Serializar y deserializar un objeto almacenado en un fichero en Kotlin

Versión simplificada

[crayon-6a9d694943fb0615219236/]

En esta versión, se utiliza el método use en combinación con las funciones ObjectOutputStream y ObjectInputStream. Esto asegura que los recursos se cierren automáticamente una vez finalizada su utilización.

Además, se utilizan las funciones FileOutputStream y FileInputStream directamente dentro de las llamadas a ObjectOutputStream y ObjectInputStream, respectivamente, para evitar la necesidad de crear variables temporales.

En resumen, esta versión simplificada mantiene la funcionalidad original de serializar y deserializar un objeto almacenado en un archivo utilizando la interfaz Serializable, pero con un código más conciso y menos líneas de código.

Versión sin simplificar

[crayon-6a9d694943fb5401884090/]

En este ejemplo, la clase Persona implementa la interfaz Serializable. Luego, la función serializarObjeto recibe un objeto y un archivo como parámetros. Crea un flujo de salida de objetos y escribe el objeto en un flujo de bytes utilizando writeObject. Luego, guarda los bytes en el archivo especificado.

La función deserializarObjeto recibe un archivo y lee los

bytes del archivo en un flujo de entrada de objetos. Utiliza readObject para leer el objeto deserializado y lo devuelve.

En la función main, creamos una instancia de Persona, serializamos el objeto y luego lo deserializamos, imprimiendo sus propiedades si se pudo deserializar correctamente.

Ten en cuenta que al utilizar la interfaz Serializable, debes asegurarte de que todas las propiedades de la clase y las clases relacionadas también sean serializables. Además, ten en cuenta que la serialización y deserialización de objetos puede tener implicaciones de seguridad y compatibilidad, especialmente cuando se trata de transferir objetos entre diferentes versiones o sistemas.