



Enunciado del ejercicio – Generación de tabla de multiplicar en archivo de texto


Realiza un programa en Java que cumpla con los siguientes requisitos:


1. Solicite al usuario que introduzca un número entero entre 1 y 10.
2. Verifique que el número ingresado esté dentro del rango válido (1–10).
 - Si el número es válido, el programa generará su tabla de multiplicar del 1 al 10.
 - Cada línea de la tabla debe seguir el formato:
 $n \times i = \text{resultado}$
3. La tabla generada debe guardarse automáticamente en un archivo de texto con el nombre `tabla-n.txt`, donde `n` representa el número introducido por el usuario.
4. Si el número no está dentro del rango permitido, el programa mostrará un mensaje de error adecuado y no generará ningún archivo.

Enunciado del ejercicio – Lectura y visualización de tabla de multiplicar desde archivo

Crea un programa en Java que permita **leer y mostrar una tabla de multiplicar desde un archivo de texto**, cumpliendo con las siguientes condiciones:

1. El programa debe solicitar al usuario que introduzca un **número entero entre 1 y 10**.
2. A partir de ese número, el programa intentará **abrir un archivo de texto llamado `tabla-n.txt`**, donde `n` representa el número introducido por el usuario.
3. Si el archivo existe:
 - El contenido del archivo (la tabla de multiplicar del número) se mostrará línea por línea por consola.
4. Si el archivo **no existe**, el programa debe notificarlo con un mensaje de error claro:
"El archivo `tabla-n.txt` no existe."
5. El programa debe utilizar estructuras adecuadas para la lectura de archivos (`BufferedReader`) y debe manejar las excepciones correspondientes (`FileNotFoundException` y `IOException`).
6. Se debe asegurar el **cierre adecuado del archivo**, incluso en caso de error.

 **Nota:** Este ejercicio es complementario al anterior, donde se genera la tabla y se guarda en un archivo de texto.

 Enunciado del ejercicio – Lectura de una línea específica de una tabla de multiplicar desde archivo

Diseña un programa en Java que realice lo siguiente:

Solicite al usuario dos números enteros entre 1 y 10, denominados n y m.

n corresponde al número cuya tabla de multiplicar se quiere consultar.

m representa la línea específica (del 1 al 10) de la tabla que se desea mostrar.

El programa buscará un archivo de texto llamado tabla-n.txt.

Si el archivo existe:


El programa mostrará únicamente la línea número m de dicho archivo, correspondiente a la multiplicación $n \times m$.

Ejemplo: si $n = 3$ y $m = 4$, se imprimirá $3 \times 4 = 12$ si esa línea existe en el archivo.

Si el archivo no existe, se mostrará un mensaje de error adecuado:

"El archivo tabla-n.txt no existe."

El programa debe implementar una correcta gestión de errores mediante excepciones como `FileNotFoundException` y `IOException`.

 Este ejercicio requiere el uso de clases como `BufferedReader`, `FileReader`, estructuras de control de flujo y manejo de excepciones.

Enunciado del ejercicio – Gestión de listín telefónico con archivos de texto

Desarrolla un programa en Java que simule un **listín telefónico**, permitiendo consultar, agregar y eliminar registros de clientes. El sistema debe cumplir los siguientes requisitos:

Requisitos funcionales:

1. **Al iniciar el programa**, se verificará la existencia del archivo listin.txt.
 - Si no existe, el programa lo creará automáticamente.
 - Este archivo almacenará la información de los clientes en formato nombre,telefono (una línea por cliente).
2. Se presentará un **menú con las siguientes opciones**:
 - 1. Consultar teléfono de un cliente
 - 2. Añadir teléfono de un nuevo cliente
 - 3. Eliminar teléfono de un cliente
 - 4. Salir
3. **Consultar teléfono**:
 - Solicita al usuario el nombre de un cliente.
 - Busca en el archivo listin.txt y muestra el número si lo encuentra.
 - Si el cliente no existe, notifica al usuario.
4. **Añadir nuevo cliente**:
 - Pide el nombre y el número del nuevo cliente.
 - Añade la información al final del archivo, manteniendo el formato adecuado.
5. **Eliminar cliente**:
 - Solicita el nombre del cliente a eliminar.
 - Crea un archivo temporal con todos los registros excepto el del cliente indicado.
 - Sustituye el archivo original por el temporal si se elimina con éxito.
 - Muestra un mensaje de éxito o error según el resultado.

Requisitos técnicos:

- El programa debe usar estructuras de control (switch, while, etc.) para gestionar el menú.
- Debe utilizar `BufferedReader` y `BufferedWriter` para las operaciones de archivo.
- Manejar adecuadamente excepciones como `IOException` y asegurar el cierre correcto de los recursos.