LECTURA DE DOCUMENTOS

Marcos Gómez Martínez

Para crear este programa, primero necesitamos identificar claramente los pasos que debemos seguir y los elementos que necesitaremos. Aquí hay una lista de lo que necesitamos:

1. Un archivo de texto del cual leeremos las líneas.

2. Un método para abrir y leer el archivo de texto.

3. Un método para imprimir cada línea del archivo.

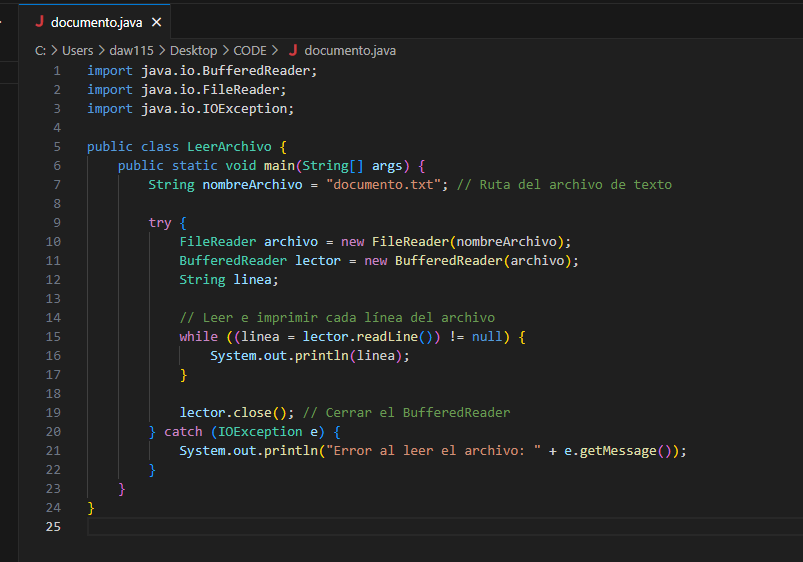
Con eso en mente, podemos abordar la programación de la siguiente manera:

1. Abrir el archivo de texto en modo de lectura.

2. Leer cada línea del archivo.

3. Imprimir cada línea en la consola.

4. Cerrar el archivo después de leer todas las líneas.



En este código, definimos una función `leer\_archivo` que toma la ruta del archivo como entrada. Luego, intentamos abrir el archivo en modo de lectura y leemos todas las líneas. Después, iteramos sobre estas líneas e imprimimos cada una en la consola. Cerramos el archivo después de leer todas las líneas. Además, manejamos las excepciones que pueden ocurrir, como el archivo no encontrado.

Para probar este código, simplemente necesitaríamos tener un archivo de texto llamado "documento.txt" en la misma carpeta que el script.

**Problemas e solución:**

El programa nos da un error cuando intentamos abrir cualquier archivo de texto con formato, puesto que no está preparado para ello. Los formatos .odt, .docx, .pdf y .rtf no son archivos de texto plano, por lo que no se pueden leer directamente usando un “BufferedReader”. Estos formatos tienen estructuras más complejas como imágenes o tablas, por lo que necesitamos usar bibliotecas más específicas.

Además estos formatos pueden contener texto en diferentes codificaciones de caracteres, lo que puede requerir manejar la codificación de caracteres adecuada al leer el archivo y así evitar problemas de caracteres extraños o ilegibles.

Por último cada uno de estos formatos tiene una estructura específica y compleja, por tanto necesitamos analizar la estructura del archivo y utilizar bibliotecas adecuadas para extraer el texto de manera efectiva.

Para solucionar el problema utilizaremos bibliotecas específicas para cada tipo de archivo:

- Para archivos .odt, usaremos Apache ODF Toolkit (por ejemplo, `odfdom-java`) en Java.

- Para archivos .docx, usaremos Apache POI o docx4j.

- Para archivos .pdf, usaremos bibliotecas como Apache PDFBox o iText.

- Para archivos .rtf no hay una biblioteca estándar en Java, sin embargo podríamos convertir los archivos .rtf a formato de texto plano (.txt) y luego leerlos con nuestro método existente.

El programa quedaría así:

