

Observaciones:

- Versión JAVA 10 como máximo
- 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.

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

```
public interface ExamenProgramacion {
    public int [] sumaColumnasMatriz(int m[][]);
    public void SqlToCsv(String sql, String fichero);
    public int insertaMediaTrabajos(Alumno a);
    public int secuenciaMayor(int []v);
    public int mediaImparesDesdePrimerPar(int []v);
    public String mayorEdad(ArrayList<Mascota> mascotas);
}
```

Ejercicio 1 (1 punto): Implementa el método sumaColumnasMatriz que recibe como parámetro una matriz y retorna como resultado un vector cuyos componentes son la suma de cada columna de la matriz. Por ejemplo, dada la matriz A:

3	4	4	2
2	1	1	0
7	1	2	3

El método debe retornar este array:

12	6	7	5
----	---	---	---

Ejercicio 2: (2 puntos): Implementa el método SqlToCsv que genera un fichero csv con el resultado de la consulta sql que se pasa como parámetro.

Por ejemplo, dada la llamada al método SqlToCsv:

```
p.SqlToCsv("Select dni,nombre,apellidos from Personas","test1.csv");
```

Se generará el siguiente fichero test1.csv

```
61111111H;Iván;López;
15151515P;María;Martínez;
87111111H;Isabel;Márquez;
```

PISTA:Para saber el número de columnas de un resultset puede utilizar las siguientes líneas de código:

```
ResultSetMetaData rsmd = rset.getMetaData();
int numeroColumnas = rsmd.getColumnCount();
```

Ejercicio 3: (1,5 puntos): Dada la tabla NOTAS de la práctica InfoRibera, y la siguientes definiciones:

```
public class Alumno {
    public int codigoAlumno;
    public int codigoModulo; //Modulo en el que está matriculado
    public ArrayList<Trabajo> listaTrabajos;
}
public class Trabajo {
    public String nombreTrabajo;
    public double nota;
}
```

Implementa el método insertaMediaTrabajos, que debe insertar en la BBDD un registro con los campos código de alumno y la nota media de sus trabajos y devolver el número de trabajos del alumno.

Ejercicio 4 (2 puntos): Implementa el método `secuenciaMayor`. Este método recorrerá el array de principio a fin y, sin utilizar estructuras de datos auxiliares (sólo variables con tipo de dato nativo) el máximo número de veces que se repitió un número de forma consecutiva.

Por ejemplo, dado el siguiente array:

2	2	5	5	5	4	4
---	---	---	---	---	---	---

Se repite 3 veces

El número que más se repite de forma consecutiva es el 5, por tanto, el método devolverá un 3.

Ejercicio 5 (1,5 puntos): Implementa el método `mediaImparesDesdePrimerPar`. Este método recorrerá el array de principio a fin y devolverá, sin utilizar estructuras de datos auxiliares (sólo variables con tipo de dato nativo) la media de los números impares que aparecen desde el primer número par.

Por ejemplo, dado el siguiente array:

3	5	2	3	7	4	5
---	---	---	---	---	---	---

Primer par

El método devolverá 5, que es la media de los elementos impares (3,7 y 5) que aparecen después del primer número par (elemento 2 en posición 3)

Ejercicio 6 (2 puntos): Dada la siguiente estructura de clases:

Mascota está definida como:

```
public class Mascota {
    public int edad;
    public String nombre;
}

public class Gato extends Mascota {
    public boolean estaMauullando;
    public Gato(int edad, String nombre, boolean estaMauullando) {
        super(edad, nombre);
        this.estaMauullando = estaMauullando;
    }
}

public class Perro extends Mascota {
    public boolean estaLadrando;

    public Perro(int edad, String nombre, boolean estaLadrando) {
        super(edad, nombre);
        this.estaLadrando = estaLadrando;
    }
}
```

Implementa el método `mayorEdad`, que devolverá el nombre de la mascota de mayor edad y que estén ladrando o mauullando. Si hay dos mascotas con la misma Edad, devolverá el nombre de la primera de ellas.