Aaplicacion5.11 - Apache NetBeans IDE 23

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

```java
...4 lines
package aaplicacion5.pkg11;

import java.util.Arrays;

/**...4 lines */
public class Aaplicacion511 {

    /**...3 lines */
    /*
    Realiza la función: int [] buscarTodos (int t[], int clave), que crea y devuelve una
    tabla con todos los índices de los elementos donde se encuentra la clave de búsqueda. En
    el caso de que clave no se encuentre en la tabla t, la función devolverá una tabla vacía.
    */
    public static void main(String[] args) {
        int[] t = {6, 5, 1, 13, 5, 3, 23, 18, 11, 5, 4};
        int clave = 5;
        int[] encontrados = buscarTodos(t, clave);
        System.out.println("La tabla con indices de la clave buscada es: " + Arrays.toString(encontrados));
    }

    static int [] buscarTodos (int t[], int clave){
        int contador = 0;
        for (int i : t){//bucle para contar cuantas veces aparece clave en la busqueda
            if (i == clave){
                contador++;
            }
        }
        //creamos la tabla indice
        int indice [] = new int[contador];
        int indiceActual = 0;
        for (int i = 0; i < t.length; i++){//bucle para recorrer el array t buscando si coincide con clave
            if (t[i] == clave) {
                indice [indiceActual] = i;//guardamos en la tabla indice el indice del elemento que coincide con clave
                indiceActual++;//incrementar
            }
        }
        return indice;
    }

}
```

aaplicacion5.pkg11.Aaplicacion511 > main

Output - Aaplicacion5.11 (run)

```
run:
La tabla con indices de la clave buscada es: [1, 4, 9]
BUILD SUCCESSFUL (total time: 0 seconds)
```

27:69 INS Unix (LF)

Aaplicacion5.12 - Apache NetBeans IDE 23

Consejos
Aprende a echar un vistazo rápidamente.

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

Source | History

```java
...4 lines
package aaplicacion5.pkg12;

import java.util.Arrays;

/**...4 lines */
public class Aaplicacion512 {

    /**...3 lines */
    /*
    Escribe la función void desordenar (int t []), que cambia de forma aleatoria los
    elementos contenidos en la tabla t. Si la tabla estuviera ordenada, dejaria de estarlo.
    */
    public static void main(String[] args) {
        int[] t = {1, 3, 7, 10, 13, 17, 25, 33, 54, 59, 68};//declaración del array
        desordenar(t);//llamada a la función desordenar, le pasamos como parametro la tabla t
        System.out.println(Arrays.toString(t));//imprimir en pantalla el array

    }
    static void desordenar (int t []){
        for (int j = 0; j <= t.length-1; j++){//bucle para recorrer el array
            int a = (int) (Math.random() * (t.length));//generamos un número aleatorio entre 0 y la cantidad de elementos del array
                int aux = t[j];//guardamos en una variable auxiliar el elemento t[j]
                t[j] = t[a];//guardamos t[a] en t[j]
                t[a] = aux; //guardamos en t[a] el elemento que tenemos en la variable auxiliar
        }
    }

}
```

aaplicacion5.pkg12.Aaplicacion512 > main

Output - Aaplicacion5.12 (run)

```
run:
[54, 17, 59, 13, 10, 7, 1, 3, 33, 68, 25]
BUILD SUCCESSFUL (total time: 0 seconds)
```

24:94 INS Unix (LF)

Aaplicacion5.13 - Apache NetBeans IDE 23

Consejos
Aprende a echar un vistazo rápidamente.

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

```java
...4 lines
package aaplicacion5.pkg13;

import java.util.Arrays;

/**...4 lines */
public class Aaplicacion513 {

    /**
     * @param args the command line arguments
     */
    /*
    Modifica la Actividad de aplicación 5.12 para que la función no modifique la tabla que se
    pasa como parámetro y, en su lugar, cree y devuelva una copia de la tabla donde se han
    desordenado los valores d e los elementos.
    */
    public static void main(String[] args) {
        int[] t = {1, 3, 7, 10, 13, 17, 25, 33, 54, 59, 68};//declaramos el array
        int copia[] = desordenar(t);//creamos una nueva tabla y la rellenamos con los elementos que devuelve la función desordenar
        System.out.print("La tabla original:        ");
        System.out.println(Arrays.toString(t));//imprimir en pantalla la tabla t
        System.out.print("La copia desordenada:     ");
        System.out.println(Arrays.toString(copia));//imprimior en pantalla la copia de t
    }
    static int[] desordenar (int t []){
        int copia[] = Arrays.copyOf(t, t.length);//hacemos una copia del array t
            for (int j = 0; j <= copia.length-1; j++){//bucle para recorrer el array
            int a = (int) (Math.random() * (copia.length));//guardamos en a un número aleatorio entre 0 y los elementos de copia -1
                int aux = copia[j];//guardamos en una variable auxiliar el elemento copia[j]
                copia[j] = copia[a];//guardamos en copia[j] el elemenmto de copia[a]
                copia[a] = aux;//guardamos en copia[a] el elemento que tenemos en la variable auxiliar
        }
        return copia;// devolvemos el array copia
    }
}
```

