



# BOLETÍN DE EJERCICIOS UD 8

## Aplicación de las estructuras de almacenamiento

### ARRAYS UNIDIMENSIONALES (VECTORES)

1. Hacer un programa que almacene en un vector de 5 elementos las ventas anuales de cada una de las 5 tiendas de una cadena de supermercados (asignárselos al vector al declararlo). A continuación calcular cuál fue la tienda que menos vendió, cuál fue la que más vendió y el importe total de ventas de las 5 tiendas.

```
public class Ud3Ej1 {  
  
    public static void main(String[] args) {  
        double[] ventas = {5000, 315, 444, 188, 6659};  
        double mayor;  
        double menor;  
        double total = 0;  
        int mayornum = 0;  
        int menornum = 0;  
  
        mayor = ventas[0];  
        menor = ventas[0];  
  
        for (int i = 0; i < ventas.length; i++) {  
            if (ventas[i] > mayor) {  
                mayor = ventas[i];  
                mayornum = i;  
            }  
            if (ventas[i] < menor) {  
                menor = ventas[i];  
                menornum = i;  
            }  
            total += ventas[i];  
        }  
        System.out.println("La tienda que menos vendió fue la " + (menornum + 1) + " con " + menor + " €");  
        System.out.println("La tienda que más vendió fue la " + (mayornum + 1) + " con " + mayor + " €");  
        System.out.println("El total de las ventas es: " + total + " €");  
    }  
}
```



2. Almacenar en un vector 5 numeros enteros leídos por teclado. Leer a continuación otro número y comprobar si está en el vector o no. En el caso de que esté visualizar que posición ocupa; sino indicarlo mediante un mensaje. Visualizar también el elemento más pequeño, el más grande y la posición de ambos.

Solución 1:

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

```
public class Ud3Ej2 {  
    public static void main(String[] args) {  
        int[] vector = new int[5];  
        int menor = 0;  
        int mayor = 0;  
        int posMenor = 0;  
        int posMayor = 0;  
        int num;  
        int cont;  
        Scanner leo = new Scanner(System.in);  
  
        for (cont = 0; cont < vector.length; cont++) {  
            System.out.println("introduzca un valor:");  
            vector[cont] = leo.nextInt();  
            if (cont == 0) {  
                menor = vector[0];  
                mayor = vector[0];  
            }  
            if (vector[cont] < menor) {  
                menor = vector[cont];  
                posMenor = cont;  
            }  
            if (vector[cont] > mayor) {  
                mayor = vector[cont];  
                posMayor = cont;  
            }  
        }  
        System.out.println("Introduzca un valor para comprobar que esta en el vector: ");  
        num = leo.nextInt();  
        for (cont = 0; cont < vector.length; cont++) {  
            if (num == vector[cont]) {  
                System.out.println("su posición es: " + (cont + 1));  
                break;  
            } else {  
                if (cont == vector.length - 1) {  
                    System.out.println("No esta en el vector");  
                }  
            }  
        }  
        System.out.println("El elemento más grande: " + mayor + " y su posición: " + (posMayor + 1));  
        System.out.println("El elemento más pequeño: " + menor + " y su posición: " + (posMenor + 1));  
    }  
}
```



```
}  
}
```

## Solución 2:

Modificar la solución anterior utilizando una variable de tipo lógico que refleje si el número a buscar está o no en el vector.

```
import java.util.Scanner;  
  
public class Ud3ej2 {  
  
    public static void main(String[] args) {  
        int[] vector = new int[5];  
        int menor = 0;  
        int mayor = 0;  
        int posMenor = 0;  
        int posMayor = 0;  
        int num;  
        int cont;  
        boolean encontrado = false;  
        Scanner leo = new Scanner(System.in);  
  
        for (cont = 0; cont < vector.length; cont++) {  
            System.out.println("Introduzca un valor:");  
            vector[cont] = leo.nextInt();  
            if (cont == 0) {  
                menor = vector[0];  
                mayor = vector[0];  
            }  
            if (vector[cont] < menor) {  
                menor = vector[cont];  
                posMenor = cont;  
            }  
            if (vector[cont] > mayor) {  
                mayor = vector[cont];  
                posMayor = cont;  
            }  
        }  
        System.out.println("Introduzca un valor para comprobar que está en el vector: ");  
        num = leo.nextInt();  
        for (cont = 0; cont < vector.length; cont++) {  
            if (num == vector[cont]) {  
                System.out.println("Su posición es: " + (cont + 1));  
                encontrado = true;  
                break;  
            }  
        }  
        if (encontrado == false) {  
            System.out.println("No está en el vector.");  
        }  
    }  
}
```



```
}
```

```
System.out.println("El elemento más grande: " + mayor + " y su posición: " + (posMayor + 1));
System.out.println("El elemento más pequeño: " + menor + " y su posición: " + (posMenor + 1));
```

```
}
```

```
}
```

3. Asignarle a un vector de 50 elementos valores aleatorios comprendidos entre 1 y 40. Escribir el contenido del vector. A continuación comprobar si un número leído por teclado está en el vector, escribiendo la posición que ocupa. En el caso de que el número esté repetido, tiene que visualizarse un mensaje con el formato:

El número 28 aparece 4 veces en las posiciones: 6, 15, 18 y 35

```
import java.util.Random;
import java.util.Scanner;
```

```
public class Ud3Ej03 {
```

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

```
        int[] array = new int[50];
        Random aleatorio = new Random();
        int numero;
        int veces = 0;
```

```
        Scanner leo = new Scanner(System.in);
```

```
        for (int i = 0; i < array.length; i++) {
            //damos valores aleatorios de 1 a 40
            array[i] = aleatorio.nextInt(40) + 1;
            System.out.print(array[i] + ", ");
        }
```

```
        System.out.println("");
        System.out.println("Introduce un numero para buscar: ");
        numero = leo.nextInt();
```

```
        //recorremos el array en su busca
        for (int i = 0; i < array.length; i++) {
            if (array[i] == numero) {
                veces++;
            }
        }
```

```
        if (veces == 0) {
            System.out.println("El número " + numero + " no aparece en el array.");
        }
```

```
        if (veces == 1) {
            System.out.print("El número " + numero + " aparece 1 vez en la posición: ");
```



```

        for (int i = 0; i < array.length; i++) {
            if (array[i] == numero){
                System.out.println(i + ".");
                break;
            }
        }
    }
}

if (veces > 1){
    System.out.print("El número " + numero + " aparece " + veces +
        " veces en las posiciones: ");
    int cont = 1;
    for (int i = 0; i < array.length; i++) {

        if (array[i] == numero){
            if (cont == veces){//último posición
                System.out.println(i);
                break;
            } else {
                if (cont == veces-1){//penúltima posición
                    System.out.print(i + " y ");
                    cont++;
                } else {//resto de posiciones
                    System.out.print(i + ", ");
                    cont++;
                }
            }
        }
    }
}
}
}
}
}
}
}
}

```

4. Hacer un programa que simule el lanzamiento de un dado 10.000 veces. Almacenar en un vector de 6 posiciones el número de veces que salió cada tirada, escribiendo a continuación el número de veces que salió cada cara del dado.

Calcular el tiempo que tarda en asignar al vector los 10.000 valores. Utilizar el método `currentTimeMillis()` de la clase `System`.

```

t0 = System.currentTimeMillis();
t1 = System.currentTimeMillis();
long tv = t1-t0;

```

```

import java.util.Random;

```

```

public class Ud3Ej4 {

    public static void main(String[] args) {

```



```
int cont;
long t0;
long t1;
int[] lanzamiento = new int[10000];
int[] almacenar = new int[6];
Random azar = new Random();
System.out.println("Generando numeros al azar...");
t0 = System.currentTimeMillis();

//guardamos en lanzamientos las 10000 tiradas
for (cont = 0; cont < lanzamiento.length; cont++) {
    lanzamiento[cont] = azar.nextInt(6) + 1;
}

for (cont = 0; cont < lanzamiento.length; cont++) {
    if (lanzamiento[cont] == 1) {
        almacenar[0]++;
    }
    if (lanzamiento[cont] == 2) {
        almacenar[1]++;
    }
    if (lanzamiento[cont] == 3) {
        almacenar[2]++;
    }
    if (lanzamiento[cont] == 4) {
        almacenar[3]++;
    }
    if (lanzamiento[cont] == 5) {
        almacenar[4]++;
    }
    if (lanzamiento[cont] == 6) {
        almacenar[5]++;
    }
}
for (cont = 0; cont < almacenar.length; cont++) {
    System.out.println("El numero " + (cont + 1) + " salió: " + almacenar[cont] + " veces");
}
t1 = System.currentTimeMillis();
System.out.println("El tiempo que necesita para calcular las tiradas es : " + (t1 - t0) + " milisegundos");
}
```

El mismo código más sencillo quedaría:

```
package ud3ej4;

import java.util.Random;

public class Ud3Ej4 {

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



```

long t0;
long t1;
int[] lanzamiento = new int[10000];
int[] almacenar = new int[6];
Random azar = new Random();
System.out.println("Generando numeros al azar...");
t0 = System.currentTimeMillis();

