

Modulo 8.- Excepciones y Archivos

Ejercicios del Módulo 8

Ejercicio 1:

Crea un programa que solicite dos números al usuario y los divida. Captura posibles errores como `ArithmeticException` (división por cero) y `InputMismatchException` si el usuario no introduce un número válido.

```
Main.java X Alumno.java
1 package paquete;
2
3 public class Main {
4     public static void main(String[] args) {
5
6         Alumno[] alumnos = {
7             new Alumno("Pablo", 8.5),
8             new Alumno("Ismael", 9.0),
9             new Alumno("Marcos", 7.8)
10        };
11
12
13        String[] lineas = new String[alumnos.length];
14        for (int i = 0; i < alumnos.length; i++) {
15            lineas[i] = alumnos[i].toString();
16        }
17
18
19        String rutaArchivo = "notas.txt";
20
21
22        ArchivoEscritor escritor = new ArchivoEscritor();
23        escritor.escribirArchivo(rutaArchivo, lineas);
24
25
26        ArchivoLector lector = new ArchivoLector();
27        lector.leerArchivo(rutaArchivo);
28    }
29 }
30
31
```

```
ArchivoEscritor.java ArchivoLector.java Alumno.java X Main.java
1 package paquete;
2
3 public class Alumno {
4     private String nombre;
5     private double nota;
6
7     public Alumno(String nombre, double nota) {
8         this.nombre = nombre;
9         this.nota = nota;
10    }
11
12    @Override
13    public String toString() {
14        return nombre + ": " + nota;
15    }
16 }
17
```

```
ArchivoEscritor.java ArchivoLector.java X Alumno.java Main.java
1 package paquete;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5
6 public class ArchivoLector {
7     public void leerArchivo(String rutaArchivo) {
8         try (BufferedReader reader = new BufferedReader(new FileReader(rutaArchivo))) {
9             System.out.println("\nContenido del archivo '" + rutaArchivo + "':");
10            String linea;
11            while ((linea = reader.readLine()) != null) {
12                System.out.println(linea);
13            }
14        } catch (IOException e) {
15            System.err.println("Error al leer el archivo: " + e.getMessage());
16        }
17    }
18 }
19
20
```

```
ArchivoEscritor.java X ArchivoLector.java Alumno.java Main.java
1 package paquete;
2 import java.io.BufferedWriter;
3 import java.io.FileWriter;
4 import java.io.IOException;
5 public class ArchivoEscritor {
6     public void escribirArchivo(String rutaArchivo, String[] lineas) {
7         try (BufferedWriter writer = new BufferedWriter(new FileWriter(rutaArchivo))) {
8             for (String linea : lineas) {
9                 writer.write(linea);
10                writer.newLine();
11            }
12            System.out.println("El archivo '" + rutaArchivo + "' ha sido creado y escrito exitosamente.");
13        } catch (IOException e) {
14            System.err.println("Error al escribir en el archivo: " + e.getMessage());
15        }
16    }
17 }
18
19
```

```
<terminated> Main (10) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (2
El archivo 'notas.txt' ha sido creado y escrito exitosamente.
```

```
Contenido del archivo 'notas.txt':
Pablo: 8.5
Ismael: 9.0
Marcos: 7.8
```

Ejercicio 2:

Crea un archivo `notas.txt`. Escribe al menos tres líneas con nombre y nota de un alumno. Luego, abre el archivo y muestra su contenido línea por línea.

```
ArchivoEscriitor.java ArchivoLector.java Main.java X
1 package paquete;
2
3 public class Main {
4     public static void main(String[] args) {
5
6         String[] lineas = {
7             "David: 8.5",
8             "Daniel: 9.0",
9             "Raul: 7.8"
10        };
11
12
13        String rutaArchivo = "notas.txt";
14
15
16        ArchivoEscriitor escritor = new ArchivoEscriitor();
17        escritor.escribirArchivo(rutaArchivo, lineas);
18
19
20        ArchivoLector lector = new ArchivoLector();
21        lector.leerArchivo(rutaArchivo);
22    }
23 }
24
25
```

```
ArchivoEscriitor.java X ArchivoLector.java X Main.java
1 package paquete;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5
6 public class ArchivoLector {
7
8     public void leerArchivo(String rutaArchivo) {
9         try (BufferedReader reader = new BufferedReader(new FileReader(rutaArchivo))) {
10             System.out.println("\nContenido del archivo " + rutaArchivo + " :");
11             String linea;
12             while ((linea = reader.readLine()) != null) {
13                 System.out.println(linea);
14             }
15         } catch (IOException e) {
16             System.err.println("Error al leer el archivo: " + e.getMessage());
17         }
18     }
19 }
20
21
```

```
ArchivoEscritor.java X ArchivoLector.java Main.java
1 package paquete;
2 import java.io.BufferedWriter;
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 public class ArchivoEscritor {
7
8     public void escribirArchivo(String rutaArchivo, String[] lineas) {
9         try (BufferedWriter writer = new BufferedWriter(new FileWriter(rutaArchivo))) {
10             for (String linea : lineas) {
11                 writer.write(linea);
12                 writer.newLine();
13             }
14             System.out.println("El archivo '" + rutaArchivo + "' ha sido creado y escrito exitosamente.");
15         } catch (IOException e) {
16             System.err.println("Error al escribir en el archivo: " + e.getMessage());
17         }
18     }
19 }
20
21
```