aaplicacion5.pkg13.Aaplicacion513 > main

Output - Aaplicacion5.13 (run)

```
run:
La tabla original:        [1, 3, 7, 10, 13, 17, 25, 33, 54, 59, 68]
La copia desordenada:     [13, 68, 3, 7, 59, 10, 33, 25, 54, 17, 1]
BUILD SUCCESSFUL (total time: 0 seconds)
```

26:53 INS Unix (LF)

Aaplicacion5.14 - Apache NetBeans IDE 23

Consejos
Aprende a echar un vistazo rápidamente.

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

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

/**...4 lines */
public class Aaplicacion514 {

    /**...3 lines */
    /*
    El ayuntamiento de tu localidad te ha encargado una aplicación que ayude a realizar
    encuestas estadísticas para conocer el nivel adquisitivo de los habitantes del municipio.
    Para ello, tendrás que preguntar el sueldo a cada persona encuestada. A priori, no conoces el
    número de encuestados. Para finalizar la entrada de datos, introduce un sueldo con valor -1.
    Una vez terminada la entrada de datos, muestra la siguiente información:
    • Todos los sueldos introducidos ordenados de forma decreciente.
    • El sueldo máximo y minimo.
    • La media de los sueldos.
    */
    public static void main(String[] args) {
        int i = 0;//creamos una variable indice
        Scanner sc = new Scanner(System.in);//introducir datos
        double sueldos[] = new double[0];//creamos el array sueldos con 0 elementos
        System.out.println("Introduzca el sueldo de las personas a analizar (Para finalizar introduzca -1): ");
        double sueldo = sc.nextDouble();
        while (sueldo != -1) {//bucle que guarda los sueldos hasta que se introduzca -1
            sueldos = Arrays.copyOf(sueldos, i + 1);//hacemos una copoia de el array sueldos agregando un elemento
            sueldos[i] = sueldo;//en la posicion i de sueldos guardamos el sueldo introducido
            sueldo = sc.nextDouble();//capturamos el siguiente número introducido
            i++;//incrementamos el indice
        }//imprimir en pantalla los datos que se piden
        System.out.println("Sueldos: " + Arrays.toString(ordenDecreciente(sueldos)));
        System.out.println("Sueldo maximo : " + sueldos[0]);
        System.out.println("Sueldo minimo : " + sueldos[sueldos.length-1]);
        System.out.println("Sueldo medio : " + media(sueldos));
    }
    static double[] ordenDecreciente(double[] t){
        Arrays.sort(t);//ordenar el array sueldos de forma creciente
        for (int i = 0; i < t.length/2; i++){//recorremos el bucle hasta la mitad
            double temp = t[i];//en una variable temporal guardamos t[i]
            t[i] = t[t.length-i-1];//en t[i] guardamos t[elementos-i-1]
            t[t.length-i-1]= temp;//en t[elementos-1-i] guardamos lo que tenemos en temp
        }
        return t;//devolvemos t
    }

    static double media(double t[]){//función que calcula la media de los valores del array
        double med = 0;
        for (int i = 0; i< t.length; i++){//recorremos el array
            med = med + t[i];//sumamos el valor de la posición actual de el array a media
        }
        return med/t.length;//devolver med dividido la cantidad de elementos del array
    }
```

aaplicacion5.pkg14.Aaplicacion514 > main

Output - Aaplicacion5.14 (run)

```
run:
Introduzca el sueldo de las personas a analizar (Para finalizar introduzca -1):
1400
1800
2000
3000
1100
1200
-1
Sueldos: [3000.0, 2000.0, 1800.0, 1400.0, 1200.0, 1100.0]
Sueldo maximo : 3000.0
Sueldo minimo : 1100.0
Sueldo medio : 1750.0
BUILD SUCCESSFUL (total time: 22 seconds)
```

40:55 INS

Aaplicacion5.15 - Apache NetBeans IDE 23

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

