# Ejercicios java

## Ejecuta una división por 0 metiendo excepciones

En un programa java puede existir algún tipo de problema o error durante la ejecución del mismo. Cuando esto sucede, se lanza una excepción .Pude ser una división entre 0 , disco duro lleno , una posición de vector un que no existe , hacer un cast invalido….

Cuando un programa lanza una excepción la ejecución del programa no continua.

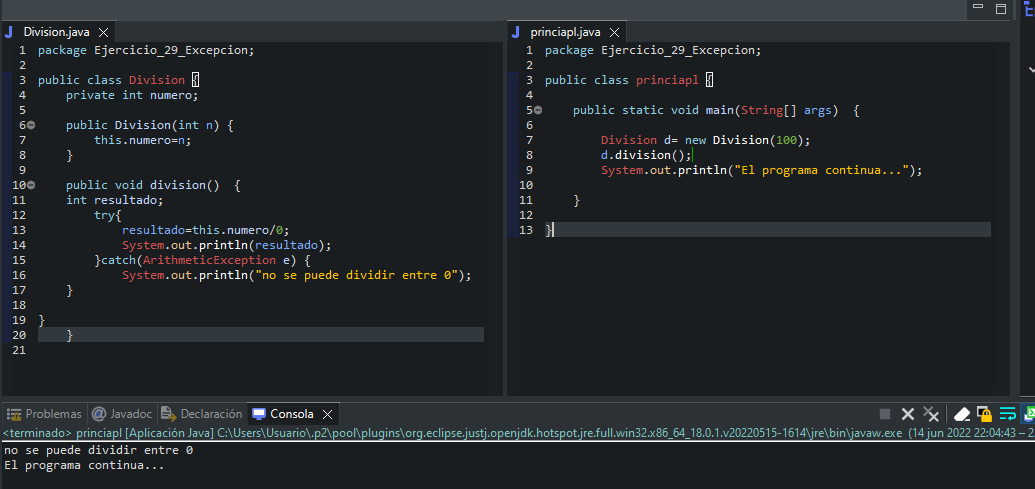
Para controlar una excepción y continuar con la ejecución:

Try-catch-fianally

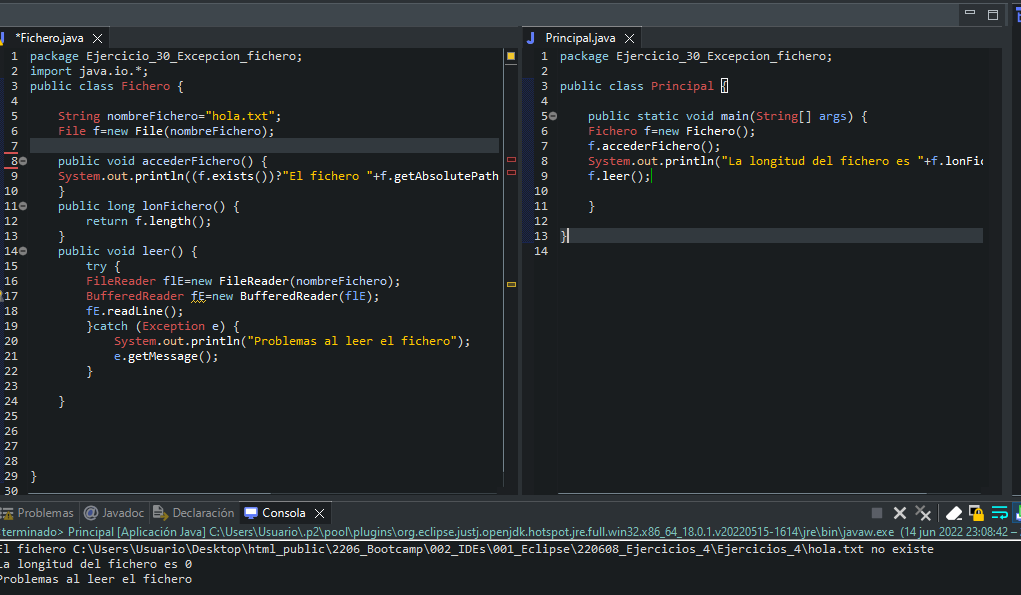
Para indicar uqe se pude lanzar una excepción:

Throws

Las excepciones se pude extender para crear excepciones propias y lanzarlas.

