

# PRUEBA 1

## BLOQUE 1

### EJERCICIO 1

En el concurso de Eurovisión participan 43 países. De los 26 finalistas que participan en la gala final, solo seis llegan a ella de forma directa. Se trata de los países conocidos como el *Big Five* (Alemania, Francia, Italia, España y Reino Unido) además del organizador, en este caso Portugal. Las otras veinte naciones logran su pase a la gran gala a través de dos semifinales.

En la final, el **voto del público** desde casa (50%) y el de los **jurados profesionales** (50%) son los que determinan quién gana el concurso.

#### Votaciones en la final

En la final votan los **43 países participantes** en la edición, hayan llegado o no a la gala.

En la primera parte de las votaciones, los portavoces de cada país anuncian los puntos de los jueces, que **van del uno al 8, después el 10 y finalmente los 12 puntos** (no hay nueve ni 11). Tras completar la ronda de los países que votan, se dará paso a la puntuación del público. La suma de ambos dará con el ganador de **Eurovisión 2018**.

Son 43 países votantes, aunque solo 26 participan en **la** final. No se puede votar **a la canción presentada por el propio país**.

**Los clasificados para la final son:** Serbia, Moldavia, Hungría, Ucrania, Suecia, Australia, Noruega, Dinamarca, Eslovenia, Holanda, Albania, República Checa, Lituania, Israel, Estonia, Bulgaria, Austria, Finlandia, Irlanda y Chipre. A los que hay que sumar el *Big five* con Alemania, Italia, Reino Unido, Francia, y España. Portugal no ha debido pasar ninguna clasificación por ser el vigente ganador de *Eurovisión*.

En este ejercicio hay que escribir un programa en **Java** para simular la votación de los jurados de cada país. No se va a tener en cuenta la votación del público.

Para cada uno de los 43 países que votan hay que generar aleatoriamente los 10 países a los que darán el juego de puntos, teniendo en cuenta que no se pueden votar a sí mismos ni votar dos veces al mismo país.

Al finalizar la votación hay que mostrar:

- Listado por orden alfabético de los países finalistas en el que se mostrará el nombre del país y la puntuación obtenida.
- Nombre y puntuaciones obtenidas por los países que han obtenido la puntuación más alta y la más baja.

Posibles aclaraciones sobre la corrección

- La carga de datos (de los 43 países) no hace falta escribirla explícitamente se puede especificar la forma en la que lo haría
- **El** código ha de seguir una lógica bien estructurada y correctamente comentada, siguiendo las buenas prácticas de programación orientada a objetos
- La sintaxis ha de ser correcta. En el caso de no recordar un nombre de un método se podrá valorar aunque no coincida "exactamente".

***Nota:** Basado en la OPE de Informática de Aragón 2018.*

**Solución: adjunta en un archivo comprimido.**

## EJERCICIO 2:

Responde a las siguientes preguntas (al final del ejercicio encontrarás una tabla para escribir las respuestas):

### 1. Dado el siguiente código muestra su salida por pantalla

```
#define N 5

int funcion (void)
{
    static int valor = 0;
    valor++;
    return valor;
}

int main()
{
    int vector[N], i, suma = 0;
    for (i = 0; i < N; i++)
    {
        vector[i] = funcion();
        suma += vector[i];
    }
    printf ("%d", suma);
    return 0;
}
```

### 2. Dado el siguiente código muestra su salida por pantalla

```
int funcion(int a, int b)
{
    int result = a;
    if (a < b)
        return result;
    result = funcion (a - b, b);
    return result;
}

int main ()
{
    printf ("%d", funcion (5, 2));
}
```

3. El siguiente método permite calcular una potencia de manera recursiva. Completa la línea que falta:

```
public int potenciaConRecursion (int m, int n) {  
    if (n==0) {  
        return 1;  
    } else {  
        *****  
    }  
}
```

4. El siguiente método pretende mostrar los valores de una pila en pantalla:

```
static void escribirPila (Pila pila) throws PilaVacía {  
    int elem;  
    elem = pila.desapilar ();  
    escribirPila (pila);  
    System.out.println (elem);  
    pila.apilar (elem);  
}
```

Dada una pila de enteros con los siguientes valores (desde la cima hasta el fondo):  
5 7 34 2 1. Indique qué aparecerá en la pantalla.