//guardamos en lanzamientos las 10000 tiradas
for (cont = 0; cont < lanzamiento.length; cont++) {
    lanzamiento[cont] = azar.nextInt(6) + 1;
}

for (cont = 0; cont < lanzamiento.length; cont++) {
    almacenar[lanzamiento[cont] - 1]++;
}
for (cont = 0; cont < almacenar.length; cont++) {
    System.out.println("El numero " + (cont + 1) + " salió: " + almacenar[cont] + " veces");
}
t1 = System.currentTimeMillis();
System.out.println("El tiempo que necesita para calcular las tiradas es : " + (t1 - t0) + " milisegundos");
}
}

```

5. Hacer un programa que almacene en un vector de 100 elementos números generados al azar comprendidos entre 20 y 50. Escribir los números generados, visualizando 20 números por línea. Escribir a continuación cuántos números fueron generados entre 21 y 30. La salida tiene que tener el forma to:

```

El 21 apareció XX veces
El 22 apareció XX veces
.....
El 30 apareció XX veces

```

```

package ud3ej5;

import java.util.Random;

public class Ud3Ej5 {

    public static void main(String[] args) {
        int[] vector = new int[100];
        int[] cuenta = new int[10];
        Random azar = new Random();
        int cont;
        int linea = 19;

        //recorro el vector, le doy valores y los escribo
        for (cont = 0; cont < vector.length; cont++) {
            vector[cont] = azar.nextInt(31) + 20;

```



```
        if (cont == linea) {
            System.out.println(vector[cont] + " ");
            linea = linea + 20;
        } else {
            System.out.print(vector[cont] + ", ");
        }
    }

//recorro vector para ver cuántas veces aparece
//cada número
for (cont = 0; cont < vector.length; cont++) {
    if (vector[cont] == 21) {
        cuenta[0]++;
    }
    if (vector[cont] == 22) {
        cuenta[1]++;
    }
    if (vector[cont] == 23) {
        cuenta[2]++;
    }
    if (vector[cont] == 24) {
        cuenta[3]++;
    }
    if (vector[cont] == 25) {
        cuenta[4]++;
    }
    if (vector[cont] == 26) {
        cuenta[5]++;
    }
    if (vector[cont] == 27) {
        cuenta[6]++;
    }
    if (vector[cont] == 28) {
        cuenta[7]++;
    }
    if (vector[cont] == 29) {
        cuenta[8]++;
    }
    if (vector[cont] == 30) {
        cuenta[9]++;
    }
}

for (cont = 0; cont < cuenta.length; cont++) {
    System.out.println("El " + (cont + 21) + " apareció " + cuenta[cont] + " veces.");
}

}
```

Ejercicio: modificar el ejercicio anterior de tal manera que el buda que cuenta cuántas veces aparecen los números solo tenga una instrucción dentro.





6. Hacer un programa que dado un número de DNI obtenga la letra del NIF. La letra correspondiente a un número de DNI se calcula mediante el siguiente algoritmo. Se obtiene el resto de dividir el número de DNI entre 23. El número resultante nos indica la posición de la letra correspondiente a ese DNI, en la siguiente cadena:

Tabla de asignación

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

No se utilizan las letras: I, Ñ, O, U.

La I y la O se evitan para evitar confusiones con otros caracteres, como 1, l ó 0.

NOTA: hacer este programa utilizando un vector para almacenar cada una de las letras de la tabla anterior.

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

```
public class Ud3Ej6 {
```

```
    public static void main(String[] args) {  
        char[] arrayLetra = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K',  
        'E'};  
        int dni;  
        int posNif;  
        Scanner leo = new Scanner(System.in);  
  
        System.out.println("Mete el dni: ");  
        dni = leo.nextInt();  
        posNif = dni % 23;  
  
        System.out.println("letra: " + arrayLetra[posNif]);  
    }  
}
```

Otra solución:

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

```
public class Ud3Ej6 {
```

```
    public static void main(String[] args) {  
        char[] arrayLetra = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K',  
        'E'};  
        int dni;  
        Scanner leo = new Scanner(System.in);  
  
        System.out.println("Mete el dni: ");  
        dni = leo.nextInt();  
        System.out.println("letra: " + arrayLetra[dni % 23]);  
    }  
}
```



7. Verificar si una cadena de texto leída por teclado es un NIF correcto o no. Suponer que los NIFs tienen 8 dígitos y a continuación una letra (no importa que sea mayúscula o minúscula).

PISTAS: dos condiciones que debe cumplir el NIF: tener 9 caracteres y que el último sea una letra. Comprobado esto, verificar que el resto de caracteres son dígitos.

Usar el método `length()` de `java.lang.String` para conocer el número de caracteres de una cadena de texto.

Usar el método estático `isLetter(char c)` de `java.lang.Character` para comprobar que un carácter es una letra.

Usar el método estático `isDigit(char c)` de `java.lang.Character` para comprobar que un carácter es un dígito.

Usar el método `substring(int inicio, int fin)` de `java.lang.String` para obtener la parte numérica del nif

8. Dada la clase `Fecha`, modificar el método `mesLetra` declarando un vector de `Strings` que represente todos los meses y de esta manera que el código del método quede mas corto.

```
public class Fecha {  
  
    private int dia;  
    private int mes;  
    private int anyo;  
  
    public int getAnyo() {  
        return anyo;  
    }  
  
    public void setAnyo(int anyo) {  
        this.anyo = anyo;  
    }  
  
    public int getDia() {  
        return dia;  
    }  
  
    public void setDia(int dia) {  
        this.dia = dia;  
    }  
  
    public int getMes() {  
        return mes;  
    }  
  
    public void setMes(int mes) {  
        this.mes = mes;  
    }  
  
    public Fecha(int dia, int mes, int anyo) {  
        this.dia = dia;  
        this.mes = mes;  
    }  
}
```



```
this.anyo = anyo;
}

public Fecha() {
    this.dia = 1;
    this.mes = 1;
    this.anyo = 2013;
}

public void asignate(int dia, int mes, int anyo) {
    this.dia = dia;
    this.mes = mes;
    this.anyo = anyo;
}

public void incrementate(int d) {

    for (; d > 0; d--) {
        dia++;
        switch (mes) {
            case 4:
            case 6:
            case 9:
            case 11:
                if (dia == 31) {
                    mes++;
                    dia = 1;
                }
                break;
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                if (dia == 32) {
                    mes++;
                    dia = 1;
                    if (mes == 13) {
                        anyo++;
                        mes = 1;
                    }
                }
                break;
            case 2:
                if ((anyo % 4 == 0 && anyo % 100 != 0) || (anyo % 400 == 0)) { //bisiesto
                    if (dia == 30) {
                        mes++;
                        dia = 1;
                    }
                } else {
```



```
        if (dia == 29) {
            mes++;
            dia = 1;
        }
    }
}
}

//metodo mesLetra para pasar el mes de numero a letra
private String mesLetra() {
    String mesString = "";

    switch (mes) {
        case 1:
            mesString = "Enero";
            break;
        case 2:
            mesString = "Febrero";
            break;
        case 3:
            mesString = "Marzo";
            break;
        case 4:
            mesString = "Abril";
            break;
        case 5:
            mesString = "Mayo";
            break;
        case 6:
            mesString = "Junio";
            break;
        case 7:
            mesString = "Julio";
            break;
        case 8:
            mesString = "Agosto";
            break;
        case 9:
            mesString = "Septiembre";
            break;
        case 10:
            mesString = "Octubre";
            break;
        case 11:
            mesString = "Noviembre";
            break;
        case 12:
            mesString = "Diciembre";
            break;
    }
    return mesString;
}
```



```

    }

    public void imprimete() {
        System.out.println(dia + "-" + mesLetra() + "-" + anyo);
    }

    public void imprimete(int fecha) {
        System.out.println(dia + "-" + mes + "-" + anyo);
    }

    public String getFecha() {
        return dia + "-" + mesLetra() + "-" + anyo;
    }
}

```

////////

```

public class PruebaFecha {
    public static void main(String[] args) {
        Fecha fechaCompra = new Fecha(30,11,2013);

        fechaCompra.imprimete();
        fechaCompra.imprimete(6);

        //incrementamos la fecha en 32 dias
        fechaCompra.incrementate(32);
        fechaCompra.imprimete();
        fechaCompra.incrementate(365);
        fechaCompra.imprimete();
    }
}

```

SOLUCIÓN:

```

private String mesLetra() {
    String[] mesString = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
        "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};

    return mesString[mes - 1];
}

```

- Hacer un programa en el que se declare un vector de 10 objetos de tipo Fecha (este vector almacenaría 10 fechas). A la hora de crear el vector, todos sus elementos tienen que tener la fecha (1 de Abril de 2011). Asignarle a continuación a todas las fechas, valores generados de manera aleatoria. Por simplificar, todos los meses son de 30 días, y el año tiene que estar comprendido entre 2000 y 2008, ambos incluidos. Imprimir el contenido del vector, una vez generado, y una vez asignadas las nuevas fechas. Incrementara continuación todas las fechas en 3 días, y volver a escribir el contenido del vector para comprobar que se hizo de manera correcta.



