1 - Garbage Collector Java

=**Proceso mediante el cual los programas de tipo Java realizan una gestión automática de la memoria**

2 - Simplify String resultado="";

if(nota>=5){

resultado="Aprobado";

} else {

resultado="Suspenso";

}

return resultado;

**= 1º return nota >= 5 ? "Aprobado" : "Suspenso";**

**2º return String.valueOf(nota >= 5 ? "Aprobado" : "Suspenso");**

**3º return String.format("%s", nota >= 5 ? "Aprobado" : "Suspenso");**

3 - Que implica que una clase implemente Serializable?

= **Implica que los objetos de esa clase pueden ser convertidos en una secuencia de bytes y por tanto. almacenados en un archivo o enviados a través de una red**

4 Vincula cada palabra con su frase  
Una variable que esté disponible para cualquier instancia de la clase y su valor sea el mismo para todas las instancias de la clase debe ser declarada como **==STATIC**  
Una variable definida en un método y por tanto sólo está disponible dentro del método, se denomina ... **==LOCAL**  
Una variable que pertenezca a la instancia de la clase y su ciclo de vida está ligado al ciclo de vida de la misma debe ser declara dentro de la clase pero fuera de cualquier método y sin el modificador.... **==STATIC**

5 Explica brevemente que significan los tres puntos de la imagen:  
static void varArgsTest（int... myValues){  
==**Indica que estamos trabajando con un parámetro varags; puedes llamar al método con cualquier número de argumentos (0-infinito) del tipo que vinene antes de los 3 puntos**

6. ¿Cual estas dos sentencias usadas en un DAO es más eficiente en SOL?  
Seleccione una:  
﻿PreparedStatement sentencia= conexion.prepareStatement("SELECT id, num factura, cliente id, base imponible, tipo iva id, serie, descripcion, total FROM facturas");  
﻿﻿PreparedStatement sentencia= conexion.prepareStatement("SELECT" FROM facturas"): \*  
== **A**

7 Qué condiciones debería cumplir la clase Factura para que puede ser usada como elemento de Hash Set?  
Set <Factura> factura = new HashSet=<>();  
== **Debe implementar el método hashCode(), Debe implementar el método equals(Object obj)**

8.

Observa estos dos métodos. ¿Qué diferencias hay entre uno y otro?  
public static String myStringTest() (  
String cadena = “”;  
for (int i = 0; i < 1000000; i++) {  
 cadena+=”:”+i;  
return cadena;

public static String myStringBuilderTest() (  
StringBuilder sb= new StringBuilder ();  
for (int i=0; i< 1000000; i++) {  
sb.append(str:":").append(i);  
｝  
return sb.toString();

Seleccione una:  
﻿﻿-Ninguna, los dos devuelven lo mismo, pero el primero (myStringTest) es más legible  
-El primero (myString Test) devuelve una cadena de texto y el otro (myString Builder) devuelve un objeto de tipo StIngßulder  
﻿﻿-Los dos devuelven lo mismo pero el segundo (myStringBuilder) es más eficiente ya no tiene que creat una mueva que se realiza una modificación  
-Los dos métodos devuelven mismo pero el primero (myStringTest) es más eficiente

== **Los dos devuelven lo mismo pero el segundo (myStringBuilder) es más eficiente ya no tiene que crear una nueva cadena cada vez que se realiza una modificación**

9. Describe brevemente que entiendes por POJO en Java

== Plain Old Java Object - **Objeto en Java que contiene datos y métodos para acceder y manipular esos datos.**

10.

*@*param conect Lon

*@*throws SQLException

public static void viewTable(Connection con) throws SQLException

System.out.print(s: "Introduzca un número de empleado: ");

Scanner sc = new Scanner(System.in);

String numEmpleado = sc.nextLine();

String consulta = "select \* from usuarios where id='"+numEmpleado+"'"

System.out.print(consulta);

sc.close();

try (Statement stmt = con.createStatement()) {

ResultSet rs = stmt.executeQuery(consulta);

while (rs.next()) {

String nombre = rs.getString(columnLabel: "nombre");

int id rs.getInt(columnLabel:"id");

System.out.println(id + ","+ nombre);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

¿Ves algún problema de seguridad con este código? Si es así explícalo, o responde negativamente.  
**== Sí, que la entrada del usuario este incluida directamente en la consulta MySQL,y permite a gente con malas intenciones inyectar código malicioso**