```
..4 lines
ackage aaplicacion5.pkg15;

mport java.util.Scanner;

**...4 lines */
ublic class Aaplicacion515 {

    /**...3 lines */
    /*
    Debes desarrollar una aplicación que ayude a gestionar las notas de un centro educativo.
    Los alumnos se organizan en grupos compuestos por 5 personas. Leer las notas (núme-
    ros enteros) del primer, segundo y tercer trimestre de un grupo. Debes mostrar al final la
    nota media del grupo en cada trimestre y la media del alumno que se encuentra en una
    posición dada (que el usuario introduce por teclado).
    */
    public static void main(String[] args) {
        int alumno;
        int notas[][] = new int[3][5];//matriz que guarda los trimestres y los alumnos
        Scanner sc = new Scanner(System.in);
        for (int j = 0; j < 3; j++){//bucle que recorre los trimestres
            System.out.println("Introduzca las notas del trimestre " + (j+1) + ": ");
            for(int i = 0; i < 5; i++){//bucle que recorre los alumnos
                System.out.println("Alumno: " + (i + 1));
                notas[j][i] = sc.nextInt();//guardamos en notas[trimestre][alumno] la nota introducida
                while (notas[j][i] < 0 || notas[j][i]>10){//mientras la nota sea mayor que 10 o menor que 0 devuelve error y puede vover a introducir el valor
                    System.out.println("La nota introducida esta fuera de rango, vuelva a introducir un valor de 0 a 10");
                    notas[j][i] = sc.nextInt();//guardamos el valor en notas[trimestre][alumno]
                }
            }
            System.out.println();
        }
        System.out.println("La media del grupo es: " + mediaGrupo(notas));//imprimir y llamar a la función mediaGrupo pasandole la matriz notas
        System.out.println("Introduzca el número de alumno: ");//se pide el númeor de alumno
        alumno = sc.nextInt();//guardamos en alumno el valor introducido
        while (alumno < 1 || alumno > 5){
        //mientras el valor introducido en alumno sea menor que 1 o mayor que 5, devolvemos error y se pude volver a introducir el valor
            System.out.println("No existe el alumno requerido, vuelva a introducir un valor de 1 a 5");
            alumno = sc.nextInt();
        }
        System.out.println("La media del alumno es: " + mediaAlumno(notas, alumno - 1));//imprimir y llamar a la función mediaAlumno pasando la matriz alumno y alumno-1
    }
    static int mediaAlumno(int notas[][], int alumno){
        int media = 0;
        for (int i = 0; i < 3; i++){//recorremos los trimestres
            media += notas[i][alumno];//sumamos las notas de alumno de cada trimestre
        }
        return media/3;//dividr media entre los trimestres
    }
    static int mediaGrupo(int notas[][]){
        int media = 0;
        for (int i = 0; i < 5; i++){//recorre la matriz
            for (int j = 0; i < 3; i++){
                media  += notas[j][i];//sumar todas las notas de la matriz
            }
        }
        return (media/(notas.length));//dividir media por la cantidad de elementos de la matriz
    }
```

aaplicacion5.pkg15.Aaplicacion515 > mediaGrupo

aaplicacion5.pkg15.Aaplicacion515 > mediaGrupo >

Output - Aaplicacion5.15 (run)

```
run:
Introduzca las notas del trimestre 1:
Alumno: 1
8
Alumno: 2
6
Alumno: 3
4
Alumno: 4
9
Alumno: 5
10

Introduzca las notas del trimestre 2:
Alumno: 1
5
Alumno: 2
7
Alumno: 3
9
Alumno: 4
8
Alumno: 5
7

Introduzca las notas del trimestre 3:
Alumno: 1
11
La nota introducida esta fuera de rango, vuelva a introducir un valor de 0 a 10
5
Alumno: 2
8
Alumno: 3
7
Alumno: 4
6
Alumno: 5
4