<terminated> Main (11) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (25)

El archivo 'notas.txt' ha sido creado y escrito exitosamente.

Contenido del archivo 'notas.txt':

David: 8.5

Daniel: 9.0

Raul: 7.8

Ejercicio 3:

Amplía el programa anterior para que en lugar de sobrescribir el archivo, se añada un nuevo alumno sin borrar los anteriores.

```
ArchivoEscritor.java ArchivoLector.java Main.java X
1 package paquete;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6
7         String rutaArchivo = "notas.txt";
8
9
10        ArchivoEscritor escritor = new ArchivoEscritor();
11        ArchivoLector lector = new ArchivoLector();
12
13
14        String[] lineasIniciales = {
15            "David: 8.5",
16            "Daniel: 9.0",
17            "Raúl: 7.8"
18        };
19
20
21        escritor.escribirArchivo(rutaArchivo, lineasIniciales, false);
22
23
24        lector.leerArchivo(rutaArchivo);
25
26
27        Scanner scanner = new Scanner(System.in);
28        System.out.println("\n¿Desea añadir un nuevo alumno? (s/n): ");
29        String respuesta = scanner.nextLine();
30
31        if (respuesta.equalsIgnoreCase("s")) {
32            System.out.println("Ingrese el nombre del alumno: ");
33            String nombre = scanner.nextLine();
34
35            System.out.println("Ingrese la nota del alumno: ");
36            String nota = scanner.nextLine();
37
38
39            String nuevaLinea = nombre + ": " + nota;
40
41
42            escritor.escribirArchivo(rutaArchivo, nuevaLinea, true);
43
44
45            lector.leerArchivo(rutaArchivo);
46        }
47
48        scanner.close();
49    }
50 }
51
```



```
ArchivoEscritor.java ArchivoLector.java X Main.java
1 package paquete;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5
6 public class ArchivoLector {
7
8     public void leerArchivo(String rutaArchivo) {
9         try (BufferedReader reader = new BufferedReader(new FileReader(rutaArchivo))) {
10             System.out.println("\nContenido del archivo '" + rutaArchivo + "':");
11             String linea;
12             while ((linea = reader.readLine()) != null) {
13                 System.out.println(linea);
14             }
15         } catch (IOException e) {
16             System.err.println("Error al leer el archivo: " + e.getMessage());
17         }
18     }
19 }
20
```

```
ArchivoEscritor.java X ArchivoLector.java Main.java
1 package paquete;
2 import java.io.BufferedWriter;
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 public class ArchivoEscritor {
7
8     public void escribirArchivo(String rutaArchivo, String[] lineas, boolean append) {
9         try (BufferedWriter writer = new BufferedWriter(new FileWriter(rutaArchivo, append))) {
10             for (String linea : lineas) {
11                 writer.write(linea);
12                 writer.newLine();
13             }
14             System.out.println("El archivo '" + rutaArchivo + "' ha sido actualizado exitosamente.");
15         } catch (IOException e) {
16             System.err.println("Error al escribir en el archivo: " + e.getMessage());
17         }
18     }
19
20     public void escribirArchivo(String rutaArchivo, String linea, boolean append) {
21         try (BufferedWriter writer = new BufferedWriter(new FileWriter(rutaArchivo, append))) {
22             writer.write(linea);
23             writer.newLine();
24             System.out.println("El archivo '" + rutaArchivo + "' ha sido actualizado exitosamente.");
25         } catch (IOException e) {
26             System.err.println("Error al escribir en el archivo: " + e.getMessage());
27         }
28     }
29 }
30
31
32
```

Problems @ Javadoc Declaration Console X
<terminated> Main (12) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (25 at
El archivo 'notas.txt' ha sido actualizado exitosamente.