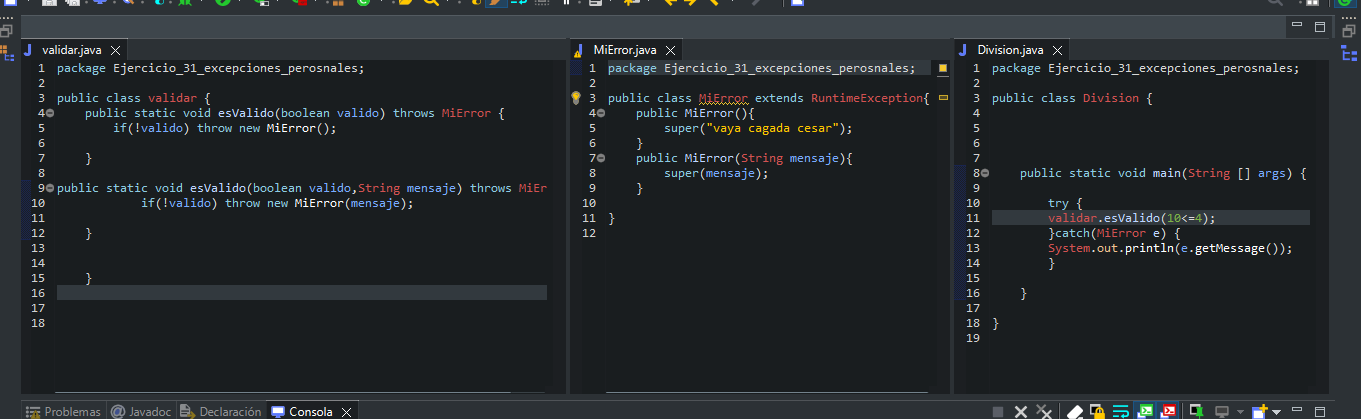


## Crea una clase que contenga un método que lea de un fichero que no existe. Haz distintas pruebas a llamar a ese método.



## Crea una excepción propia, que extienda de una existente (puede ser solo crear los constructores) y pruebas de lanzar ese tipo de excepción sobre la clase del ejercicio 30 (podéis duplicar la clase si no queréis tocar la del 30)

El programador java pude crear sus propias clase de excepciones



## Crea un método que lea de un fichero y muestra su contenido por pantalla

La vocación del lenguaje java de ser independinte del sistema operativo supone una dificltad añadida en los procesos de escritura y lectura de ficheros , sobre todo en aplicaciones que acceden a ficheros a través de su URL.

La escritura y lectura de ficheros se basa en flujos de datos, sobre estos fujos se pude leer y escribir datos, los cuales son el canal de de comunicación entere el programa en al memoria del ordenador y el fichero de soporte (disco) de almacenamiento.Los flujos se puden clasificar en flujos de bytes(clases InputStream y OutputStream) y fujos de caracteres (clases Reader y Writer) .Existen otras clases que sin conectar directamente con el origen de datos , por ejemplo un fichero,perimten transformar los datos por las clases generadoras de flujos.A estas clases se les denomina filtros.

El método generald de lectura en ficheros consiste en enlazar un flujo y un filtro , creando un canal completo de tansferencia de datos,

LAS CLASES READ Y WRITER

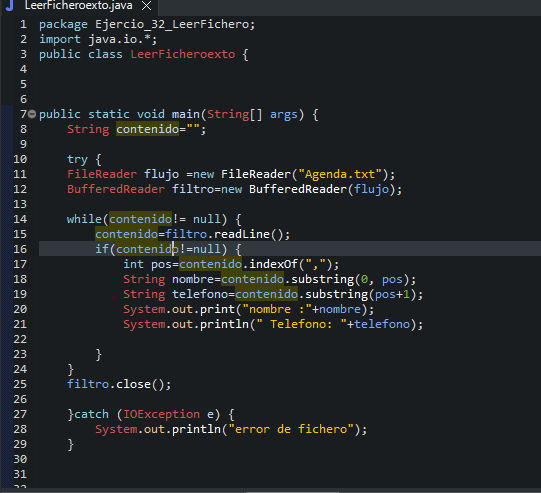
Son las clases nastractas de las que derivan las clases que prmiten crar fujos de caractres Unicode(16bits) de lectura y escritura en ficheros , la siguientes son las clases mas usadas para la lecutra y escritura derivavas de Reader y Writer:

FileReader permite crear fuos de bytes procedente de un fichero de bytes

FilerWriter permite crear flujos de bytes para en enviar a un fichero

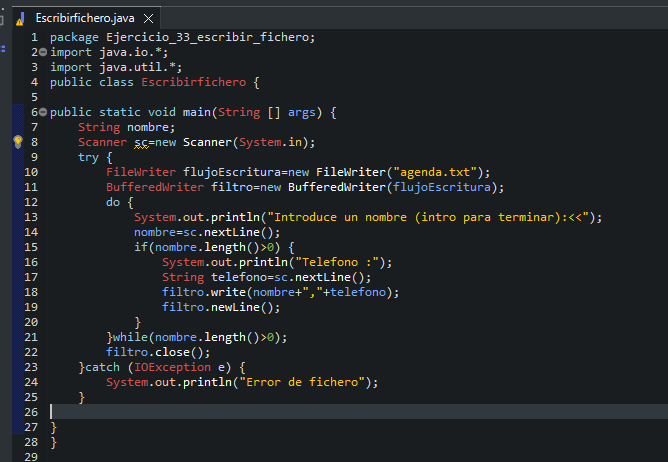
BufferedReader Permite crear un flujo de entrada de datos con area de almacenamiento (buffer) con lo que permite la lectura de datos mayores de un byte;

bufferedWriter Permite crear un flujo de envio de datos con area de almacenamiento (buffer);



## Crea un método que escriba en un fichero.

Para ecribir un fichero línea a línea en Java se recomienda utilizar un objeto flujo derivado de clase FileWriter y un filtro bufferedWriter , sobre este ultimo se aplica el método newLine que inserta los carasers de final de línea en un fichero de texto.



## Crea un método que borre un fichero dado.

Mencion especial se merece la clase File de utilidades de identificación de ficheros y directorios.El objeto de la clase File los podremos llmar fuente identifica al fichero.

Tiene los siguientes constructores:

File(String directorio)

File(String fichero)

File(Sring directorio,String Fichero)

File(File directorio,String fichero)

Si el objeto es un fichero:

Boolean isFile()

Boolean exists()

Bóllenla canRead();

Boolean delete()

Long length()

Long lasModifiqued();

Boolean renameTo()

