

Prácticas de Ficheros

Sebastián Leonte

Practica 1.1 – Borrado Recursivo

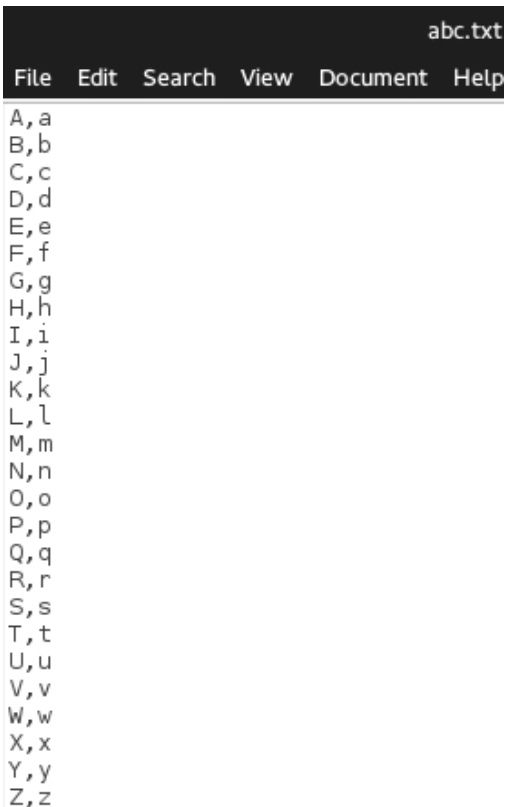
```
public void borrarDirRecursivo(File ficheroActual) {  
    File[] lista = ficheroActual.listFiles();  
    for (File f : lista) {  
        if (f.isFile()) {  
            f.delete();  
            System.out.println("Borrando archivo: " + f.toString());  
        } else {  
            borrarDirRecursivo(f);  
        }  
    }  
    ficheroActual.delete();  
    System.out.println("Borrando carpeta: " + ficheroActual.toString());  
}
```

```
-----  
Borrando archivo: C:\holo\1\2\2.txt  
Borrando carpeta: C:\holo\1\2|  
Borrando carpeta: C:\holo\1\Nueva carpeta  
Borrando carpeta: C:\holo\1  
Borrando carpeta: C:\holo  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ficheros de texto.

Alfabeto.txt

```
public void abecedario() throws IOException {
    System.out.println("Indique la ruta del fichero a escribir: ");
    String ruta = in.nextLine();
    File fichero = new File(ruta);
    fichero.createNewFile();
    if (fichero.exists() && fichero.isFile()) {
        PrintWriter printWriter = null;
        try {
            printWriter = new PrintWriter(new BufferedWriter(new FileWriter(fichero)));
            for (char s = 'A'; s <= 'Z'; s++) {
                printWriter.println(s + "," + Character.toLowerCase(s));
            }
            printWriter.close();
        } catch (FileNotFoundException fnfe) {
        } catch (IOException ioe) {
        }
    } else {
        System.out.println("El fichero no existe o no es un fichero");
    }
}
```



The screenshot shows a text editor window titled 'abc.txt'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Document', and 'Help'. The text area contains the alphabet from 'A,a' to 'Z,z', with each letter and its lowercase counterpart separated by a comma and a space, and each pair on a new line.

```
A,a
B,b
C,c
D,d
E,e
F,f
G,g
H,h
I,i
J,j
K,k
L,l
M,m
N,n
O,o
P,p
Q,q
R,r
S,s
T,t
U,u
V,v
W,w
X,x
Y,y
Z,z
```

Acceso directo a un fichero

```
public void modificarUnEmpleado(int numEmpleado) {
    System.out.println("Indique el nombre del fichero de donde leer el Empleado: ");
    File file = new File(in.nextLine() + ".bin");
    try {
        RandomAccessFile randomAccessFile = new RandomAccessFile(file, "rw");
        int longitudFichero = (int) randomAccessFile.length();
        if ((numEmpleado * 16) <= longitudFichero) {
            int pos = (numEmpleado - 1) * 16;
            randomAccessFile.seek(pos);
            System.out.println("Datos Antiguos:");
            System.out.println("Numero de empleado: " + randomAccessFile.readInt());
            System.out.println("Departamento: " + randomAccessFile.readInt());
            System.out.println("Salario: " + randomAccessFile.readDouble());
            System.out.println("Indique el departamento: ");
            int departamento = Integer.parseInt(in.nextLine());
            System.out.println("Indique el salario: ");
            double salario = Double.parseDouble(in.nextLine());
            randomAccessFile.seek(pos);
            randomAccessFile.writeInt(numEmpleado);
            randomAccessFile.writeInt(departamento);
            randomAccessFile.writeDouble(salario);
            System.out.println("Datos Nuevos:");
            System.out.println("Numero de empleado: " + numEmpleado);
            System.out.println("Departamento: " + departamento);
            System.out.println("Salario: " + salario);
        } else {
            System.out.println("No existe ese empleado");
        }
    }
}
```

Indique el nombre del fichero de donde leer el Empleado:

Empleados

Datos Antiguos:

Numero de empleado: 1

Departamento: 1

Salario: 25.0

Indique el departamento:

2

Indique el salario:

50

Datos Nuevos:

Numero de empleado: 1

Departamento: 2

Salario: 50.0

BUILD SUCCESSFUL (total time: 28 seconds)

Volcados de Fichero de objetos.

```
public ArrayList pasarAArrayList() {
    ArrayList<Persona> arrayList = new ArrayList<>();

    ObjectInputStream objectInputStream = null;
    try{
        objectInputStream = new ObjectInputStream(new FileInputStream(file));
        while(true){
            Persona persona = (Persona)objectInputStream.readObject();
            arrayList.add(persona);
        }
    } catch (FileNotFoundException e) {

    } catch (EOFException eofe){

    } catch (IOException | ClassNotFoundException e) {

    } finally {
        return arrayList;
    }
}

public void arraylistAFichero(ArrayList<Persona> arrayList){
    ObjectOutputStream objectOutputStream = null;
    try{
        objectOutputStream = new ObjectOutputStream(new FileOutputStream(file));
        for(Persona p: arrayList){
            objectOutputStream.writeObject(p);
        }
        objectOutputStream.close();
    } catch (FileNotFoundException e) {

    } catch (IOException e) {

    }
}
```