Contenido del archivo 'notas.txt':

David: 8.5

Daniel: 9.0

Raúl: 7.8

¿Desea añadir un nuevo alumno? (s/n):

n
|

Ejercicio 4:

Crea una clase `Producto` con atributos `nombre` y `precio`. Crea un objeto de esa clase, serialízalo y guárdalo en un archivo. Luego deserialízalo y muestra sus atributos en pantalla.

```
Producto.java X Serializador.java Main.java X
1 package paquete;
2
3 public class Main {
4     public static void main(String[] args) {
5
6         String rutaArchivo = "producto.ser";
7
8
9         Producto producto = new Producto("PS5", 500);
10
11
12         Serializador serializador = new Serializador();
13
14
15         serializador.serializar(rutaArchivo, producto);
16
17
18         Producto productoDeserializado = (Producto) serializador.deserializar(rutaArchivo);
19
20
21         if (productoDeserializado != null) {
22             System.out.println("\nAtributos del objeto deserializado:");
23             System.out.println(productoDeserializado);
24         }
25     }
26 }
27
28
```

```
Producto.java Serializador.java X Main.java
1 package paquete;
2 import java.io.FileInputStream;
3 import java.io.FileOutputStream;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
6 import java.io.IOException;
7
8 public class Serializador {
9
10     public void serializar(String rutaArchivo, Object objeto) {
11         try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(rutaArchivo))) {
12             oos.writeObject(objeto);
13             System.out.println("Objeto serializado y guardado en el archivo '" + rutaArchivo + "'.");
14         } catch (IOException e) {
15             System.err.println("Error al serializar el objeto: " + e.getMessage());
16         }
17     }
18
19     public Object deserializar(String rutaArchivo) {
20         try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(rutaArchivo))) {
21             Object objeto = ois.readObject();
22             System.out.println("Objeto deserializado desde el archivo '" + rutaArchivo + "'.");
23             return objeto;
24         } catch (IOException | ClassNotFoundException e) {
25             System.err.println("Error al deserializar el objeto: " + e.getMessage());
26             return null;
27         }
28     }
29 }
30
31
32
```



```
Producto.java X Serializador.java Main.java
1 package paquete;
2 import java.io.Serializable;
3
4 public class Producto implements Serializable {
5     private static final long serialVersionUID = 1L;
6     private String nombre;
7     private double precio;
8
9
10    public Producto(String nombre, double precio) {
11        this.nombre = nombre;
12        this.precio = precio;
13    }
14
15
16    public String getNombre() {
17        return nombre;
18    }
19
20    public double getPrecio() {
21        return precio;
22    }
23
24
25    public String toString() {
26        return "Producto [nombre=" + nombre + ", precio=" + precio + "]";
27    }
28 }
29
30
```

```
Problems @ Javadoc Declaration Console X
<terminated> Main (13) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (2
Objeto serializado y guardado en el archivo 'producto.ser'.
Objeto deserializado desde el archivo 'producto.ser'.

Atributos del objeto deserializado:
Producto [nombre=PS5, precio=500.0]
```

Ejercicio 5:

Crea un método que reciba por parámetro el nombre de un archivo. Si no existe, lanza una excepción usando `throws`. Desde `main`, captura la excepción y muestra un mensaje amigable para el usuario.

```
ArchivoUtils.java ArchivoNoEncontradoException.java Main.java X
1 package paquete;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6
7         Scanner scanner = new Scanner(System.in);
8         System.out.println("Ingrese el nombre del archivo a verificar: ");
9         String nombreArchivo = scanner.nextLine();
10
11         try {
12             ArchivoUtils.verificarExistenciaArchivo(nombreArchivo);
13         } catch (ArchivoNoEncontradoException e) {
14
15             System.err.println("Error: " + e.getMessage());
16         } finally {
17
18             scanner.close();
19         }
20     }
21 }
22
23
24
```

```
ArchivoUtils.java ArchivoNoEncontradoException.java X Main.java
1 package paquete;
2
3 public class ArchivoNoEncontradoException extends Exception {
4
5     public ArchivoNoEncontradoException(String mensaje) {
6         super(mensaje);
7     }
8 }
9
```

```
ArchivoUtils.java X ArchivoNoEncontradoException.java Main.java
1 package paquete;
2 import java.io.File;
3
4 public class ArchivoUtils {
5     public static void verificarExistenciaArchivo(String nombreArchivo) throws ArchivoNoEncontradoException {
6         File archivo = new File(nombreArchivo);
7         if (!archivo.exists()) {
8             throw new ArchivoNoEncontradoException("El archivo '" + nombreArchivo + "' no existe.");
9         }
10        System.out.println("El archivo '" + nombreArchivo + "' existe.");
11    }
12 }
13
14
```

```
Problems @ Javadoc Declaration Console X
<terminated> Main (14) [Java Application] C:\Program Files\Java\jdk-23\bin\j
Ingrese el nombre del archivo a verificar:
hola
Error: El archivo 'hola' no existe.
```

<https://github.com/CR1PTAS/programaci-n.git>