimo");
        else
            System.out.println("" + numero[i]);
    }
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);

    int tamaño = 10;
    int[] numero = new int[tamaño];

    System.out.println("Vaya introduciendo números enteros y pulsando INTIO;");
    for(int i = 0; i < tamaño; i++)
    {
        numero[i] = sc.nextInt();
    }
    System.out.println("");
    mostrar(numero);
}
```

```
Vaya introduciendo números enteros y pulsando INTO;
88
62
13
95
22
3000
2000
-10
-20
19

88
62
13
95
22
3000 máximo
2000
-10
-20 mínimo
19
```

Ejercicio6_1_5.

Implementa un programa que calcule la **denominación ordinal** de los reyes de una secuencia histórica, según lo siguiente:

- ❖ El programa **solicitará la cantidad de reyes** que se van a introducir y, a continuación, recibirá los nombres de los reyes.
- ❖ Presentará por pantalla **dichos nombres**, pero colocándoles el ordinal correspondiente.
- ❖ Así, por ejemplo, si hay **dos Felipes** en los nombres de los reyes, el primero debería aparecer como Felipe 1º y el segundo como Felipe 2º.

```
public static int repetido(String[] nombre, int i)
{
    int repetido = 1;

    for (int j = 0; j < i; j++)
    {
        if(nombre[i].equals(nombre[j]))
        {
            repetido++;
        }
    }
    return repetido;
}
```

```
public static void mostrar(String[] nombre)
{
    for(int i = 0; i < nombre.length; i++)
    {
        System.out.printf("%s %d° %n", nombre[i], repetido(nombre, i));
    }
}
```

```

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);

    int tamaño;
    System.out.print("Introduzca el número total de nombres de reyes: ");
    tamaño = sc.nextInt();

    String[] nombre = new String[tamaño];

    System.out.println("Vaya introduciendo los nombres de los reyes y pulsando INTRO.");
    for (int i = 0; i < tamaño; i++)
    {
        nombre[i] = sc.next();
    }
    System.out.printf("\nLos reyes introducidos sono:");
    mostrar(nombre);
}

```

```

Introduzca el número total de nombres de reyes: 7
Vaya introduciendo los nombres de los reyes y pulsando INTRO.
Felipe
Carlos
Carlos
Fernando
Carlos
Carlos
Felipe

    Los reyes introducidos sono:Felipe 1º
Carlos 1º
Carlos 2º
Fernando 1º
Carlos 3º
Carlos 4º
Felipe 2º
BUILD SUCCESSFUL (total time: 18 seconds)

```

Ejercicio6_1_6.

Crea y carga una matriz de **4 filas por 4 columnas**, para que se muestre de la siguiente manera:

Debes utilizar los **siguientes métodos**:

- public static char[][] crearMatriz(int n), para crear la matriz de tamaño n.
- public static void imprimirMatriz(char[][] m), para mostrar el contenido.


```
public static char[][] crearMatriz(int n)
{
    return new char[n][n];
}

public static void imprimirMatriz(char[][] m)
{
    int posicionX = 0;

    for (int i = 0; i < m.length; i++)
    {
        for(int j = 0; j < m.length; j++)
        {
            if(posicionX == j )
                m[i][j] = 'X';
            else
                m[i][j] = '-';
            System.out.printf("%s ", m[i][j]);
        }
        System.out.println("");
        posicionX++;
    }
}

public static void main(String[] args)
{
    int n = 4;

    char[][] m = crearMatriz(n);
    imprimirMatriz(m);
}
```

X	-	-	-
-	X	-	-
-	-	X	-
-	-	-	X

Ejercicio6_1_7.

Escribe un programa que a partir de un array de 4 filas y 5 columnas cuyos datos los indicamos directamente en el código, muestre **las sumas parciales** de filas y **columnas igual** que si de una hoja de cálculo se tratara. La suma total debe aparecer en la esquina inferior derecha.

