|  |  |
| --- | --- |
| Alumnos: | Entrega: 20/5/2025 |

1. Responda:
2. ¿Cómo se convierte una clase en un flujo de bytes?
3. ¿Cuáles son los miembros de la interfaz Serializable?
4. ¿Cuáles son las clases que permiten la persistencia de objetos?
5. ¿Cuál es la diferencia en el uso de FileReader y FileInputStream?
6. ¿Cuáles son las excepciones que pueden lanzarse en un fallo de entrada/salida?
7. ¿Cuál es el paquete a importar para obtener las clases de Entrada/Salida?
8. Complete el siguiente cuadro:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Random Access File | Almacenar Bytes | Almacenar caracteres |
| Declaración de archivo para escribir |  |  |  |
| Declaración de archivo para leer |  |  |  |
| Escribir en archivo |  |  |  |
| Leer en archivo |  |  |  |
| Cerrar Archivo |  |  |  |

1. Cree un programa que muestre en una ventana, botones con las opciones:
2. Escribir archivo de datos: al ser elegida, deberá pedir dos datos que serán guardados: uno libre, ingresado por el usuario, en un campo de texto y otro ya determinado, en un combo de botones de opción o de chequeo. Esto se repetirá las veces que el usuario desee, guardando los datos en una colección, la que luego será persistida.
3. Leer los datos guardados: al ser elegida, se deberá mostrar con textos explicativos, lo que se ingresó. En este caso, si el archivo está vacío, porque aún no se guardó nada, se debe indicar con un mensaje.
4. Terminar: al ser elegida, se deberá informar en qué unidad de almacenamiento, carpeta y nombre del archivo donde se encuentra guardada la información, o advertir que no se guardó, si ese es el caso.

Las características deseables de este programa son:

1. Cada vez que termina de funcionar una opción, se vuelve a mostrar el menú de botones.
2. Captura de excepciones referidas al almacenamiento de la información.
3. Textos claros que ayuden al usuario en todo el proceso.
4. Información clara y precisa, tanto la que se pide como la que se entrega.
5. El programa deberá realizarse utilizando la forma de almacenamiento de objetos.
6. Responda:
   1. **Esta sentencia**

try(DataOutputStream dos=new DataOutputStream(new FileOutputStream("D:\\fich\_binario.ddr"));){ }

**¿Es equivalente a la siguiente?**

DataOutputStream dos=new DataOutputStream(new FileOutputStream("D:\\fich\_binario.ddr"));

try { }

* 1. **Es posible ejecutar este código sin errores? Si los hay, indique dónde y cómo se resuelve.**

ArrayList lista1= new ArrayList(Arrays.asList(new int[]{12, 15, 11, 4, 32});

ObjectOutputStream salida=new ObjectOutputStream(new FileOutputStream("media.obj"));

salida.writeObject("guardar este string y un objeto\n");

salida.writeObject(lista1);

salida.close();

* 1. **Teniendo el siguiente código,**

Figura fig2=new Circulo(12,19, 60);

ObjectOutputStream salida=new ObjectOutputStream(new FileOutputStream("figura.obj"));

salida.writeObject(fig2);

salida.close();

**¿Cómo deben ser las sentencias para leer en el archivo lo guardado?**

* 1. **Observando este código, complete**:

public class Escribir {  
public static void main(String []args) {  
  
String saludo="Hola";  
  
try {  
  
File archivo=new File("texto.txt");  
  
 **Crear objeto de archivo de escritura de bytes**

\_\_\_\_\_\_\_\_\_\_\_\_ escribir=new \_\_\_\_\_\_\_\_\_\_\_\_\_(archivo);  
  
escribir.writeObject(saludo);  
  
escribir.close();  
}  
catch(Exception e) {  
System.out.println("Error con el archivo"); }  
 }  
}

* 1. Respecto al código del punto d. Responda:

1. ¿Cómo se crearía el archivo para escritura de objetos?
2. ¿Cuáles son los motivos que posiblemente lancen la excepción indicada?
3. Viendo que sólo ha sido capturada la clase Exception ¿Cómo se puede determinar el problema que lanzó la excepción?
4. ¿Cómo se podría volver a intentar la escritura del archivo, conociendo la causa de la excepción lanzada? (Elabore el código)