**Ficheros de acceso aleatorio**

Dos nuevos constructores:

RandomAccessFile (String fichero, String modoAcceso)

RandomAccessFile (File f, String modoAcceso)

Modo de Acceso:

r (solo lectura),

rw (lectura y escritura)

**Métodos RandomAccessFile**

* **long getFilePointer()** Devuelve posición actual del puntero del fichero
* **void seek(long posicion)** Coloca puntero del fichero en posición determinada, empezando por el comienzo
* **long length()** Devuelve tamaño del fichero en bytes. Posición length() marca EOF
* **int skipBytes(intdesplazamiento)** Desplaza puntero desde posición actual el número de bytes indicados en desplazamiento

**Métodos RandomAccessFile de lectura y de escritura**

* **writeInt(int), writeLong(long), writeDouble(double), writeBytes(String), writeChars(String), writeBoolean(boolean), raf.writeChar(char), …**
* **readInt(),readLong(),readChar(), readDouble(),readBoolean(),  …**

**EJERCICIO DE FICHEROS DE ACCESO ALEATORIO**

Crea un fichero al que vamos a añadir cinco productos que habremos cargado previamente en un ArrayList

Cada producto tendrá:

* un identificador (id): entero
* un nombre String [10]
* un precio double
* un descuento booleano
* un tipo char

ArrayList<Producto> productos = new ArrayList<>();

productos.add(new Producto(1, "Producto 1", 10.5, true, 'T'));

…

Abriremos el fichero de Acceso Aleatorio

try (RandomAccessFile raf = new RandomAccessFile("c:\\pruebas\\ejemplo\_raf.dat", "rw")) {

**Escribiremos** cada uno de los campos de cada producto en el fichero.

Observa como:

* Tenemos un método diferente en función del tipo de elemento que vayas a escribir en el fichero
* Los Strings los tenemos que convertir en un StringBuilder para que tenga un tamaño determinado, en nuestro caso 10 caracteres

for (Producto p : productos) {

raf.writeInt(p.getId());

StringBuilder sb = new StringBuilder(p.getNombre());

sb.setLength(10);

raf.writeChars(sb.toString());

raf.writeDouble(p.getPrecio());

raf.writeBoolean(p.isDescuento());

raf.writeChar(p.getTipo());

}

Ahora vamos a leer un registro, por ejemplo el tercero:

Cada registro ocupa 35 bytes: un identificador entero (4 bytes), un nombre String de 10 caracteres (2 bytes cada carácter 20 bytes) , un precio double (8 bytes), un descuento booleano (1 byte) y un tipo char (2 bytes). **Total: 35 bytes**

Crea en la clase Producto un método length que devuelva el tamaño en bytes de todos los campos del registro

Posicionamos el cursor en la posición donde comienza el tercer registro:

* El primero empieza en la posición cero
* el segundo en la 35 (tamaño del registro)
* El tercero en la 35x2.
* ….
* Por lo que crea el método que te devuelva la posición exacta donde debes posicionar el cursor

raf.seek(**posición calculada**);

Y comenzamos a leer:

System.out.println(**raf.readInt()**);

String nombre = "";

for (int i = 0; i < 10; i++) {

nombre += **raf.readChar()**;

}

System.out.println(nombre);

System.out.println(**raf.readDouble()**);

System.out.println(**raf.readBoolean()**);

System.out.println(**raf.readChar()**);

**Práctica**

Vamos a añadir a los productos del ejercicio anterior un campo booleano después del identificador llamado ***eliminado***, que se marcará con *true* cuando se pretenda eliminar el producto.

Debemos tener el siguiente menú:

**PRODUCTOS:**

1. **Mostrar todos** (mostraría todos los que no están eliminados)
2. **Mostrar por ID** (buscaría uno por un ID concreto y los mostraría si existe y no está eliminado)
3. **Añadir** (se añadiría al final del fichero)
4. **Modificar** (habría que indicar el id)
5. **Eliminar** (habría que indicar el id)
6. **Salir**

Extra:

Añadimos al menú las opciones:

* Limpiar eliminados (Se eliminarán definitivamente todos los registros marcados como eliminados)
* Ordenar (Se ordenaría el fichero por Id)