

# Programación – DAM -EL

## Convocatoria Extraordinaria

Observaciones:

- Versión JAVA 10 como máximo
- Ficheros adjuntos: HighScores.sql, Jugador.java, Partida.java, Arcade.java
- Prestar atención a las restricciones de uso de estructuras y tipos de datos indicadas en el enunciado

En este examen se va a evaluar la adquisición de los siguientes resultados de aprendizaje especificados en el Decreto 252/2011, de 12 de Agosto, por el que se establece el currículo de ciclo formativo de grado superior correspondiente al título superior en Desarrollo de Aplicaciones Multiplataforma en la Comunidad Autónoma de Castilla-La Mancha:

RA1: Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

RA2: Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

RA3: Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

RA7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

RA8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

RA9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

# Programación – DAM -EL

## Convocatoria Extraordinaria

Crea una clase llamada Tetris.java que implemente la siguiente interfaz:

```
public interface Arcade {  
    public ArrayList<Integer> lineasTetris(int m[][]);  
    public int[][] tableroToMatriz(String fichero);  
    public int insertaPuntuacion(Jugador a);  
    public int mejorRacha(ArrayList<Partida> listaPartidas);  
    public int mediaPais(String pais);  
}
```

**Ejercicio 1 (2 puntos):** Implementa el método `lineasTetris` que recibe como parámetro una matriz que representa el tablero de un juego de Tetris (de cualquier número de filas y columnas) y retorna como resultado un `ArrayList` con las filas donde se haya producido una “línea”. Cada elemento de la matriz del tablero tendrá un número entero que representa si es está ocupada por un bloque o no. Cada número entero podrá ser:

- 0 – casilla vacía
- 1 – casilla ocupada por un bloque de color rojo
- 2 – casilla ocupada por un bloque de color verde
- 3 – casilla ocupada por un bloque de color azul

En el juego del Tetris, se considera que se ha conseguido una “línea” cuando todas las casillas de la misma fila de la matriz están ocupadas por bloques.

Por ejemplo, dada la matriz A:

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	0	3	0	0
1	3	2	1	2
1	2	2	3	1
0	2	1	0	1
2	3	1	1	2

→ Línea en fila 4!  
→ Línea en fila 5!  
→ Línea en fila 7!

El método debe retornar este `ArrayList`:

4	5	7
---	---	---

# Programación – DAM -EL

## Convocatoria Extraordinaria

**Ejercicio 2: (2 puntos):** Implementa el método `tableroToMatriz` que genera una matriz a partir del fichero que se pasa como parámetro. El fichero tendrá números enteros separados por ; al estilo csv.

Por ejemplo, dada la llamada al método `tableroToMatriz`:

```
int m[][] miMatriz=tableroToMatriz("miTablero.txt");
```

y con el fichero `miTablero.txt` siguiente:

0;0;0;0;0;0;0;0
0;0;0;0;0;0;0;0
1;1;0;0;2;3;0;0
1;1;1;2;2;3;3;3
3;3;3;3;3;2;1;0
0;0;1;1;1;1;1;1

→ Debe devolver la matriz →

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	0	0	2	3	0	0
1	1	1	2	2	3	3	3
3	3	3	3	3	2	1	0
0	0	1	1	1	1	1	1

**Ejercicio 3: (2 puntos):** implementar el método `insertaPuntuacion` que calcula la puntuación más alta de las partidas de un jugador y la inserta en la tabla `HighScores` del fichero `HighScores.sql`. Si hubiera más de una partida con la misma puntuación, se tomará la última jugada. Una vez insertado, este método, además, calculará el ranking del jugador (mediante el método `setRanking`).

**Ejercicio 4 (2 puntos):** Implementa el método `mejorRacha`. Este método recorrerá la lista de partidas de un jugador de principio a fin y, sin utilizar estructuras de datos auxiliares (sólo variables con tipo de dato nativo) devolverá el máximo número de partidas que, de forma consecutiva, ha mejorado su puntuación.

Por ejemplo, dadas las siguientes puntuaciones:

16	17	8	7	29	32	34	25	17	18	19	7	38	40
----	----	---	---	----	----	----	----	----	----	----	---	----	----

Mejor racha = 4

**Ejercicio 5 (2 puntos):** Implementa el método `mediaPais` que retorne la media de las puntuaciones de los jugadores de un país y que tengan entre 18 y 25 años. El método toma como parámetros:

- el país del cual se quiere calcular la media de puntuaciones
- un array de jugadores

Descargar estructura de clases (Solo enunciado) No tiene solución

<https://bit.ly/ExamenProgramacionExtraordinario>