```
public class Fecha {  
  
    private int dia;  
    private int mes;  
    private int anyo;  
  
    public int getAnyo() {  
        return anyo;  
    }  
  
    public void setAnyo(int anyo) {  
        this.anyo = anyo;  
    }  
  
    public int getDia() {  
        return dia;  
    }  
  
    public void setDia(int dia) {  
        this.dia = dia;  
    }  
  
    public int getMes() {  
        return mes;  
    }  
  
    public void setMes(int mes) {  
        this.mes = mes;  
    }  
  
    public Fecha(int dia, int mes, int anyo) {  
        this.dia = dia;  
        this.mes = mes;  
        this.anyo = anyo;  
    }  
  
    public Fecha() {  
        this.dia = 1;  
        this.mes = 1;  
        this.anyo = 2013;  
    }  
  
    public void asignate(int dia, int mes, int anyo) {  
        this.dia = dia;  
        this.mes = mes;  
        this.anyo = anyo;  
    }  
  
    public void incrementate(int d) {
```



```
for (; d > 0; d--) {  
    dia++;  
    switch (mes) {  
        case 4:  
        case 6:  
        case 9:  
        case 11:  
            if (dia == 31) {  
                mes++;  
                dia = 1;  
            }  
            break;  
        case 1:  
        case 3:  
        case 5:  
        case 7:  
        case 8:  
        case 10:  
        case 12:  
            if (dia == 32) {  
                mes++;  
                dia = 1;  
                if (mes == 13) {  
                    anyo++;  
                    mes = 1;  
                }  
            }  
            break;  
        case 2:  
            if ((anyo % 4 == 0 && anyo % 100 != 0) || (anyo % 400 == 0)) { //bisiesto  
                if (dia == 30) {  
                    mes++;  
                    dia = 1;  
                }  
            } else {  
                if (dia == 29) {  
                    mes++;  
                    dia = 1;  
                }  
            }  
        }  
    }  
}
```

```
//metodo mesLetra para pasar el mes de numero a letra  
private String mesLetra() {  
    String[] mesString = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",  
        "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};  
  
    return mesString[mes - 1];  
}
```



```
public void imprime() {
    System.out.println(dia + "-" + mesLetra() + "-" + anyo);
}

public void imprime(int fecha) {
    System.out.println(dia + "-" + mes + "-" + anyo);
}

public String getFecha() {
    return dia + "-" + mesLetra() + "-" + anyo;
}
}

////////////////////////////////////
public class Ud3Ej9 {

    public static void main(String[] args) {
        Fecha[] listaFechas = new Fecha[10];
        int cont;

        //Construimos los 10 objetos del vector
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont] = new Fecha(1, 4, 2011);
        }

        //Recorremos el vector y escribimos las fechas que contiene
        System.out.println("Contenido del vector al crearlo: ");
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont].imprime();
        }

        //recorremos el vector y le damos valores aleatorios//
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont].setDia(aux.nextInt(30) + 1);
            listaFechas[cont].setMes(aux.nextInt(12) + 1);
            listaFechas[cont].setAnyo(aux.nextInt(9) + 2000);
        }
        System.out.println("Contenido del vector con fechas aleatorias: ");
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont].imprime();
        }

        //recorremos el vector y incrementamos todas las fechas en tres dias//
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont].incrementate(3);
        }
        System.out.println("Contenido del vector con fechas incrementadas en 3 dias: ");
        for (cont = 0; cont < listaFechas.length; cont++) {
            listaFechas[cont].imprime();
        }
    }
}
```





10. Hacer un programa en el que se declare un vector de 5 datos de tipo `EcuacionSegundoGrado`. A continuación, de manera aleatoria, asignar al vector 5 ecuaciones que tengan solución. A la propiedad `a` hay que darle valores aleatorios comprendidos entre -5 y 10, a la propiedad `b` valores comprendidos entre 1 y 5 y a la propiedad `c` valores comprendidos entre 1 y 100. Escribir los valores de las cinco ecuaciones, así como de sus soluciones. Escribir también cuántas ecuaciones se generaron hasta dar con una válida para cada elemento del vector. El formato de salida del programa será:

Ecuación 1:

Coefficiente a: xx

Coefficiente b: xx

Coefficiente c: xx

Solución 1: xx

Solución 2: xx

Ecuaciones generadas hasta obtener una con soluciones: xx

```
public class EcuacionSegundoGrado {  
  
    private double a;  
    private double b;  
    private double c;  
  
    public EcuacionSegundoGrado(double a, double b, double c) {  
        this.a = a;  
        this.b = b;  
        this.c = c;  
    }  
  
    public EcuacionSegundoGrado() {  
    }  
  
    public void setCoeficientes(double a, double b, double c) {  
        this.a = a;  
        this.b = b;  
        this.c = c;  
    }  
  
    public double getA() {  
        return a;  
    }  
  
    public double getB() {  
        return b;  
    }  
  
    public double getC() {
```



```
        return c;
    }

    public void setA(double a) {
        this.a = a;
    }

    public void setB(double b) {
        this.b = b;
    }

    public void setC(double c) {
        this.c = c;
    }

    public boolean averigua() {
        if ( a == 0 || ((b*b)-(4*a*c)) < 0 ){
            return false;
        } else {
            return true;
        }
    }

    public double obternerSolucion1(){
        return (-b + Math.sqrt(b*b-4*a*c))/(2*a);
    }

    public double obternerSolucion2(){
        return (-b - Math.sqrt(b*b-4*a*c))/(2*a);
    }
}
```

### Solución:

```
import java.util.Random;

public class Ud3Ej10 {

    public static void main(String[] args) {
        EcuacionSegundoGrado[] ecuaciones = new EcuacionSegundoGrado[5];
        Random aleatorio = new Random();
        int n;
        int cont;

        //recorremos el vector para construir los objetos del mismo
        for (int i = 0; i < ecuaciones.length; i++) {
            ecuaciones[i] = new EcuacionSegundoGrado();

            cont = 0;
            while (true) {
```



```
//el valor para a tiene que estar comprendido entre -5 y 10
//son 16 valores. genero un número al azar comprendido entre 1 y 16
n = aleatorio.nextInt(16) + 1;
//ahora le resto 6 al valor generado para que esté en el rango -5 y 10
n -= 6;
ecuaciones[i].setA(n);

n = aleatorio.nextInt(5) + 1;
ecuaciones[i].setB(n);

n = aleatorio.nextInt(100);
ecuaciones[i].setC(n);
cont++;
if (ecuaciones[i].averigua()) {
    break;
}

}

System.out.println("Ecuación " + (i + 1) + ":");
System.out.println("\nCoeficiente a: " + ecuaciones[i].getA());
System.out.println("Coeficiente b: " + ecuaciones[i].getB());
System.out.println("Coeficiente c: " + ecuaciones[i].getC());

System.out.println("\nSolución 1: " + ecuaciones[i].obtenerSolucion1());
System.out.println("Solución 2: " + ecuaciones[i].obtenerSolucion2());

System.out.println("\nEcuaciones generadas hasta obtener una con soluciones: " + cont + "\n\n");
}
}
}
```

11. Declarar un vector de nombre lista que estará formado por 25 datos de tipo entero generados al azar comprendidos entre 25 y 50.

Declarar otro vector de nombre auxi formado por 4 datos de tipo real. Almacenar en este vector lo siguiente:

Posición 0: número de elementos pares en lista  
Posición 1: número de elementos impares de lista  
Posición 2: media de los términos impares de lista  
Posición 3: suma de los números enteros comprendidos entre 1 y el valor par más pequeño del vector lista

Escribir a continuación el contenido del vector lista, así como los resultados obtenidos y almacenados en auxi.

12. Escribir un programa en el que se declare un vector de 50 números enteros de nombre numeros. Asignarle al vector valores al azar comprendidos entre 20 y 80. Declarar otro vector de nombre calculos de 4 posiciones de tipo real. Almacenar en las posiciones de ese vector de forma consecutiva lo siguiente:

- Elemento mas pequeño de numeros



- Elemento mayor de numeros
- Suma de todos los elementos de numeros
- Media aritmética de todos los elementos de numeros

Escribir a continuación los elementos del vector y los cuatro valores calculados

```
import java.util.Random;
```

```
public class Ud3Ej14 {
```

```
    public static void main(String[] args) {  
        int[] numeros = new int[50];  
        double [] calculos = new double[4];  
        calculos[0] = 80;  
        calculos[1] = 20;  
        calculos[2] = 0;  
        Random aleatorio = new Random();
```

```
        for (int i = 0; i < numeros.length; i++) {  
            numeros[i] = aleatorio.nextInt(61) + 20;  
            if (numeros[i] < calculos[0]) {  
                calculos[0] = numeros[i];  
            }  
            if (numeros[i] > calculos[1]) {  
                calculos[1] = numeros[i];  
            }  
            calculos[2] += numeros[i];  
            System.out.print(numeros[i] + " ");  
        }
```

```
        calculos[3] = calculos[2] / 50;  
        System.out.println(" ");  
        System.out.println("Elemento mas pequeño de números: " + calculos[0]);  
        System.out.println("Elemento mayor de números: " + calculos[1]);  
        System.out.println("Suma de todos los elementos de números: " + calculos[2]);  
        System.out.println("Media aritmética de todos los elementos de números: " + calculos[3]);  
    }  
}
```



13. Hacer un programa en el que se declaren 3 vectores de nombres: nombre, apellido1 y apellido2, que tengan cada uno de ellos 10 nombres y 10 apellidos, dándole valores al declararlos sin que haya valores repetidos.

Declarar otro vector de nombre nombresApellidos de longitud 30, que contenga el contenido de los 3 vectores anteriores. Primero irán los nombres y después los apellidos. Asignarle valores a este vector utilizando el método **arrayCopy**.

- Escribir a continuación el nombre y apellidos de 5 personas, eligiendo aleatoriamente los valores de los vectores nombre y apellidos. Estos 5 valores tienen que almacenarse previamente en un vector de 5 Strings.
- Repetir el apartado a) eligiendo los valores del vector nombresApellidos.
- (opcional) Los nombres y apellidos no pueden estar repetidos.