La media del grupo es: 6
Introduzca el número de alumno:
5
La media del alumno es: 7
BUILD SUCCESSFUL (total time: 1 minute 38 seconds)
```

65:96 INS

Aaplicacion5.18 - Apache NetBeans IDE 23

<default config> 306,1/453,0MB

Consejos
Aprende a echar un vistazo rápidamente.

Projects

- Aaplicacion2.11
- Aaplicacion2.13
- Aaplicacion2.14
- Aaplicacion2.15
- Aaplicacion2.16
- Aaplicacion2.17
- Aaplicacion2.18
- Aaplicacion3.13
- Aaplicacion3.14
- Aaplicacion3.16
- Aaplicacion3.17
- Aaplicacion3.19
- Aaplicacion3.20
- Aaplicacion4.11
- Aaplicacion4.12
- Aaplicacion4.13
- Aaplicacion4.14
- Aaplicacion4.15
- Aaplicacion4.16
- Aaplicacion4.17
- Aaplicacion4.18
- Aaplicacion4.19
- Aaplicacion5.11
- Aaplicacion5.12
- Aaplicacion5.13
- Aaplicacion5.14
- Aaplicacion5.15
- Aaplicacion5.18
- Aaplicacion6.11
- Aaplicacion6.12
- Aaplicacion6.13
- Apropuestas2.2
- Apropuestas2.3
- Apropuestas2.4
- Apropuestas2.5

Aaplicacion511.java | Aaplicacion512.java | Aaplicacion513.java | Aaplicacion514.java | Aaplicacion515.java | Aaplicacion518.java

Source History

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

/**...4 lines */
public class Aaplicacion518 {

    /**...3 lines */
    /*
    Escribe un programa que solicite los elementos de una matriz de tamaño 4 x 4. La aplica-
    ción debe decidir si la matriz introducida corresponde a una matriz mágica, que es aquella
    donde la suma de los elementos de cualquier fila o de cualquier columna valen lo mismo.
    */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int matriz[][] = new int [4][4];//creamos la matriz de 4x4
        for(int i = 0; i < 4; i++){//recorremos la matriz para introducir cada valor
            for (int j = 0; j < 4; j++){
                System.out.println("Introducir valor para fila:  " + (i+1) + " Columna: "+ (j+1));
                matriz [i][j] = sc.nextInt();
            }
        }
        System.out.println(Arrays.deepToString(matriz));//imprimir la matriz
        System.out.println("La matriz es magica: " + esMagica(matriz));//llamada a la funcion esMagica

    }
    static boolean esMagica(int matriz[][]){//funcion que decide si la matriz es magica
        int sumaFila0 = 0;//variable que almacena la suma de los elementos de la fila 0
        int sumaFila, sumaColumna;
        boolean magica = true;
        for (int k = 0; k < 4; k++){//recorremos la matriz
            sumaFila0 += matriz[0][k];//almacenamos la suma de los elementos de la fila 0
        }
        for ( int l = 1; l < 4; l++){//recorremos la matriz a partir de la fila 1
            sumaFila = 0;
            for ( int m = 0; m < 4; m++){
                sumaFila += matriz[l][m];//almacenamos la suma de la fila l
            }
            if (sumaFila0 != sumaFila){
                //si la suma de elementos de la fila 0 es distinta a las de las demas filas no es magica
                magica = false;
            }
        }
        for ( int m = 0; m < 4; m++){//recorremos la matriz por columnas
            sumaColumna = 0;
            for ( int l = 0; l < 4; l++){
                sumaColumna += matriz[l][m];//guardamos la suma de los elementos de la columna m
            }
            if (sumaFila0 != sumaColumna){
                //si la suma de los elementos de la fila 0 es distintade la suma de cada columna no es magica
                magica = false;
            }
        }
        return magica;//devolver si es magica
    }
}
```

aaplicacion5.pkg18.Aaplicacion518 > esMagica

aaplicacion5.pkg18.Aaplicacion518 > esMagica

Output - Aaplicacion5.18 (run)

```
run:
Introducir valor para fila:  1 Columna: 1
1
Introducir valor para fila:  1 Columna: 2
14
Introducir valor para fila:  1 Columna: 3
14
Introducir valor para fila:  1 Columna: 4
4
Introducir valor para fila:  2 Columna: 1
11
Introducir valor para fila:  2 Columna: 2
7
Introducir valor para fila:  2 Columna: 3
6
Introducir valor para fila:  2 Columna: 4
9
Introducir valor para fila:  3 Columna: 1
8
Introducir valor para fila:  3 Columna: 2
10
Introducir valor para fila:  3 Columna: 3
10
Introducir valor para fila:  3 Columna: 4
5
Introducir valor para fila:  4 Columna: 1
13
Introducir valor para fila:  4 Columna: 2
2
Introducir valor para fila:  4 Columna: 3
3
Introducir valor para fila:  4 Columna: 4
15
[[1, 14, 14, 4], [11, 7, 6, 9], [8, 10, 10, 5], [13, 2, 3, 15]]
La matriz es magica: true
BUILD SUCCESSFUL (total time: 17 seconds)
```

64:46 INS