

METODOS PARA CADENAS

- **char charAt(int index)** Proporciona el carácter que está en la posición dada por el entero index.
- **int compareTo(String s)** Compara las dos cadenas. Devuelve un valor menor que cero si la cadena s es mayor que la original, devuelve 0 si son iguales y devuelve un valor mayor que cero si s es menor que la original.
- **int compareToIgnoreCase(String s)** Compara dos cadenas, pero no tiene en cuenta si el texto es mayúsculas o no.
- **String concat(String s)** Añade la cadena s a la cadena original.
- **boolean endsWith(String s)** Devuelve true si la cadena termina con el texto s
- **boolean equals(String s)** Compara ambas cadenas, devuelve true si son iguales
- **boolean equalsIgnoreCase(String s)** Compara ambas cadenas sin tener en cuenta las mayúsculas y las minúsculas.
- **int indexOf(String s)** Devuelve la posición en la cadena del texto s
- **int indexOf(String s, int primeraPos)** Devuelve la posición en la cadena del texto s, empezando a buscar desde la posición primeraPos
- **int lastIndexOf(String s)** Devuelve la última posición en la cadena del texto s
- **int lastIndexOf(String s, int primeraPos)** Devuelve la última posición en la cadena del texto s, empezando a buscar desde la posición primeraPos
- **int length()** Devuelve la longitud de la cadena
- **String replace(char carAnterior, char ncarNuevo)** Devuelve una cadena idéntica al original pero
- **String replaceAll(String str1, String str2)** Cambia todas las apariciones de la cadena uno por la cadena dos
- **String startsWith(String s)** Devuelve true si la cadena comienza con el texto s.
- **String substring(int primeraPos, int segundaPos)** Devuelve el texto que va desde primeraPos a segundaPos.
- **char[] toCharArray()** Devuelve un array de caracteres a partir de la cadena dada
- **String toLowerCase()** Convierte la cadena a minúsculas
- **String toUpperCase()** Convierte la cadena a mayúsculas
- **String trim()** Elimina los blancos que tenga la cadena tanto por delante como por detrás

MÉTODOS DE LA CLASE MATH

MÉTODO	DESCRIPCIÓN	PARÁMETROS	TIPO DE DATO DEVUELTO
abs	Devuelve el valor absoluto de un número.	Un parámetro que	El mismo que introduces.
ceil	Devuelve el entero más cercano por arriba.	Double	Double
floor	Devuelve el entero más cercano por debajo.	Double	Double
round	Devuelve el entero más cercano.	Double o float	long (si introduces un double) o int
exp	Devuelve el exponencial de un número.	Double	Double
max	Devuelve el mayor de dos entre dos valores.	Dos parámetros que	El mismo tipo que introduces.
min	Devuelve el menor de dos entre dos valores.	Dos parámetros que	El mismo tipo que introduces.
random	Devuelve un número aleatorio entre 0 y 1. Se	Ninguno	Double
sqrt	Devuelve la raíz cuadrada de un número.	Double	Double
pow	Devuelve un número elevado a un exponente.	Dos parámetros double	Double

EXCEPCIONES

Exception, es un error genérico.

NumberFormatException, captura letras por números.

ArithmeticException, realiza una división por cero.

InputMismatchException, la captura de datos no es correcta.

ArrayIndexOutOfBoundsException, si sobrepasamos el tamaño del array.

ClassNotFoundException, al tratar de utilizar una clase.

DataFormatException, error en el formato de datos.

IllegalAccessException, se intenta acceder a una clase a la que no se tiene permiso. Como puede ser ejecutar un método privado de otra clase.

IOException, excepciones producidas al realizar tareas de entrada/salida por el programa. Como pueden ser: EOFException, FileNotFoundException, MalformedURLException, SocketException, UnknownHostException, UTFDataFormatException.

NoSuchFieldException, no se encuentra un determinado atributo.

NoSuchMethodException, no se encuentra un determinado método.

RuntimeException, producidos en tiempo de ejecución. Por ejemplo:

ArithmeticException, ClassCastException,
IndexOutOfBoundsException, NegativeArraySizeException,
NullPointerException.

Métodos más utilizados definidos por Throwable	
String getMessage()	Devuelve una descripción de la excepción
Void printStackTrace()	Muestra el rastreo de pila
String toString()	Devuelve un objeto String que contiene una descripción completa de la excepción. Este método se invoca por println() al representar un objeto Throwable

COLECCIONES

boolean add(E e)	Añade un nuevo elemento al final de la lista.
boolean remove(E e):	Elimina la primera ocurrencia del elemento indicado.
boolean contains(E e):	Comprueba si el elemento especificado está en la colección.
void clear():	Elimina todos los elementos de la colección
int size():	Devuelve el número de elementos en la colección.
boolean isEmpty(Collection<?> C)	Comprueba si la colección está vacía.

add()	añadir un elemento
add(numero,elemento)	añade un elemento en la posición indicada
size()	indica el tamaño
get(numero)	devuelve el elemento que se encuentra en esa posición
contains(elemento)	nos indica si ese elemento existe. Devuelve un booleano.
indexOf(elemento)	devuelva la primera ocurrencia donde aparece
lastIndexOf(elemento)	devuelve la última ocurrencia donde aparece
remove(numero)	borra el elemento indicado
remove(elemento)	borra todos los elementos que coincide con el escrito
clear()	borra el ArrayList
isEmpty()	nos indica si está vacío devuelve un true
set(numero,elemento)	modifica el valor de la posición indicada

FICHEROS

➤ FileReader y File Writer

Constructors

Constructor and Description

FileReader(File file)

Creates a new FileReader, given the File to read from.

