

NOMBRE .....

Escribir los programas en JAVA que se indican a continuación, **ES RESPONSABILIDAD TUYA ESCRIBIR CORRECTAMENTE LA SOLUCIÓN**. Si no es legible o no cumple lo expuesto, el examen no podrá ser corregido y la nota será 1.

**Ejercicio 1.- (2 puntos)**

Implementa un algoritmo que sea capaz de dibujar el símbolo '>' utilizando asteriscos. El usuario nos introducirá un número entero mayor o igual a 2. En base a ese número, se debe pintar el símbolo por consola. Ejemplos:

3	4	5	6
<pre> **   **     **   ** ** </pre>	<pre> **   **     **       **     **   ** ** </pre>	<pre> **   **     **       **         **       **     **   ** ** </pre>	<pre> **   **     **       **         **           **       **     **   ** ** </pre>

**Ejercicio 2.- (2 puntos)**

A la vista del siguiente código, dibuja lo que aparecerá en la consola.

```

public class Mascota {

    private String nombre;
    private int id;
    public static String fecha = "15/11/2022";

    public Mascota(String nombre, int id) {
        this.nombre = nombre;
        this.id = id;
    }

    public static String getFecha() {
        return fecha;
    }

    public static void setFecha(String fecha) {
        Mascota.fecha = fecha;
    }

    @Override
    public String toString() {
        return "Mascota (" + id + " :: " + fecha + "): " + nombre;
    }
}

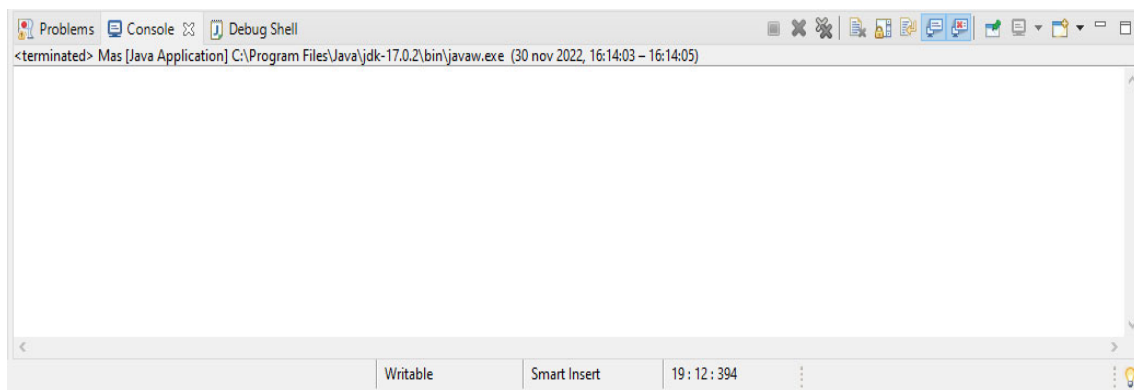
```

NOMBRE .....

```
}

public class Principal {
    public static void main(String[] args) {
        Mascota m1 = new Mascota("Tornado",1);
        Mascota m2 = new Mascota("Calcetin",3);
        Mascota m3 = new Mascota("Golum",2);

        System.out.println(m1.toString());
        m2.setFecha("16/11/2022");
        System.out.println(m2.toString());
        System.out.println(m3.toString());
    }
}
```



### Ejercicio 3.- (2 puntos)

A la vista del siguiente código, dibuja lo que aparecerá en la consola.

