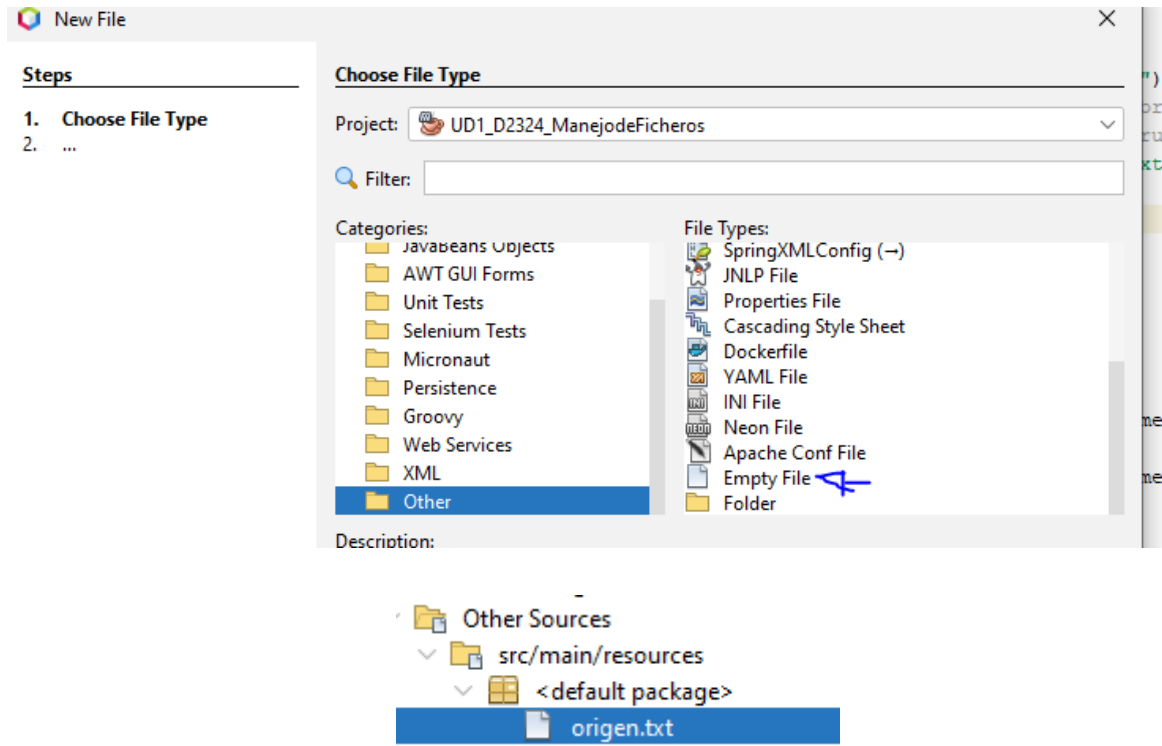


ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

ENUNCIADO 9: Leer/escribir ficheros de texto carácter a carácter.

Crea un fichero en la carpeta resources (dentro de `src/main`) denominado `origen.txt`. El fichero puedes crearlo directamente mediante la opción `new File` → `Other` → `Empty File` → Escribir nombre del fichero (`origen.txt`) y escribir directamente en él el contenido del mismo (varias líneas de texto).



Otra opción es crearlo en el propio código y luego incorporarle nosotros directamente el contenido.

Escribe un programa que permita crear un fichero de texto (`destino.txt`) a partir del fichero de texto creado (`origen.txt`).

El proceso de lectura/escritura (I/O) de los ficheros de texto lo harás carácter a carácter utilizando las clases `FileReader` y `FileWriter`.

¿Que ocurre si le indicas como segundo parámetro al fichero de destino la opción `True`?

Es importante que observes que todos ficheros deben abrirse y cerrarse. En este ejercicio si declaramos las variables en el `try` ya se cierran sin necesidad de hacerlo en el `finally`.