```
package ud3ej15;

import java.util.Random;

public class Ud3Ej15 {

    public static void main(String[] args) {
        String[] nombre =
{"Juan", "Luis", "Dani", "Raul", "Marcos", "Jose", "Celia", "Alba", "Sara", "Patricia"};
        String[] apellido1 = {"Lozano", "Iglesias", "Dorribo", "Swathssenager", "Del
Toro", "González", "Quintas", "Loureiro", "Carbonero", "Conde"};
        String[] apellido2 =
{"Fernandez", "Carballo", "Panza", "Balboa", "Gayoso", "Gómez", "Cobas", "Cárdenas", "Lippo",
"Vázquez"};
        String[] nombresApellidos = new String [30];
        String[] personas5 = new String [5];
        Random aleatorio = new Random();

        System.arraycopy (nombre, 0, nombresApellidos, 0, 10);
        System.arraycopy (apellido1, 0, nombresApellidos, 10, 10);
        System.arraycopy (apellido2, 0, nombresApellidos, 20, 10);

        System.out.println("Nombres obtenidos de los tres vectores: ");
        for (int i = 0; i < personas5.length; i++) {
            personas5[i] = (nombre[aleatorio.nextInt(10)]+ " " + apellido1[aleatorio.nextInt(10)]+ " "
+apellido2[aleatorio.nextInt(10)]);
            System.out.println(personas5[i]);
        }

        System.out.println("Nombres obtenidos del vector nombresApellidos: ");
        for (int i = 0; i < personas5.length; i++) {
            personas5[i] = (nombresApellidos[aleatorio.nextInt(10)]+ " " +
nombresApellidos[aleatorio.nextInt(10)+ 10]+ " " +nombresApellidos[aleatorio.nextInt(10)+20]);
```



```
        System.out.println(personas5[i]);
    }
}
}
```

14. Visualizar el contenido del vector de 5 strings utilizado en el apartado a) del ejercicio anterior ordenado alfabéticamente.
15. Asignarle a un vector 100 valores de tipo entero comprendidos entre 1 y 200. Leer por teclado un número y comprobar si coincide con alguno de los elementos del vector. Utilizar el método `binarySearch` de la clase `Arrays`.

```
public static int binarySearch(int[] a, int key)
```

Searches the specified array of ints for the specified value using the binary search algorithm. The array must be sorted (as by the [sort\(int\[\]\)](#) method) prior to making this call. If it is not sorted, the results are undefined. If the array contains multiple elements with the specified value, there is no guarantee which one will be found.

Parameters:

a - the array to be searched  
key - the value to be searched for

Returns:

index of the search key, if it is contained in the array; otherwise,  $-(\text{insertion point}) - 1$ . The *insertion point* is defined as the point at which the key would be inserted into the array: the index of the first element greater than the key, or `a.length` if all elements in the array are less than the specified key. Note that this guarantees that the return value will be  $\geq 0$  if and only if the key is found.

16. Asignarle a un vector 3000 números enteros comprendidos entre 1 y 2000. Escribir el contenido del vector ordenado de menor a mayor. Utilizar el método `sort` de la clase `Arrays`.
17. Hacer un programa que pida por teclado un número `n` y que escriba los primos comprendidos entre 1 y `n`. Utilizar el algoritmo de la criba de Eratóstenes.

La criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores que un número natural dado `n`. Se forma una tabla con todos los números naturales comprendidos entre 2 y `n`, y se van tachando los números que no son primos de la siguiente manera: Comenzando por el 2, se tachan todos sus múltiplos; comenzando de nuevo, cuando se encuentra un número entero que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente. El proceso termina cuando el cuadrado del mayor número confirmado como primo es mayor que `n`.

[http://es.wikipedia.org/wiki/Criba\\_de\\_Erat%C3%B3stenes](http://es.wikipedia.org/wiki/Criba_de_Erat%C3%B3stenes)



## ARRAYS BIDIMENSIONALES (TABLAS O MATRICES)

Ejemplo de un programa que utiliza una tabla donde sus filas no tienen el mismo número de columnas:

```
import java.util.Random;

public class Tabla {
    public static void main(String[] args) {
        int tabla[][] = {{4,3,2,5,6},
                        {4,5,6},
                        {1,2}};

        Random rnd = new Random();

        for(int i = 0 ;i < tabla.length; i++){
            for(int j = 0;j < tabla[i].length; j++){
                tabla[i][j] = rnd.nextInt(32);
            }
        }

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

18. Hacer un programa en el que se declare una matriz de dimensiones 5 x 8 (5 filas y 8 columnas) de datos de tipo entero. Asignarle a los elementos de la matriz valores generados al azar y comprendidos entre 20 y 40. Recorrer a continuación la matriz y escribir en pantalla el valor de sus elementos.

```
import java.util.Random;

public class Tabla {

    public static void main(String[] args) {
        int[][] tabla = new int[5][8];
        Random rnd = new Random();

        for (int i = 0; i < 5; i++) { //i < tabla.length
            for (int j = 0; j < 8; j++) { //tabla[0].length
                tabla[i][j] = rnd.nextInt(21) + 20;
            }
        }

        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 8; j++) {
```



```
        System.out.print(tabla[i][j] + " ");
    }
    System.out.println();
}
}
}
```

19. Hacer un programa en el que se declare una matriz de dimensiones 5 x 8 (5 filas y 8 columnas) de datos de tipo entero. Asignarle a los elementos de la matriz valores generados al azar y comprendidos entre 20 y 40. Calcular cuál es el elemento mas pequeño y mayor de la tabla. Escribir también la tabla para comprobar los resultados. La salida tiene que tener el formato:

```
Elemento menor:  XX          Fila: XX Columna: XX
Elemento mayor:  XX          Fila: XX Columna: XX
```

```
import java.util.Random;
```

```
public class Ud3Ej21 {

    public static void main(String[] args) {
        int tabla[][] = new int[5][8];
        Random azar = new Random();
        int mayor = 0;
        int filaMayor = 0;
        int columnaMayor = 0;
        int pequeno = 40;
        int filaPequeno = 0;
        int columnaPequeno = 0;

        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 8; j++) {
                tabla[i][j] = azar.nextInt(21) + 20;
            }
        }

        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 8; j++) {
                System.out.print(tabla[i][j] + " ");

            }
            System.out.println();
        }

        //Recorro la tabla para encontrar el más pequeño.
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 8; j++) {
                if (tabla[i][j] < pequeno) {
                    pequeno = tabla[i][j];
                    filaPequeno = i;
                    columnaPequeno = j;
                }
            }
        }
    }
}
```





```

    }
}

//Recorro la tabla para encontrar el más grande.
for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 8; j++) {
        if (tabla[i][j] > mayor) {
            mayor = tabla[i][j];
            filaMayor = i;
            columnaMayor = j;
        }
    }
}

```

```

System.out.println("Elemento menor: " + pequeno + "Fila: " + filaPequeno + "columna: " +
columnaPequeno);
System.out.println("Elemento Mayor: " + mayor + "Fila: " + filaMayor + "columna: " + columnaMayor);

}
}

```

#### Otra solución recorriendo la tabla una vez:

```

package ud3ej21;

import java.util.Random;

public class Ud3Ej21 {

    public static void main(String[] args) {
        int tabla[][] = new int[5][8];
        Random azar = new Random();
        int mayor = 0;
        int filaMayor = 0;
        int columnaMayor = 0;
        int pequeno = 40;
        int filaPequeno = 0;
        int columnaPequeno = 0;

        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 8; j++) {
                tabla[i][j] = azar.nextInt(21) + 20;
                System.out.print(tabla[i][j] + " ");
                if (tabla[i][j] < pequeno) {
                    pequeno = tabla[i][j];
                    filaPequeno = i;
                    columnaPequeno = j;
                }
                if (tabla[i][j] > mayor) {
                    mayor = tabla[i][j];
                    filaMayor = i;
                }
            }
        }
    }
}

```



```
        columnaMayor = j;
    }
}
System.out.println();
}
```

```
System.out.println("Elemento menor: " + pequeno + " Fila: " + filaPequeno + " columna: " +
columnaPequeno);
System.out.println("Elemento Mayor: " + mayor + " Fila:" + filaMayor + " columna: " + columnaMayor);
}
}
```

20. Repetir el ejercicio anterior dándole a la tabla unas dimensiones de 6 x9, de tal manera que si el elemento menor o el mayor aparecen varias veces, visualicemos cuántas veces aparecen y en que posiciones están. Si solo están una vez no hay que visualizar que aparecen 1 vez.