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

    String[] cosas= {"moto","coche","arbol","casa"};

    for (int i = 0; i < cosas.length; i++) {
        for(int j=0; j<i;j++) {
            System.out.print(cosas[i].charAt(j));

            if(j==i) {
                System.out.println("-->");
            }
        }
    }
}
```

NOMBRE .....

**Ejercicio 4.- (4 puntos)**

Una empresa de venta de artículos online necesita gestionar los pedidos que le hacen sus clientes desde su sitio web. Para ello tenemos que desarrollar la clase **Pedido**, que tendrá al menos las siguientes propiedades privadas (deben llamarse exactamente así):

**codPedido** (int, el código del pedido que será asignado por nuestro sistema. Habrá que darle un valor único a cada pedido que se irá generando automáticamente incrementándose en 1 para cada nuevo pedido y empezando en 1)

**codigosArticulo** (array de int, contendrá todos los códigos de artículo de los que consta el pedido. Un pedido podrá constar de un máximo de 10 artículos)

**numArticulos** (int, el número de artículos de los que consta el pedido, que debe mantenerse actualizado en todo momento)

**nombrePersona** (String, el nombre de la persona a la que hay que enviar el pedido)

**direccionEnvio** (String, la dirección a la que debe enviarse el pedido)

Se pide crear la clase **Pedido**, respetando las siguientes indicaciones: **(0.5p cada apartado)**

- Declara todas las propiedades necesarias para lo que se solicita.
- Crea un método constructor que cree el pedido a partir del **nombre** y la **dirección** del cliente. Deberá inicializar todo lo necesario y asignarle código al pedido según lo indicado.
- Crea el método **private int buscaPosicionLibre()**, que debe buscar la primera posición libre en el array **codigosArticulo** y devolverla. Se considera libre aquella posición que tiene el valor 0. Devolverá -1 si no hay posiciones libres.
- Crea el método **public boolean incluirArticulo(int codigoArticulo)**, que incluirá el código de artículo pasado como parámetro en el array **codigosArticulo** del pedido, haciendo uso del método desarrollado en el apartado anterior, y actualizará la información necesaria. Si se ha podido incluir devolverá **true**, en caso contrario devolverá **false** (si supondría superar los 10 artículos que se pueden incluir en el pedido).

NOMBRE .....

e) Crea el método **private int buscaPosicionArticulo(int codigoArticulo)**, que debe buscar y devolver la primera posición del array **codigosArticulo** que tiene el código indicado como parámetro. Devolverá -1 si no encuentra el artículo.

f) Crea el método **public boolean eliminarArticulo(int codigoArticulo)**, que eliminará del array **codigosArticulo** el código pasado como parámetro, y actualizará la información necesaria. Si se ha podido eliminar devolverá **true**, en caso contrario devolverá **false**.

g) Crea el método necesario para que desde el programa principal salga por pantalla toda la información necesaria al ejecutar la instrucción **System.out.println(p)**, siendo p una referencia a un objeto **Pedido**.

h) Crea una clase con método **main()**, que cree un array de 5 pedidos, cree **dos** pedidos y los meta en el array, incluya **varios** artículos en cada uno de ellos, y muestre toda la información de los pedidos de dicho array.

**Solución Ejercicio 1:**

**NOMBRE .....**

**Solución Ejercicio 4:**

**NOMBRE .....**

NOMBRE .....

Escribir los programas en JAVA que se indican a continuación:

Créate un único proyecto con el nombre **ApellidosEval1**, con tus apellidos obviamente.

Las clases que contengan el método main() para cada ejercicio deben llamarse **ApellidosExNum.java** donde Num es el número del ejercicio (por ejemplo LopezPerezEx1.java).