```

public static void rellenar(int[][] lista)
{
    for (int i = 0; i < lista.length; i++)
    {
        for(int j = 0; j < lista[i].length; j++)
        {
            lista[i][j] = (int) (Math.random()*201);
        }
    }
}

public static void mostrar(int[][] lista, int tamañoX, int tamañoY)
{
    int sumaFila = 0, sumaColumna = 0, sumaTotal = 0;

    for (int i = 0; i < lista.length; i++)
    {
        for(int j = 0; j < lista[i].length; j++)
        {
            System.out.printf("%7d", lista[i][j]);
            sumaFila += lista[i][j];
        }
        System.out.printf(" | %5d %n", sumaFila);
        sumaTotal += sumaFila;
        sumaFila = 0;
    }

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

    int[][] listaInvertida = invertirMatriz(lista, tamañoX, tamañoY);

    for (int i = 0; i < listaInvertida.length; i++)
    {
        for(int j = 0; j < listaInvertida[i].length; j++)
        {
            sumaColumna += listaInvertida[i][j];
        }
        System.out.printf(" %5d", sumaColumna);
        sumaTotal += sumaColumna;
        sumaColumna = 0;
    }
    System.out.printf(" | %5d %n", sumaTotal);
}

public static int[][] invertirMatriz(int[][] lista, int tamañoX, int tamañoY)
{
    int[][] listaInvertida = new int[tamañoY][tamañoX];
    for (int i = 0; i < lista.length; i++)
    {
        for(int j = 0; j < lista[i].length; j++)
        {
            listaInvertida[j][i] = lista[i][j];
        }
    }
    return listaInvertida;
}

public static void main(String[] args)
{
    int tamañoX = 4, tamañoY = 5;
    int[][] lista = new int[tamañoX][tamañoY];

    rellenar(lista);
    mostrar(lista, tamañoX, tamañoY);
}

```

140	173	187	137	101		738
54	200	101	159	40		554
25	72	105	147	103		452
158	19	19	135	61		392

377	464	412	578	305		4272
BUILD SUCCESSFUL (total time: 0 seconds)						

Ejercicio6_1_8.

Realiza un programa que realice lo siguiente:

- ◆ Rellene un array de **6 filas por 10 columnas** con números enteros positivos comprendidos entre 0 y 1000 (ambos incluidos).
- ◆ El programa deberá **dar la posición tanto del máximo como del mínimo**.

```

public static void rellenar(int[][] lista)
{
    for (int i = 0; i < lista.length; i++)
    {
        for(int j = 0; j < lista[i].length; j++)
        {
            lista[i][j] = (int) (Math.random()*1001);
        }
    }
}

public static void mostrar(int[][] lista)
{
    int maximo = 0, minimo = lista[0][0], maximoFila = 0, maximoColumna = 0, minimoFila = 0, minimoColumna = 0;

    for(int i = 0; i < lista.length; i++)
    {
        for(int j = 0; j < lista[i].length; j++)
        {
            System.out.printf("%6d", lista[i][j]);
            if(maximo < lista[i][j])
            {
                maximo = lista[i][j];
                maximoFila = i;
                maximoColumna = j;
            }
            else if(minimo > lista[i][j])
            {
                minimo = lista[i][j];
                minimoFila = i;
                minimoColumna = j;
            }
        }
        System.out.println("");
    }

    System.out.println("");
    System.out.printf("El máximo es %d y está en la fila %d, columna %d %n", maximo, maximoFila, maximoColumna);
    System.out.printf("El mínimo es %d y está en la fila %d, columna %d %n", minimo, minimoFila, minimoColumna);
}

public static void main(String[] args)
{
    int[][] lista = new int[6][10];

    rellenar(lista);
    mostrar(lista);
}

```

```
run:
```

```
506 891 291 146 581 609 912 293 106 701
661 433 28 550 688 674 83 763 945 234
340 153 358 358 428 9 524 631 594 517
247 521 624 2 384 644 382 448 954 389
158 517 157 312 289 240 13 890 528 340
592 32 492 123 396 69 51 793 345 714
```

```
El máximo es 954 y está en la fila 3, columna 8
```

```
El mínimo es 2 y está en la fila 3, columna 3
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio6_1_9.