Ejemplo de salida:

```
Elemento menor:  23          Fila: 1  Columna: 3
Elemento mayor:  39          Aparece 2 veces en:

Fila: 2      Columna: 3
Fila: 1      Columna: 5
```

```
import java.util.Random;
```

```
public class Ud3ej22 {

    public static void main(String[] args) {
        int tabla[][] = new int[6][9];
        Random azar = new Random();
        int mayor = 0;
        int filaMayor = 0;
        int columnaMayor = 0;
        int pequeno = 40;
        int filaPequeno = 0;
        int columnaPequeno = 0;
        int vecesmayor = 0;
        int vecesmenor = 0;

        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 9; j++) {
                tabla[i][j] = azar.nextInt(21) + 20;
            }
        }

        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 9; j++) {
                System.out.print(tabla[i][j] + " ");
            }
        }
    }
}
```



```
        System.out.println();
    }

    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 9; j++) {
            if (tabla[i][j] < pequeno) {
                pequeno = tabla[i][j];
                filaPequeno = i;
                columnaPequeno = j;
            }
        }
    }

    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 9; j++) {
            if (tabla[i][j] > mayor) {
                mayor = tabla[i][j];
                filaMayor = i;
                columnaMayor = j;
            }
        }
    }
}

//recorro tabla para contar las veces que aparece el más pequeño
//y el mas grande
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 9; j++) {
        if (tabla[i][j] == pequeno) {
            vecesmenor++;
        }
        if (tabla[i][j] == mayor) {
            vecesmayor++;
        }
    }
}

if (vecesmayor == 1) {
    System.out.println("Elemento mayor: " + mayor + " Fila: " + filaMayor + " Columna: " +
columnaMayor);
} else {
    System.out.println("Elemento mayor: " + mayor + " aparece " + vecesmayor + " veces en: ");
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 9; j++) {
            if (tabla[i][j] == mayor) {
                System.out.println(" Fila: " + i + " Columna: " + j);
            }
        }
    }
}

if (vecesmenor == 1) {
```



```
System.out.println("Elemento menor: " + pequeno + " Fila: " + filaPequeno + " Columna: " +
columnaPequeno);
```

```
} else {
    System.out.println("Elemento menor: " + pequeno + " aparece " + vecesmenor + " veces en: ");
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 9; j++) {
            if (tabla[i][j] == pequeno) {
                System.out.println(" Fila: " + i + " Columna: " + j);
            }
        }
    }
}
}
```

21. Hacer un programa que genere y escriba una matriz unitaria de orden 10. Una matriz unitaria de orden 10 es una matriz de 10x10 en la que todos los elementos valen 0, excepto los de la diagonal principal que valen 1. La diagonal principal es la que va de la esquina superior izquierda a la inferior derecha.

```
public class Ud3Ej23 {

    public static void main(String[] args) {
        int[][] unitaria = new int[10][10];
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 10; j++) {
                if (i == j) {
                    unitaria[i][j] = 1;
                } else {
                    unitaria[i][j] = 0;
                }
            }
        }
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 10; j++) {
                System.out.print(unitaria[i][j]);
            }
            System.out.println("");
        }
    }
}
```



22. Hacer un programa que averigüe si una matriz cuadrada de orden 4 es simétrica. Al declarar la matriz asignarle valores que representen una matriz simétrica. Probar el programa con la matriz inicial, y después cambiar uno de los valores para que no sea simétrica. Una matriz es simétrica cuando  $a[f][c]=a[c][f]$  para cualquier elemento de la matriz.

```
public class Ud3Ej24 {

    public static void main(String[] args) {
        int[][] matriz = {{5, 2, 3, 6},
                           {1, 5, 2, 0},
                           {3, 2, 5, 7},
                           {6, 0, 7, 5}};
        boolean simetrica = true;

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                if (matriz[j][i] != matriz[i][j]) {
                    simetrica = false;
                    break;
                }
            }
            if (simetrica == false) {
                break;
            }
        }
        if (simetrica == true) {
            System.out.println("Esta matriz es simetrica");
        } else {
            System.out.println("Esta matriz no es simetrica");
        }
    }
}
```

////////////////////OTRA SOLUCION////////////////////

```
package ud3ej24;

public class Ud3Ej24 {

    public static void main(String[] args) {
        int[][] matriz = {{5, 2, 3, 6},
                           {1, 5, 2, 0},
                           {3, 2, 5, 7},
                           {6, 0, 7, 5}};
        boolean simetrica = true;

        for (int i = 0; i < 4 && simetrica; i++) {
            for (int j = 0; j < 4; j++) {
                if (matriz[j][i] != matriz[i][j]) {
                    simetrica = false;
                    break;
                }
            }
        }
    }
}
```



```

    }
  }
}
if (simetrica == true) {
    System.out.println("Esta matriz es simétrica");
} else {
    System.out.println("Esta matriz no es simétrica");
}
}
}
}

```

23. Hacer un programa que le de valores aleatorios comprendidos entre 5 y 20 ambos incluídos, a una tabla de 3 filas y 7 columnas. Escribir el valor más pequeño de cada fila y el mas grande de cada columna con el formato:

```

Número menor fila 1  xx      está en la columna      xx
Número menor fila 2  xx      está en la columna      xx
.....
Número mayor columna 1 xx  está en la fila          xx
Número mayor columna 2 xx  está en la fila          xx
.....

```

```
import java.util.Random;
```

```
public class Ud3Ej25 {
```

```

    public static void main(String[] args) {
        Random aleatorio = new Random();
        int[][] matriz = new int[3][7];
        int mayor;
        int pequeno;
        int filaMayor;
        int columnaPequeno = 0;

```

```

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 7; j++) {
                matriz[i][j] = aleatorio.nextInt(21) + 5;
                System.out.print(matriz[i][j] + " ");

```

```

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

```

```

        //recorro la tabla para encontrar al mas pequeño de cada fila
        for (int i = 0; i < 3; i++) {
            pequeno=matriz[i][0];
            columnaPequeno=0;
            for (int j = 0; j < 7; j++) {
                if (matriz[i][j] < pequeno) {
                    pequeno = matriz[i][j];

```



```

        columnaPequeno = j;
    }
}

System.out.println("Número menor fila " + (i+1) + " " + pequeno + " esta en la columna " +
(columnaPequeno+1));
}

System.out.println("");
//recorro la tabla para localizar el mayor de las columnas
for (int j = 0; j < 7; j++) {
    mayor=matriz[0][j];
    filaMayor=0;
    for (int i = 0; i < 3; i++) {
        if (matriz[i][j] > mayor) {
            mayor = matriz[i][j];
            filaMayor = i;
        }
    }
}

System.out.println("Número mayor columna " + (j+1) + " " + mayor + " esta en la fila " +
(filaMayor+1));
}
}
}

```

24. Hacer un programa que asigne números a una matriz de orden 4x5 comprendidos entre 1 y 10. El programa tiene que escribir la suma de los elementos de cada fila y la suma de los números que son pares de cada columna con el formato siguiente:

```

suma de los elementos de la fila 0 -->XX
suma de los elementos de la fila 1 -->XX
.....
suma de los elementos de la fila 3 -->XX
-----
suma de los elementos pares de la columna 0 -->XX
suma de los elementos pares de la columna 1 -->XX
.....
suma de los elementos pares de la columna 4 -->XX

```

```
import java.util.Random;
```

```
public class Ud3Ej26 {
```

```

    public static void main(String[] args) {
        Random aleatorio = new Random();
        int[][] matriz = new int[4][5];
        int sumaFilas = 0;
        int sumaPares = 0;

```

```

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 5; j++) {

```



```

        matriz[i][j] = aleatorio.nextInt(10) + 1;
        System.out.print(matriz[i][j] + " ");
    }
    System.out.println();
}
System.out.println("");

for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 5; j++) {
        sumaFilas = sumaFilas + matriz[i][j];
    }
    System.out.println("La Suma de los elementos de la fila " + (i) + " : " + sumaFilas);
    sumaFilas = 0;
}
for (int j = 0; j < 5; j++) {
    for (int i = 0; i < 4; i++) {
        if (matriz[i][j] % 2 == 0) {
            sumaPares += matriz[i][j];
        }
    }
    System.out.println("La Suma de los elementos pares de la columna "
        + (j) + " : " + sumaPares);
    sumaPares = 0;
}

}
}

```

25. Repetir el programa anterior, pero la suma de los elementos de cada fila tiene que almacenarse en un vector de 4 posiciones y la suma de los elementos pares de cada columna tiene que almacenarse en un vector de 5 posiciones.

```

import java.util.Random;

public class Ud3Ej27 {

    public static void main(String[] args) {
        Random aleatorio = new Random();
        int[][] matriz = new int[4][5];
        int[] sumaFil = {0, 0, 0, 0};
        int[] sumaCol = {0, 0, 0, 0, 0};

        //recorremos la tabla para dar valores y escribir en pantalla
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 5; j++) {
                matriz[i][j] = aleatorio.nextInt(10) + 1;
                System.out.print(matriz[i][j] + " ");
            }
        }
    }
}

```





```
        System.out.println();
    }
    System.out.println("");

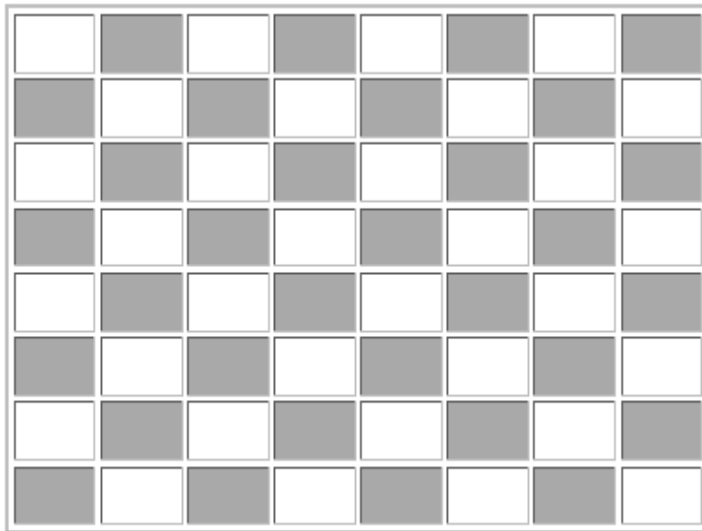
    //recorremos la tabla y guardamos en el vector sumaFil
    //la suma de cada fila
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 5; j++) {
            sumaFil[i] += matriz[i][j];
        }
    }

    //recorremos la tabla y guardamos en el vector sumaCol
    //la suma de los pares de cada columna
    for (int j = 0; j < 5; j++) {
        for (int i = 0; i < 4; i++) {
            if (matriz[i][j] % 2 == 0) {
                sumaCol[j] += matriz[i][j];
            }
        }
    }

    //recorro sumaFil y escribo resultados
    int cont = 0;
    for (int sumaFila : sumaFil) {
        System.out.println("La suma de los elementos de la fila " + (cont++) + " --> " + sumaFila);
    }

    //recorro sumaCol y escribo resultados
    for (int i = 0; i < sumaCol.length; i++) {
        System.out.println("La suma de los elementos pares de la columna " + i + " --> " + sumaCol[i]);
    }
}
```

