

Entender cómo funciona el Scanner

Analogía del Scanner con un Buzón



Imaginate que tenés un buzón de correo en tu casa y deseas leer las cartas que llegan a ese buzón. El objeto **Scanner** sería como tu mano que abre el buzón, toma una carta a la vez y te la entrega para que puedas leer su contenido. Aquí está cómo funciona la analogía:

1. **Crear un objeto Scanner:** Esto sería equivalente a tener una mano lista para abrir el buzón.
2. **Seleccionar la fuente de entrada:** Puedes decirle a tu mano que el buzón de correo es la fuente de entrada que debe leer.
3. **Leer los datos:** Cuando desees leer una carta, tu mano abre el buzón y toma una carta. El **Scanner** en Java lee datos de manera similar. Utilizas métodos como `nextInt()`, `nextLine()`, etc, para decirle al **Scanner** que lea datos específicos desde la fuente de entrada.
4. **Almacenar los datos:** Una vez que tengas la carta en tu mano, puedes guardarla en una carpeta o leer su contenido. De manera similar, los datos leídos por el **Scanner** se pueden almacenar en variables para su posterior procesamiento o uso en tu programa.
5. **Cerrar el buzón:** Cuando hayas terminado de leer todas las cartas, es importante cerrar el buzón para evitar que las nuevas cartas se acumulen y se pierdan las antiguas. En Java, usar `close()` en el objeto **Scanner** es equivalente a cerrar el buzón después de leer todas las entradas.
6. **Recursos liberados:** Al cerrar el buzón, aseguras que no se desperdicien recursos innecesariamente. En Java, al cerrar el **Scanner**, liberas los recursos del sistema que se asignaron para la lectura de la fuente de entrada.

En pocas palabras: un objeto **Scanner** en Java actúa como una "mano" que te permite leer y procesar datos de una fuente de entrada, como si estuvieras abriendo y leyendo cartas de un buzón. Es importante abrir y cerrar adecuadamente el **Scanner** para asegurarte de que los datos se manejen de manera correcta y eficiente, y para liberar los recursos cuando ya no se necesiten.