Escribe un programa en Java que gestione las notas de varios alumnos usando **arrays paralelos**.

El programa debe:

1. Pedir por teclado el **número de alumnos** que se van a registrar (por ejemplo, entre 1 y 30).
2. Declarar **dos arrays** del mismo tamaño:

String[] nombres → para los nombres de los alumnos.

double[] notas → para las notas correspondientes.

Rellenar ambos arrays pidiendo, para cada alumno, su **nombre** y su **nota**, guardando ambos datos en la misma posición de los arrays.

A continuación, pedir por teclado el **nombre de un alumno** a buscar.

Buscar ese nombre en el **array nombres** (búsqueda secuencial):

Si se encuentra, mostrar por pantalla su nota correspondiente del array notas.

Si no se encuentra, hay que **indicarlo**.

```
package ejercicio6_1_9;

import java.util.Scanner;
import java.util.Arrays;
import java.util.List;

public class Ejercicio6_1_9
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        int numeroAlumnos;
        String buscar;

        System.out.print("Número de alumnos: ");
        numeroAlumnos = sc.nextInt();

        String[] clase = new String[numeroAlumnos];
        double[] nota = new double[numeroAlumnos];

        for(int i = 1; i <= numeroAlumnos; i++)
        {
            System.out.print("Nombre del alumno " + i + ": ");
            clase[i - 1] = sc.next();
            System.out.print("Nota de " + clase[i - 1] + ": ");
            nota[i - 1] = sc.nextDouble();
        }

        System.out.println("");
        System.out.print("Introduce el nombre del alumno a buscar: ");
        buscar = sc.next();

        if(!Arrays.asList(clase).contains(buscar))
            System.out.println("Alumno no registrado.");
        else
            System.out.println("La nota de luis es: " + nota[Arrays.asList(clase).indexOf(buscar)]);
    }
}
```

```
Número de alumnos: 3
Nombre del alumno 1: Ana
Nota de Ana: 7,5
Nombre del alumno 2: Luis
Nota de Luis: 8
Nombre del alumno 3: Marta
Nota de Marta: 6,25

Introduce el nombre del alumno a buscar: Luis
La nota de luis es: 8.0
BUILD SUCCESSFUL (total time: 21 seconds)
```

Ejercicio6_1_10.

Crea un programa que una dos arrays, según lo siguiente:

- ❖ En el **main** declara dos arrays de enteros ya inicializados, por ejemplo:
 - ❖ `int[] a = {1, 2, 3};`
 - ❖ `int[] b = {4, 5};`
- ❖ Crea una función:
 - ❖ `public static int[] concatenar(int[] a, int[] b)` que devuelva un nuevo array con todos los elementos de a seguidos de todos los de b.
- ❖ Muestra el resultado con **Arrays.toString**.


```

package ejercicio6_1_10;

import java.util.Arrays;

public class Ejercicio6_1_10
{
    public static int[] concatenar(int[] a, int[] b)
    {
        int[] concatena0 = new int[a.length + b.length];
        for(int i = 0; i < concatena0.length; i++)
        {
            if(a.length > i)
                concatena0[i] = a[i];
            else
                concatena0[i] = b[i - a.length];
        }

        Arrays.sort(concatena0);
        return concatena0;
    }

    public static void main(String[] args)
    {
        int[] a = {1, 2, 3};
        int[] b = {4, 5};

        System.out.println("Array a: " + Arrays.toString(a));
        System.out.println("Array b: " + Arrays.toString(b));

        System.out.println("Array concatenado: " + Arrays.toString(concatenar(a, b)));
    }
}

```

```

Array a: [1, 2, 3]
Array b: [4, 5]
Array concatenado: [1, 2, 3, 4, 5]
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio6_1_11.

Crea un programa que lea los **argumentos (palabras)** en las propiedades del proyecto y realice lo siguiente:

- ❖ Usando **bucle for-each**:
Muestra **cada palabra** en una línea.

Calcula el **número total de vocales** (a, e, i, o, u minúsculas o mayúsculas) que aparecen en todas las palabras y las muestre al lado de la palabra.

Muestra el total de **vocales encontradas**.

```

public static void main(String[] args)
{
    int vocales = 0, vocalesTotal = 0;

    System.out.print("Argumentos recibidos:");
    for (String palabra : args)
    {
        System.out.print(" " + palabra);
    }

    System.out.printf("\n\n");
    for (String palabra : args)
    {
        palabra = palabra.toLowerCase();
        for(int i = 0; i < palabra.length(); i++)
        {
            switch(palabra.charAt(i))
            {
                case 'a', 'e', 'i', 'o', 'u' -> vocales++;
            }
        }
        vocalesTotal += vocales;
        System.out.printf("%s -> vocales: %d \n", palabra, vocales);
        vocales = 0;
    }
    System.out.printf("\nTotal de vocales: %d \n", vocalesTotal);
}

```

```

Argumentos recibidos: Hola DAM programacion

hola -> vocales: 2
dam -> vocales: 1
programacion -> vocales: 5

Total de vocales: 8
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio6_1_12. Matriz irregular triangular

Crea un **programa** que:

Pida por teclado un número n (entre 1 y 10).

Cree una matriz irregular de enteros de tamaño n filas:

```
int[][] tri = new int[n][];
```

donde la fila i tenga i + 1 columnas.

Rellene el triángulo con números consecutivos empezando en 1, de izquierda a derecha y de arriba a abajo.

Muestre por pantalla el triángulo con **un espacio** entre números.

```

import java.util.Scanner;

public class Ejercicio6_1_12
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        int n, contador = 1;

        System.out.print("Introduce n (1-10): ");
        n = sc.nextInt();

        int[][] tri = new int[n][];

        System.out.println("");
        for(int i = 0; i < tri.length; i++)
        {
            tri[i] = new int[i + 1];
            for(int j = 0; j < tri[i].length; j++)
            {
                tri[i][j] = contador;
                System.out.print(tri[i][j] + " ");
                contador++;
            }
            System.out.println("");
        }
    }
}

```

```

Introduce n (1-10): 4

1
2 3
4 5 6
7 8 9 10
BUILD SUCCESSFUL (total time: 1 second)

```

Ejercicio6_1_13.

Investiga en qué consiste el algoritmo de **ordenación de la burbuja** e impleméntalo en java para ordenar un **array de 10 enteros** elegidos al azar.

```

import java.util.Arrays;

public class Ejercicio6_1_13
{
    public static int[] burbuja(int[] lista)
    {
        int buffer;

        for(int i = 0; i < lista.length; i++)
        {
            for(int j = 0; j < lista.length - 1; j++)
            {
                if (lista[j] > lista[j + 1])
                {
                    buffer = lista[j + 1];
                    lista[j + 1] = lista[j];
                    lista[j] = buffer;
                    i = 0;
                }
            }
        }

        return lista;
    }

    public static void main(String[] args)
    {
        int[] lista = {2, 3, 5, 4, 1, 6, 9, 7, 8, 10};

        System.out.println("Desordenado: " + Arrays.toString(lista));
        System.out.println("Ordenado: " + Arrays.toString(burbuja(lista)));
    }
}

```

```

Desordenado: [2, 3, 5, 4, 1, 6, 9, 7, 8, 10]
Ordenado: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
BUILD SUCCESSFUL (total time: 0 seconds)

```