26. Se necesita representar en una tabla de 8 x 8 las el color de fondo de un tablero como el de la imagen. El color blanco está representado por un 0 y el negro por un 1. Darle valores a la tabla e imprimirla.



```
import java.util.Scanner;

public class Ej28 {

    static Scanner in = new Scanner(System.in);

    public static void main(String[] args) {
        int f = in.nextInt();
        int c = in.nextInt();
        boolean enBlanco = true; //Indica el color en el que empieza la fila
        boolean blanco = enBlanco; //Toma los valores true or false
        //alternativamente
        int[][] matriz = new int[f][c];

        for (int i = 0; i < f; i++) {
            for (int j = 0; j < c; j++) {
                if (blanco) {
                    matriz[i][j] = 0;
                    blanco = !blanco;
                } else {
                    matriz[i][j] = 1;
                    blanco = !blanco;
                }
            }
            enBlanco = !enBlanco;
            blanco = enBlanco;
        }

        // ESCRIBIMOS LA MATRIZ
        mostrar(matriz);
    }

    public static void mostrar(int[][] matriz){
        int F = matriz.length;
        int C = matriz[0].length;
        for (int i = 0; i < C; i++) {
            System.out.print("-----");
        }
        System.out.println("");
        for (int i = 0; i < F; i++) {
            System.out.print("|");
            System.out.print(" ");
            for (int j = 0; j < C; j++) {
```



```

        if (matriz[i][j] < 10) {
            System.out.print(matriz[i][j]);
            System.out.print(" ");
        }
        else {
            System.out.print(matriz[i][j]);
            System.out.print(" ");
        }
        System.out.print("|");
        if (j != C-1) {
            System.out.print(" ");
        }
    }
    System.out.println("");
}
for (int i = 0; i < C; i++) {
    System.out.print("----");
}
System.out.println("");
}
}

```

27. Hacer un programa que genere e imprima un cuadrado latino de orden 10 (matriz 10x10). Un cuadrado latino de orden n es aquel en el que la primera fila tiene los n<sup>º</sup> naturales del 1 a n, y cada una de las filas restantes de la matriz es igual a su fila superior desplazada una posición hacia la derecha, entrando por la izquierda el número que se pierde en el desplazamiento. Por ejemplo, un cuadrado latino de orden 5 es:

1	2	3	4	5
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1

```

public class CuadradoLatino {

    public static void main(String[] args) {
        int cuadrado[][] = new int[10][10];
        int cont = 1;

        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 10; j++) {
                if (j == 9) { //ultimo elemento de una fila
                    cuadrado[i][j] = cont;
                    cont--;
                } else {
                    cuadrado[i][j] = cont;
                }
                if (cont == 10) { //inicializo cont para la siguiente
fila
                    cont = 0;
                }
                cont++;
            }
        }
    }
}

```



```

    }

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            System.out.print(" " + cuadrado[i][j]);
        }
        System.out.println();
    }
}
}

```

28. Se define el punto de silla de una matriz como un elemento de la matriz que es el más pequeño de su fila y el más grande de su columna. Hacer un programa en el que se le den valores a una matriz en la declaración y que calcule si tiene o no punto de silla.

```

import java.util.ArrayList;
import java.util.Scanner;

public class Ej30 {

    static Scanner in = new Scanner(System.in);

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

        boolean puntosilla = true;
        int menor;
        int mayor;
        int pos = 0;

        ArrayList<Integer> coordf = new ArrayList<Integer>();
        ArrayList<Integer> coordc = new ArrayList<Integer>();

        for (int i = 0; i < f; i++) {
            //Localizo el menor de la fila
            pos = 0;
            menor = matriz[i][pos];
            for (int j = 0; j < c; j++) {
                if (matriz[i][j] < menor) {
                    menor = matriz[i][j];
                    pos = j;
                }
            }
            puntosilla = true;
            //Recorro la columna en la que esta el menor
            for (int j = 0; j < f && puntosilla; j++) {
                mayor = matriz[j][pos];
                if (matriz[j][pos] > mayor) {
                    puntosilla = false;
                }
            }
            if (puntosilla) {
                coordf.add(i);
            }
        }
    }
}

```



```

        coorc.add(pos);
    }
}

for (int i = 0; i < c; i++) {
    System.out.print("--");
}
System.out.println("");
for (int i = 0; i < f; i++) {
    for (int j = 0; j < c; j++) {
        System.out.print(matriz[i][j]);
        if (j != c - 1) {
            System.out.print("|");
        }
    }
    System.out.println("");
}
for (int i = 0; i < c; i++) {
    System.out.print("--");
}
System.out.println("");

if (coorf.isEmpty()) {
    System.out.println("No tiene punto de silla");
} else {
    for (int i = 0; i < coorf.size(); i++) {
        System.out.println("Hay un punto de silla en:" + (1 +
coorf.get(i)) + "," + (1 + coorc.get(i)));
    }
}
}
}

```

29. Hacer un programa que esté asignando valores al azar comprendidos entre 1 y 50 a una matriz de orden 8 x 10 hasta que se cumpla que la matriz generada tenga al menos un punto de silla. Cuando se halla generado una matriz que tenga puntos de silla se ha de escribir en pantalla la matriz generada, el punto de silla junto con sus coordenadas y el número de matrices que se generaron hasta que se encontró una con puntos de silla. A continuación se muestra un ejemplo de la salida del programa:

```

39 29  9 11 35 36 10 13 15 26
33  5 16  6  5  9  1 35 30 50
14  4 50  6  8 16 31 15 41 29
39 18 15 21  4 45  5 28 50 28
 3  4 21 28 29 19 23 37 21 39
47 22 14  4 18 21 31 44 14 34
21  2  6 34 38 12 25 13 41 31
36 31 49 46 40 40 47 45 36 37

```

El elemento de valor 31 y situado en la fila 8 y la columna 2 es un punto de silla.

el bucle se ha ejecutado 9477 veces

30. Un cuadrado mágico de orden N es una matriz cuadrada de dimensiones NxN que contiene a los números naturales de 1 a N<sup>2</sup>, tal que la suma de cualquier columna, fila o diagonal es siempre la misma. El algoritmo para construir el cuadrado mágico es:



- 1.- El número 1 se coloca en la casilla central de la primera fila.
- 2.- Cada número siguiente se coloca en la casilla correspondiente a la fila anterior y columna posterior.
- 3.- Si el número es el siguiente a un múltiplo de N no se aplica la regla 2, sino que se coloca en la casilla de la fila posterior en la misma columna.
- 4.- La fila anterior a la primera es la última.
- 5.- La columna posterior a la última es la primera.

Hacer un programa que pida por teclado un número impar N comprendido entre 3 y 49, y que a continuación genere e imprima un cuadrado mágico de orden N.

```
package ud3ej32;
```

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

```
public class Ud3Ej32 {
```

```
    public static void main(String[] args) {  
        Scanner leo = new Scanner(System.in);  
        int[][] matriz;  
        int n;  
        int f = 0;  
        int c;
```

```
        System.out.println("Introduzca las dimensiones de la tabla:");  
        n = leo.nextInt();  
        c = n / 2;  
        matriz = new int[n][n];  
        for (int i = 1; i <= n * n; i++) {  
            matriz[f][c] = i;  
            if (i % n != 0) { //coloco el numero posterior a un multiplo de n  
                f--;  
                c++;  
                if (f < 0) {  
                    f = n - 1;  
                }  
                if (c >= n) {  
                    c = 0;  
                }  
            } else {  
                f++;  
                if (f >= n) {  
                    f = 0;  
                }  
            }  
        }  
    }
```

```
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n; j++) {  
                if (matriz[i][j] < 10) {  
                    System.out.print("0" + matriz[i][j] + " ");  
                } else {
```



```

        System.out.print(matriz[i][j] + " ");
    }
}
System.out.println("");
}

}
}

```

31. Hacer un programa en java que nos pida la fila y columna donde ir colocando los números del 1 al 9 de tal manera que sea un cuadrado mágico de orden 3. Antes de pedir la fila y la columna tiene que visualizar que números tiene el cuadrado (0 al principio). Una vez introducidas las coordenadas de los nueve números tiene que indicar si es o no un cuadrado mágico.