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

ENUNCIADO 10: Leer/escribir ficheros de texto linea a linea.

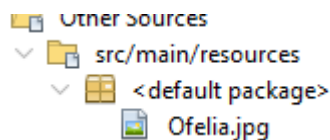
Escribe un programa que permita crear un fichero de texto (destino.txt) a partir de un fichero de texto (origen.txt). En este caso no lo vamos a crear si no existe, sino que indicamos que no existe.

El proceso de lectura/escritura (I/O) de los ficheros de texto lo harás linea a linea utilizando las clases FileReader y FileWriter, así como BufferedReader y BufferedWriter. Además deberás comprobar que el fichero origen.txt existe, si no existe no realizará el proceso.

ENUNCIADO 11: Leer/escribir ficheros binarios

Escribe un programa que permita crear un fichero binario a partir de un fichero binario en origen. Elige un fichero binario (ejem: una imagen) para realizar la prueba. El proceso de lectura/escritura (I/O) de los ficheros lo harás utilizando las clases FileInputStream y FileOutputStream.

En nuestra propuesta de solucion proponemos a ofelia.jpg , pero podría ser cualquier imagen, video,..



ENUNCIADO 12_13: Leer/escribir ficheros de datos básicos.

Entendemos por datos básicos que guardaremos tipos de datos definidos en java: int,double, String...

Escribe un programa que permita crear un fichero de datos (nombre ,edad y salario de personas) denominado fichorigen.dat. Para evitar la petición de datos por teclado, los datos los inicializarás en la aplicación.

A continuación crearás otro programa que permita leer el fichorigen.dat y mostrará los datos por pantalla.

El proceso de lectura/escritura (I/O) de los ficheros lo harás utilizando las clases FileInputStream, FileOutputStream, DataOutputStream y DataInputStream.

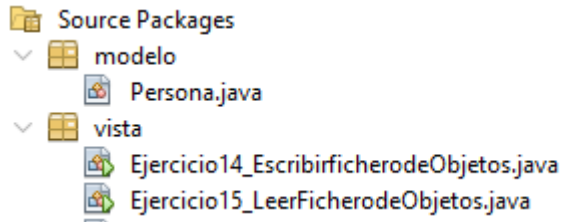
ENUNCIADO 14_15: Leer/escribir ficheros de objetos.

Escribe un programa que permita crear un fichero de objetos de la clase persona , guardaremos nombre , edad y salario, denominado fichbinarioorigen.dat. Para evitar la petición de datos por teclado, los datos los inicializarás en la aplicación.

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

Importante: Para poder escribir objetos de tipo persona. Necesitarás crear una clase de tipo persona. Esta clase la crearás en un paquete denominado modelo. Es importante que la clase implemente la interfaz Serializable. La estructura quedará así:



A continuación crearás otro programa que permita leer el fichero origen.dat y mostrará los datos por pantalla.

El proceso de lectura/escritura (I/O) de los ficheros lo harás utilizando las clases FileInputStream, FileOutputStream, ObjectOutputStream y ObjectInputStream.

ENUNCIADO 16_17: Leer/escribir RAF

Escribe un programa que permita crear un fichero de acceso aleatorio, un random access file, denominado ficheroRAF.dat

Los datos que va a guardar tendrá este formato:

numeroempleado entero + nombre (10 caracteres)+ edad entero + salario double

Teniendo en cuenta que los enteros son 4 bytes, los caracteres el doble de bytes de lo que ocupan y los doubles 4 bytes. Tenemos que el registro ocupa $4+20+4+8=36$ bytes.

Para evitar la petición de datos por teclado, los datos los inicializarás en la aplicación.

A continuación crearás un programa que permita leer los datos del fichero creado. Puedes verificar que puedes leer un registro cualquiera, por eso es de acceso aleatorio, desplazándote con el seek directamente al registro mediante la fórmula $(\text{numeroderegistroabuscar}-1) \times \text{tamañoderegistro}$.