FileReader(FileDescriptor fd)

Creates a new FileReader, given the FileDescriptor to read from.

FileReader(String fileName)

Creates a new FileReader, given the name of the file to read from.

Constructors

Constructor and Description

FileWriter(File file)

Constructs a FileWriter object given a File object.

FileWriter(File file, boolean append)

Constructs a FileWriter object given a File object.

FileWriter(FileDescriptor fd)

Constructs a FileWriter object associated with a file descriptor.

FileWriter(String fileName)

Constructs a FileWriter object given a file name.

FileWriter(String fileName, boolean append)

Constructs a FileWriter object given a file name with a boolean indicating whether or not to append the data written.

MÉTODOS

int read(): devuelve en ASCII el byte leído. Nos devuelve -1 si no tiene más información el fichero. En algún caso puede que falle el -1 y debemos preguntar por 65535.

int skip(long): salta n caracteres.

S

String readLine(): devuelve toda una línea del fichero.

close(): cierra el stream

void write(caracter): escribe el carácter en el fichero.

- Creación de objeto BufferedReader para capturar datos de un fichero

BufferedReader<nombre_objeto> =new BufferedReader(fichero FileReader);

CONSTRUCTORES :

BufferedReader(objeto Reader);

BufferedReader(objeto Reader, int tamaño);

MÉTODOS

Int read(): leer un carácter del fichero.

String readLine(): leer una línea de texto.

PARA CERRAR EL CANAL

void close(), cierra el fichero.

➤ FileInputStream y FileOutputStream

Constructors

Constructor and Description

FileInputStream(File file)

Creates a FileInputStream by opening a connection to an actual file, the file named by the File object file in the file system.

FileInputStream(FileDescriptor fdObj)

Creates a FileInputStream by using the file descriptor fdObj, which represents an existing connection to an actual file in the file system.

FileInputStream(String name)

Creates a FileInputStream by opening a connection to an actual file, the file named by the path name name in the file system.

Constructors

Constructor and Description

FileOutputStream(File file)

Creates a file output stream to write to the file represented by the specified File object.

FileOutputStream(File file, boolean append)

Creates a file output stream to write to the file represented by the specified File object.

FileOutputStream(FileDescriptor fdObj)

Creates a file output stream to write to the specified file descriptor, which represents an existing connection to an actual file in the file system.

FileOutputStream(String name)

Creates a file output stream to write to the file with the specified name.

FileOutputStream(String name, boolean append)

Creates a file output stream to write to the file with the specified name.

MÉTODOS de la clase InputStream	
Método	Descripción
int available()	Devuelve el número de bytes de entrada disponibles para lectura.
void close()	Cierra el origen de entrada. Los posteriores intentos de lectura generan IOException.
void mark(int numBytes)	Añade una marca al punto actual de flujo de entrada que se valida hasta que se lean numBytes.
boolean markSupported()	Devuelve true si el flujo de invocación admite mark()/reset().
int read()	Devuelve una representación entera del siguiente byte de entrada disponible. Devuelve -1 al llegar al final del flujo.
int read(byte buffer[])	Intenta leer hasta los bytes de buffer.length en búfer y devuelve el número real de bytes leídos correctamente. Se devuelve -1 al llegar al final del flujo.
int read(byte buffer[], int desplazamiento, int numBytes)	Intenta leer hasta los bytes de numBytes en búfer comenzando en búfer[desplazamiento] y devuelve el número de bytes leídos correctamente. Devuelve -1 al llegar al final del flujo.
void reset()	Restablece el puntero de entrada a la marca definida previamente.
long skip(long numBytes)	Ignora los bytes de entrada indicados por numBytes y devuelve el número de bytes ignorados

MÉTODOS de la clase OutputStream	
Método	Descripción
void flush()	Envía a su destino la salida que se haya guardado en búfer. Es decir, vacía el búfer de salida.
void close()	Cierra el origen de salida. Los posteriores intentos de escritura generan IOException.
void write(int b)	Escribe un solo byte en un flujo de salida. El parámetro es int, lo que le permite invocar write() con expresiones sin tener que convertirlas de nuevo a byte.
void write(byte buffer [])	Escribe una matriz completa de bytes en un flujo de salida.
void write(byte buffer[], int desplazamiento, int numBytes)	Escribe un subintervalo de numBytes bytes desde el búfer de matriz, empezando por búfer[desplazamiento]

➤ RandomAccessFile

Constructors

Constructor and Description

RandomAccessFile(File file, String mode)

Creates a random access file stream to read from, and optionally to write to, the file specified by the File argument.

RandomAccessFile(String name, String mode)

Creates a random access file stream to read from, and optionally to write to, a file with the specified name.

- **Metodos:**
 - **Void Seek(long pos)**
 - **long getFilePointer()**
 - **long length()**
 - **Int skipbytes(int d):** desplaza el puntero del fichero las posiciones en bytes que le estamos indicando.

MÉTODOS DE SALIDA	MÉTODOS DE ENTRADA
void writeBoolean(boolean val)	boolean readBoolean()
void writeByte(int val)	byte readByte()
void writeChar(int val)	char readChar()
void writeDouble(double val)	Double readDouble()
void writeFloat(float val)	float readFloat()
void writeInt(int val)	Int readInt()
void writeLong(float val)	long readLong()
void writeShort(float val)	short readShort()