### Ejemplo de un array de 3 dimensiones:

32. Una cadena de supermercados necesita almacenar información sobre sus ventas. La cadena tiene 10 tiendas, estando formada la línea de cajas por 5. Es necesario mantener información de cuánto vendió cada una de las cajas de cada una de las tiendas en cada uno de los 12 meses del año. Hacer un programa que genere todos estos valores con datos al azar comprendidos entre 6000 y 12000 y que los almacene en una estructura de datos adecuada (sería un array de 3 dimensiones). A continuación escribir un informe de ventas por caja con el siguiente formato:

Tienda	caja	mes	
1	1	Enero	XX
1	1	Febrero	XX
.....			
1	1	Diciembre	XX
TOTAL .....			XXX

Tienda	caja	mes	
1	2	Enero	XX
1	2	Febrero	XX
.....			
1	2	Diciembre	XX
TOTAL .....			XXX

Tienda	caja	mes	
10	5	Enero	XX
10	5	Febrero	XX
.....			
10	5	Diciembre	XX
TOTAL .....			XXX

Declarar las constantes TIENDAS y CAJAS que representen el número de tiendas y de cajas de cada tienda con



las que vamos a trabajar. Para hacer este programa poner 2 tiendas y 3 cajas.

```
package ud3ej36;

import java.util.Random;

public class Ud3ej36 {

    public static void main(String[] args) {
        final int TIENDA = 10;
        final int CAJA = 5;
        int[][][] ventas = new int[TIENDA][CAJA][12];
        String[] mes = {"Enero ", "Febrero ", "Marzo ", "Abril ", "Mayo ", "Junio ", "Julio ",
"Agosto ", "Septiembre", "Octubre ", "Noviembre ", "Diciembre "};
        int sumaCaja;
        int sumaTienda;
        Random azar = new Random();

        for (int i = 0; i < TIENDA; i++) {
            for (int j = 0; j < CAJA; j++) {
                for (int k = 0; k < 12; k++) {
                    ventas[i][j][k] = azar.nextInt(6001) + 6000;
                }
            }
        }

        for (int i = 0; i < TIENDA; i++) {
            sumaTienda = 0;
            System.out.println("Tienda " + i + " caja " + " mes ");
            for (int j = 0; j < CAJA; j++) {
                sumaCaja = 0;
                for (int k = 0; k < 12; k++) {
                    System.out.println((i + 1) + " " + (j + 1) + " " + mes[k] + " " + ventas[i][j][k]);
                    sumaCaja = sumaCaja + ventas[i][j][k];
                }
                System.out.println("TOTAL CAJA....." + sumaCaja);
                sumaTienda += sumaCaja;
            }
            System.out.println("TOTAL TIENDA....." + sumaTienda);
        }
    }
}
```

33. Hacer un programa en el que un caballo de ajedrez recorra las 64 casillas de un tablero de ajedrez a partir de unas coordenadas iniciales leídas por teclado. Se utilizará el algoritmo de WARNSDORFF que da la regla de que el caballo debe de ir siempre a la casilla en la que sean menores las posibilidades de moverse a casillas aún no visitadas.

[http://en.wikipedia.org/wiki/Knight%27s\\_tour](http://en.wikipedia.org/wiki/Knight%27s_tour)





1	4	59	20	41	6	43	22
34	19	2	5	60	21	40	7
3	64	35	58	37	42	23	44
18	33	48	61	46	57	8	39
49	14	63	36	53	38	45	24
32	17	52	47	62	27	56	9
13	50	15	30	11	54	25	28
16	31	12	51	26	29	10	55

**34. HACER UN PROGRAMA QUE CALCULE EL PRODUCTO DE DOS NÚMEROS LEÍDOS POR TECLADO UTILIZANDO EL ALGORITMO ÁRABE**

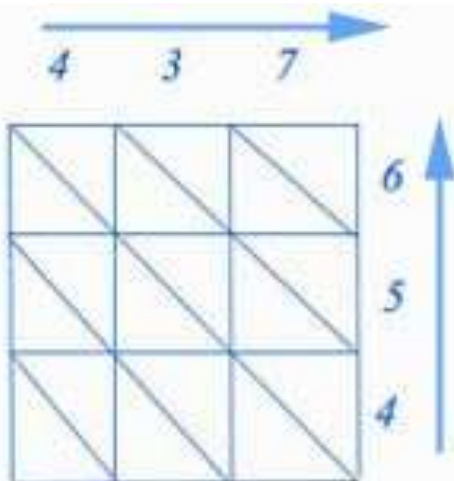
Luca Pacioli recoge en su Summa de arithmetica, geometria, proporzioni et proporzionalita , editado en Venecia en 1.494; un procedimiento árabe del siglo XIII para obtener el producto de dos números de cualquier cantidad de cifras, que se aparta del habitual que conocemos, si bien es completamente equivalente a éste.

El ejemplo de la imagen multiplica 437X 456. Como ambos números tienen tres cifras, se parte de un cuadrado subdividido en 3x3 celdas, y se subdivide cada celda como se indica en la figura de la izquierda. Uno de los números a multiplicar se escribe sobre el segmento superior escrito en sentido habitual, y en otro en el lateral, de abajo arriba .

En cada celda se escribe el producto de los dígitos correspondientes, colocando las unidades en la subcasilla superior, y las decenas, si las hubiere, en la inferior.

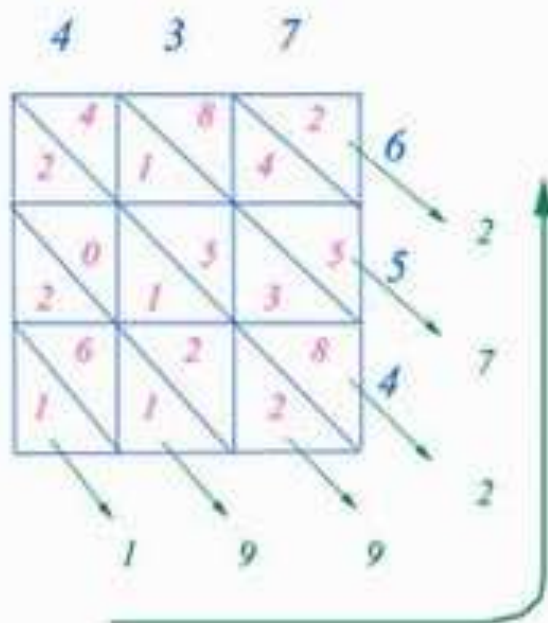
Se suma diagonalmente, acarreando a la diagonal siguiente las llevadas, si las hubiere.

El resultado aparece en las sumas de las diagonales, en el sentido que indica la flecha verde.



$$437 \times 456 = 199.272$$

*Tio Petros*





# USO DE ALGUNAS COLECCIONES

## ARRAYLIST

<http://ocw.ehu.es/enseanzas-tecnicas/introduccion-a-la-programacion-en-java/mod-ii/notas/tema-6>

<http://tutorials.jenkov.com/java-collections/index.html>

Programa que crea un ArrayList de Objetos Alumno.

```
import java.util.ArrayList;
import java.util.List;

public class PruebaArrayList {
    public static void main(String[] args) {
        List<Alumno> alumnos = new ArrayList<Alumno>();

        //Añadimos valores al ArrayList, método add
        Alumno a = new Alumno("Manuel", "López");
        alumnos.add(a);
        Alumno b = new Alumno("Luisa", "Menéndez");
        alumnos.add(b);

        Alumno c = new Alumno("Miguel", "Martínez");
        alumnos.add(c);

        //recorremos el ArrayList
        for (Alumno alumno : alumnos) {
            System.out.println(alumno.getNombre());
            System.out.println(alumno.getApellidos());
        }

        //obtenemos un elemento de una posición determinada del ArrayList
        a = alumnos.get(1);
        System.out.println("alumno en la posición 1");
        System.out.println(a.getNombre());
        System.out.println(a.getApellidos());

        alumnos.remove(1);
        System.out.println("listado después de borrar el alumno que está en la
posición 1 ");
        for (Alumno alumno : alumnos) {
            System.out.println(alumno.getNombre());
            System.out.println(alumno.getApellidos());
        }
    }
}

class Alumno {
    private String nombre;
    private String apellidos;

    public Alumno(String nombre, String apellidos) {
        this.nombre = nombre;
        this.apellidos = apellidos;
    }
}
```



```

    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}

```

Programa que crea un ArrayList de datos Integer, le da valores y los escribe en pantalla.

```

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class PruebaArrayList {
    public static void main(String[] args) {
        ArrayList <Integer> numeros = new ArrayList<>();
        Random aleatorio = new Random();

        for (int i = 0; i < 20; i++) {
            numeros.add(aleatorio.nextInt(50));
        }

        System.out.println("Escribimos los números generados ");

        for (Integer numero : numeros) {
            System.out.println(numero);
        }
    }
}

```

35. Para desarrollar el juego de “encuentra parejas”, es necesario asignar a una tabla de dimensiones 4 x 4 parejas de números comprendidos entre 1 y 8 (dos 1s, dos 2s, dos 3s...) distribuidas de manera aleatoria en la tabla. Hacer un programa en el que se declare una tabla de 4 filas y 4 columnas y se le asignen las parejas de números.

Ejemplo de valores distribuidos aleatoriamente:



4	8	5	4
8	1	7	6
6	3	2	5
2	7	3	1

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;

public class Parejas {

    public static void main(String[] args) {
        int[][] tablaParejas = new int[4][4];
        int a = 0;
        Integer v[] = {1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8};

        ArrayList<Integer> listaNumeros = new ArrayList<>(Arrays.asList(v));

        Collections.shuffle(listaNumeros);

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                tablaParejas[i][j] = listaNumeros.get(a++);
            }
        }

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.print(tablaParejas[i][j]+" ");
            }
            System.out.println("");
        }
    }
}
```

36. Hacer un programa en el que se asignen a un ArrayList 20 valores generados al azar comprendidos entre 20 y 60 ambos incluidos, y que calcule y escriba el elemento menor y mayor.