5. El siguiente algoritmo que pretende eliminar de una lista calificada ordenada el elemento, si existe, cuya clave coincida con el dato que se pasa como argumento:

```
static void eliminar (NodoLista lista, int dato) {  
    if (lista != null)  
        if (lista.clave < dato) eliminar (lista.sig, dato);  
        else if (lista.clave == dato) lista = lista.sig;  
        else System.out.println ("ERROR: La clave no existe");  
    else System.out.println ("ERROR: La clave no existe");  
}
```

¿Cuál de las siguientes afirmaciones es correcta?

- A. Funciona correctamente salvo al eliminar el primer elemento de la lista.
- B. Funciona correctamente en cualquier caso.
- C. No funciona en ningún caso.

**6. El siguiente método pretende mostrar los valores de una pila en pantalla:**

```
static void escribirPila (Pila pila) throws PilaVacía {  
    int elem;  
    elem = pila.desapilar ();  
    escribirPila (pila);  
    System.out.println (elem);  
    pila.apilar (elem);  
}
```

**Dada una pila de enteros con los siguientes valores (desde la cima hasta el fondo): 5, 7, 34, 2, 1; Indique qué aparecerá en la pantalla.**

- A. 1, 2, 34, 7, 5,
- B. 5, 7, 34, 2, 1,
- C. Se producirá un error.

**7. Se proponen los siguientes tipos de datos para implementar una cola mediante una lista enlazada por medio de punteros:**

*Opción cola1.*

```
class NodoCola { int dato;  
    NodoCola siguiente; }  
  
public class tad_cola implements Cola { private NodoCola inicio; public tad_cola () {  
    inicio = null; }  
}
```

*Opción cola2.*

```
class NodoCola { int dato;  
    NodoCola siguiente; }  
  
public class tad_cola implements Cola { private NodoCola principio; private NodoCola fin; public tad_cola () {  
    principio = null;  
    fin = null; }  
}
```

**¿Cuál de las siguientes afirmaciones es correcta?**

- A. La opción cola2 no permite definir una cola.
- B. Ambas opciones son válidas, pero para cola1 hay que realizar un tratamiento, iterativo o recursivo, para hacer alguna de las operaciones básicas (encolar/desencolar).
- C. Ambas opciones son válidas, pero cola2 requiere la realización de un tratamiento, iterativo o recursivo, para realizar alguna de las operaciones básicas (encolar/desencolar).

8. **El siguiente algoritmo:**

```
static int contarPila (Pila pila) throws PilaVacía {  
    int elem, resul;  
        if (! pila.pilaVacía ()) {  
            elem = pila.desapilar ();  
            resul = 1 + contarPila (pila);  
        } else resul = 0;  
    return resul;  
}
```

**Selecciona una respuesta.**

- A. Producirá un error de compilación.
- B. Suma correctamente pero pierde el contenido de la pila.
- C. Suma correctamente y conserva el contenido de la pila, pues se ha pasado como argumento por valor.

| Respuestas |  |
|------------|--|
| 1          |  |
| 2          |  |
| 3          |  |
| 4          |  |
| 5          |  |
| 6          |  |
| 7          |  |
| 8          |  |

### EJERCICIO 3

**Dada una clase abstracta Instrumento con un atributo “tipo” y un método abstracto tocar(), y dos clases hijas Violín y Clarinete, escribe el código que permite definir las tres clases y los métodos necesarios (no implementes el código interior de los métodos).**

## EJERCICIO 4

**Dibuja un diagrama de clases donde se represente el siguiente código:**

```
interface Player {
    void play();
    void stop();
    void pause();
    void reverse();
}

interface Recorder extends Player {
    void record();
}

class TapePlayer implements Recorder {
    public void play() { ... }
    public void stop() { ... }
    public void pause() { ... }
    public void reverse() { ... }
    public void record() { ... }
}

class DVDPlayer extends Player {
    public void play() {
        super.play(); // change state
    }
}

class CDPlayer extends Player {
    public void play() {
        super.play(); // change state
    }
}
```