Al finalizar el examen, genera un fichero comprimido que contenga todo el **proyecto** (no el workspace) y lo envías por correo electrónico a la dirección y [juan.jose.gimeno@cpilosenlases.com](mailto:juan.jose.gimeno@cpilosenlases.com) y [daniel.val@cpilosenlases.com](mailto:daniel.val@cpilosenlases.com)

Usa la clase **Leer** suministrada y guárdala en el mismo proyecto. Si tienes tu propia clase Leer, cópiala en el proyecto de tu examen y envíala al final dentro del fichero zip.

**ES RESPONSABILIDAD TUYA ENVIAR CORRECTAMENTE LA SOLUCIÓN. Si no es legible o no cumple lo anteriormente expuesto, el examen no podrá ser corregido y la nota será 1.**

### Ejercicio 1.- (2.5 puntos)

Escribe una clase llamada Binario con un método main que genere dos arrays de 8 booleanos aleatorios llamados b1 y b2.

- Implementa un método para imprimir el array donde true será representado por un 1 y false por un 0.
- Implementa un método que compare dos arrays elemento a elemento y que genere un nuevo array. En el nuevo array, el valor vendrá determinado por la operación OR con el valor que contiene b1 y b2. Comparando posición a posición, si encontramos un 1 en alguno de los dos vectores, el resultado en el nuevo array será 1, y 0 en caso contrario.

B1	1	0	1	1	0	1	1	1
B2	1	1	1	0	0	1	0	1
resultado	1	1	1	1	0	1	1	1

La salida del programa debe ser algo similar a esto

1 0 1 1 0 1 1 1 comparado con 1 1 1 0 0 1 0 1 es 1 1 1 1 0 1 1 1

Rúbrica:

Implementar la clase principal con los vectores aleatorios. **0.5p**

Método para imprimir vectores. **1p**





NOMBRE .....

Cada vez que pulsemos la tecla intro deberá extraerse una bola del sorteo y su correspondiente premio y mostrarse en pantalla de la siguiente manera.

Extracción número 1: El número xxxxx obtiene un premio de xxxxxx;

Extracción número 2: El número xxxxx obtiene un premio de xxxxxx;

Extracción número 3: El número xxxxx obtiene un premio de xxxxxx;

Y así sucesivamente hasta acabar el sorteo.....

## 2- Menú con opciones: (2p)

Una vez concluido muestra un menú con las opciones Mostrar resultados, Consultar número y Salir.

Mostrar resultados: mostrará el resultado del sorteo con los premios.

Consultar número: nos permitirá consultar si nuestro número está premiado. Se mostrará un mensaje de salida indicando si nuestro número ha sido premiado o no y en caso de ser premiado el valor del premio. Si el número no es válido deberá mostrar el mensaje **número no valido**.

### Ejercicio 4.- (2.5 puntos)

Implementa un algoritmo que genere etiquetas. Se necesita definir una clase **Datos** donde contener los datos de la persona, **nombre**, **apellido1**, **apellido2** y **usuario**. Elige bien el tipo de dato que necesitas en cada caso. El atributo **usuario**, se genera en el constructor a partir de los tres primeros caracteres del **nombre**, seguido de un punto y los tres primeros caracteres del **apellido1**.

Se necesita una clase **Dirección** con los datos de la dirección, como son **calle**, **número**, el **código postal**, **ciudad** y **población**.

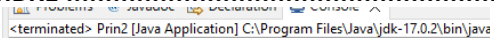
Por último, implementa la clase **Persona** compuesta por un atributo **Datos** y otro atributo **Dirección**.

Para probar las clases que has desarrollado, construye una clase **ApellidosEx4.java**, con un método main donde se instancian dos objetos Persona y se muestran las etiquetas correspondientes.

Añade a la clase principal un método estático **getUser(Persona p)**, donde nos devuelva el nombre de usuario de un objeto Persona que se le pasa como parámetro.

Ejemplo:

NOMBRE .....



```
<terminated> Prin2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\java
```

Dan.Val

Daniel Val Pérez

calle el olivo, 25

50004 Zaragoza

Zaragoza

Jua.Gim

Juan Jose Gimeno del Monte

avenida los monegros, 42

50008 Zaragoza

Zaragoza

- a) Implementa las clases (0.5p)
- b) Implementa la clase principal con un algoritmo coherente (0.5p)
- c) Implementa el método ***getUser(Persona p)*** (0.5p)
- d) Genera las etiquetas con el formato del ejemplo (1p)