```
import java.util.ArrayList;
import java.util.Random;

public class Ud3Ej39 {

    public static void main(String[] args) {

        Random azar = new Random();
        ArrayList<Integer> numeros = new ArrayList<>();
```



```
int mayor, menor;

for (int i = 0; i < 20; i++) {
    numeros.add(azar.nextInt(41) + 20);
}

mayor = numeros.get(0);
menor = numeros.get(0);
for (Integer numero : numeros) {
    if (numero > mayor) {
        mayor = numero;
    }
    if (numero < menor) {
        menor = numero;
    }
}

System.out.println(numeros);
System.out.println("mayor " + mayor);
System.out.println("menor " + menor);
}
}
```

#### Otra solución:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class Ud3ej39 {

    public static void main(String[] args) {
        ArrayList<Integer> numeros = new ArrayList<>();
        Random azar = new Random();
        int menor = 0;
        int mayor = 0;

        for (int i = 0; i < 20; i++) {
            numeros.add(azar.nextInt(41) + 20);
        }

        for (int i = 0; i < numeros.size(); i++) {
            if (i == 0) {
                menor = numeros.get(i);
                mayor = numeros.get(i);
            }
            if (numeros.get(i) < menor) {
                menor = numeros.get(i);
            }
            if (numeros.get(i) > mayor) {
```



```

        mayor = numeros.get(i);
    }
}
System.out.println(numeros);
System.out.println("El número menor es: " + menor);
System.out.println("El número mayor es: " + mayor);

}
}

```

37. Diseñar una aplicación en la que se lean por teclado 5 números enteros, almacenándolos en un ArrayList de nombre lista. Visualizar el contenido de lista. Después se debe pedir otro número y una posición en la que hay que insertarlo en lista. Insertar el número, siempre que la posición leída sea menor al número de elementos de lista. Visualizar de Nuevo lista para comprobar que la inserción se hizo bien.

```

package ud3ej40;

import java.util.ArrayList;
import java.util.Scanner;

public class Ud3Ej40 {

    public static void main(String[] args) {
        Scanner leo = new Scanner(System.in);
        ArrayList<Integer> numeros = new ArrayList<>();
        int numero, posicion;

        System.out.println("Introduce 5 valores");
        for (int i = 0; i < 5; i++) {
            numeros.add(leo.nextInt());
        }
        System.out.println(numeros);

        System.out.println("Introduce la posicion en la que quieras introducir un numero: ");
        posicion = leo.nextInt();
        System.out.println("Introduce el numero que quieras introducir:");
        numero = leo.nextInt();
        if (posicion < numeros.size()) {
            numeros.add(posicion, numero);
        } else {
            System.out.println("posicion fuera del rango del arrayList");
        }
        System.out.println(numeros);
    }
}

```

38. La **Lotería Primitiva** es un juego de azar regulado por la Organización Nacional de Loterías y Apuestas del Estado (LAE) que consiste en elegir 6 números diferentes entre 1 y 49. Para acertar la Combinación Ganadora en el sorteo de **Lotería Primitiva** correspondiente, se extraen 6 bolas de un bombo. Además se



extrae una bola extra como número complementario, y otra bola de un bombo aparte, entre el 0 y el 9, que hace de número de «reintegro».

Hacer un programa que genere 5 combinaciones de la lotería primitiva. Tiene que visualizar:

Combinación 1: 4, 2, 7,23,45,34    Complementario: 13    Reintegro: 5

Combinación 2: .....

**NOTA:** Guardar los numeros en un ArrayList y "menearlos" mediante el método shuffle.

```
/*
    Shuffle elements of Java ArrayList example
    This java example shows how to shuffle elements of Java ArrayList object using
    shuffle method of Collections class.
*/

import java.util.ArrayList;
import java.util.Collections;

public class ShuffleElementsOfArrayListExample {

    public static void main(String[] args) {

        //create an ArrayList object
        ArrayList arrayList = new ArrayList();

        //Add elements to Arraylist
        arrayList.add("A");
        arrayList.add("B");
        arrayList.add("C");
        arrayList.add("D");
        arrayList.add("E");

        System.out.println("Before shuffling, ArrayList contains : " + arrayList);

        /*
            To shuffle elements of Java ArrayList use,
            static void shuffle(List list) method of Collections class.
        */

        Collections.shuffle(arrayList);

        System.out.println("After shuffling, ArrayList contains : " + arrayList);

    }
}

package ud3ej41;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Ud3Ej41 {

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





```
ArrayList<Integer> numeros = new ArrayList<>();
ArrayList<Integer> reintegro = new ArrayList<>();

for (int i = 1; i <= 49; i++) {
    numeros.add(i);
}
for (int i = 0; i <= 9; i++) {
    reintegro.add(i);
}

for (int i = 0; i < 5; i++) {
    Collections.shuffle(numeros);
    Collections.shuffle(reintegro);

    System.out.print("Combinacion " + (i + 1) + ": ");

    for (int j = 0; j < 6; j++) {
        System.out.print(numeros.get(j) + " ");
    }
    System.out.print(" Complementario: " + numeros.get(6));
    //no hace falta un ArrayList para el reintegro,
    //podría generar un número al azar entre 0 9
    System.out.print(" Reintegro: " + reintegro.get(0));
    System.out.println("");
}
}
}
```

39. Hacer un programa en el que se declaren 2 vectores de nombres: nombre, apellido1, que tengan cada uno de ellos 10 nombres y 10 apellidos, dándole valores al declararlos sin repeticiones.

Escribir a continuación el nombre y el apellido de 8 personas, sin que aparezcan nombres y apellidos repetidos.

AYUDA: Copiar en dos ArrayList el contenido de los vectores nombres y apellido1. Desordenar su contenido y coger los 8 primeros elementos.

Se pueden asignar los vectores a listas mediante el método asList() de la clase Arrays.



## CONJUNTOS

En programación es relativamente frecuente encontrarse en la necesidad de almacenar en la memoria del computador conjuntos de elementos. Para ello es posible optar entre diferentes estructuras de datos, que son una opción razonable cuando se van a usar, sobre todo, para saber si un elemento forma parte del conjunto. La elección de la palabra conjunto no es casual. Un aspecto a tener en cuenta de este tipo de colecciones es que, por definición, sus elementos no se repiten y están ordenados.

Por lo que se refiere a Java, y dicho informalmente, un objeto de la clase `TreeSet<Integer>` almacena un conjunto de números enteros. Aunque de recién creado, un `TreeSet<Integer>` está vacío y no contiene ningún elemento, usando los métodos de esa clase podemos añadirle tantos elementos como deseemos. Por ejemplo, con las instrucciones siguientes:

```
TreeSet<Integer> a = new TreeSet<Integer>();
```

```
a.add(1); //añade 1 al conjunto a
```

```
a.add(2); //añade 2 al conjunto a
```

```
a.add(2); //no añade este elemento por qué ya está
```

40. Hacer un programa en el que se generen de manera aleatoria 8 valores comprendidos entre 1 y 10 sin que haya números repetidos. Almacenar los números en un conjunto (`TreeSet`).

41. Dado un `String`, escribir en pantalla los caracteres por los que está formado, escribiendo una vez los caracteres repetidos.

```
String nombre = "dabalearrozalazorraelabad";
```

```
Salida: [a, b, d, e, l, o, r, z]
```

```
package ud3ej45;
```

```
import java.util.TreeSet;
```

```
public class Ud3ej45 {
```

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

```
        String frase = "dabalearrozalazorraelabad";
```

```
        TreeSet<Character> conjunto = new TreeSet<>();
```

```
        for (int i = 0; i < frase.length(); i++) {
```

```
            conjunto.add(frase.charAt(i));
```

```
        }
```

```
        System.out.println(conjunto);
```

```
    }
```

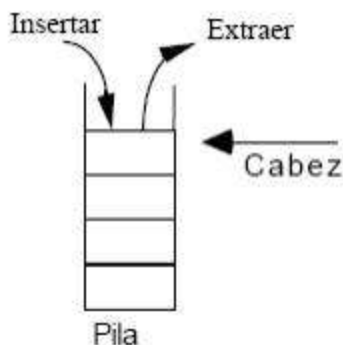
```
}
```

## PILAS

Una pila es una estructura de datos en la que la inserción y la extracción de elementos se realiza sólo por un extremo que se denomina cabeza. como consecuencia, los elementos de una pila serán eliminados en orden inverso al que se insertaron. es decir, el último elemento que se metió en la pila será el primero en salir de ella.

Debido al orden en que se insertan eliminan los elementos en una pila, también se le conoce como estructura lifo (last in, first out: último en entrar, primero en salir).y

ejemplo grafico:



en resumen: una pila es una estructura de datos homogénea (elementos del mismo tipo), secuencial y de tamaño variable. sólo es posible un modo de acceso a esta estructura: a través de la cabeza de la pila.

```
import java.util.Stack;

public class PruebaArrayList {
    public static void main(String[] args) {
        Stack<Integer> pila = new Stack<Integer>();

        pila.add(1);
        pila.add(5);
        pila.add(9);

        while (!pila.isEmpty()) {
            System.out.println(pila.pop());
        }
    }
}
```

42. Hacer un programa que dada una cadena de caracteres la visualice al revés. Utilizar una pila.  
package pila;



```
import java.util.Stack;

public class Pila {

    public static void main(String[] args) {
        String frase = "holaquetalestas";
        Stack<Character> pila = new Stack<>();

        for (int i = 0; i < frase.length(); i++) {
            pila.add(frase.charAt(i));
        }
        while (!pila.isEmpty()) {
            System.out.println(pila.pop());
        }
    }
}
```