

SOLUCIONES EJERCICIOS DE FICHEROS

Problema 1

Explica qué diferencia hay entre las clases InputStream/OutputStream y Reader/Writer

InputStream es una clase abstracta derivada de la clase "Object", y sirve para definir como leer datos sin que importe su origen
OutputStream también es una clase abstracta derivada de la clase "Object", pero en este caso, es para escribir.
Reader es una clase abstracta derivada de la clase "Object" que define como leer el flujo de caracteres, Writer también es una clase abstracta derivada de la clase "Object", sirve para suministrar una buena cantidad de métodos para crear, escribir y procesar el flujo de caracteres de salida.

Problema 2

¿Qué es la codificación de caracteres?

Método que permite convertir un carácter del alfabeto en un símbolo de otro sistema de representación. Un ejemplo claro, sería el código ASCII.

Problema 3

Haz un programa que lea el contenido de un fichero "hola.txt" y lo muestre por pantalla. Nota: usa Reader

```
public static void main(String[] args) {  
  
    try {  
        String path = "C:\\Users\\eduardo\\Desktop\\crebas.sql";  
        File pepe = new File(path);  
        BufferedReader miArchivo = new BufferedReader(new FileReader(pepe));  
        String ln = miArchivo.readLine();  
        while (ln != null) {  
            System.out.println(ln);  
            ln = miArchivo.readLine();  
        }  
    } catch (IOException e) {  
        System.err.println("Excepción " + e.getMessage());  
    }  
}
```

NOTA: Realiza la lectura del fichero línea a línea

Problema 4

Tomando el ejercicio anterior, amplíalo para que después de mostrarlo, pregunte al usuario si desea invertir el texto. Si el usuario responde "sí", deberemos reescribir hola.txt con el texto escrito al revés. Nota: usa Writer.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    BufferedReader entrada;
    FileWriter salida;
    String ln = null;
    StringBuilder sb=new StringBuilder();
    try {
        File ent = new File("C:\\Users\\eduardo\\Desktop\\crebas.sql");
        entrada = new BufferedReader(new FileReader(ent));
        File sal = new File("C:\\Users\\eduardo\\Desktop\\sal.sql");
        salida = new FileWriter(sal);

        ln = entrada.readLine();
        while (ln != null){
            System.out.println(ln);
            ln = entrada.readLine();
            if (ln!=null) {
                System.out.println(ln);
                sb.append(ln);
            }
        }
        System.out.print("¿Quieres invertir el texto? 1-Si, 2-No: ");
        int respuesta = sc.nextInt();
        if (respuesta == 1)
            ln = sb.reverse().toString();
        else
            ln = sb.toString();
        System.out.print(ln);
        salida.write(ln);
        salida.close();
    } catch (IOException e) {
        System.err.println("Excepción " + e.getMessage());
    }
}
```

Problema 5

Haz un programa que sirva para duplicar ficheros. El usuario nos pasará la ruta al fichero a copiar, y nosotros lo copiaremos como nombre + ".copia". Por ejemplo, c:\aaa\fichero.txt lo duplicaremos en c:\aaa\fichero.txt.copia. Nota: Usa InputStream/OutputStream.

```
public static void main(String[] args) {
    //declaració de classes utilitzades
    InputStream miArchivoIn;
    OutputStream miArchivoOut;
    try{
        String filein = "C:\\Users\\eduardo\\Desktop\\crebas.sql";
        miArchivoIn = new FileInputStream(new File(filein));
        String fileout = "C:\\Users\\eduardo\\Desktop\\sal.sql";
        miArchivoOut = new FileOutputStream(new File(fileout));
        int i = miArchivoIn.read();
        while(i != -1){
            miArchivoOut.write(i);
            i = miArchivoIn.read();
        }
        miArchivoOut.close();
        miArchivoIn.close();

        System.out.println("El archivo se ha duplicado:");
    }catch (IOException ex){
        System.err.println("Excepción " + ex.getMessage());
    }
}
```

Problema 6

Haz un programa que nos muestre el contenido de una carpeta, marcando con <DIR> las subcarpetas. Pide al usuario la ruta absoluta a esa carpeta, y guárdala en un string.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Escriu la ruta del fixer a copiar: ");
    String rutaIn = sc.next();
    File dir = new File(rutaIn);
    String[] ficheros = dir.list();
    if (ficheros == null)
        System.out.println("No hay ficheros en el directorio"
            + " especificado");
    else {
        for (int x=0; x<ficheros.length;x++){
            File pepe = new File(rutaIn+"\\")+ficheros[x]);
            if (pepe.isDirectory())
                System.out.println("<DIR> " + ficheros[x]);
            else
                System.out.println("<FILE> " + ficheros[x]);
        }
    }
}
```

Problema 7

Utilizando la siguiente relación de clases, donde tanto Moto como Coche derivan de la clase base Vehiculo:

```
public class Vehiculo {
    public String color;
    public int caballos;
    public String marca;
    public String modelo;

    public Vehiculo () {
        this.color="rojo";
        this.caballos=300;
        this.marca="Ferrari";
        this.modelo="F15";
    }
}

public class Moto extends Vehiculo {
    private int precio;
    private String descripcion;
    public Moto () {
        precio=1000;
        descripcion="Es una moto";
    }
    public int getPrecio() {
        return precio;
    }
    public String getDescripcion() {
        return descripcion;
    }
}

public class Coche extends Vehiculo {
    private int numPuertas;
    private int capacidadMaletero;

    public Coche () {
        this.numPuertas=4;
        this.capacidadMaletero=100;
    }
}
```

Crea un array de 5 Vehículos que permita al alumno decidir qué tipo de elemento inserta en cada posición (de la clase Moto o Coche). Añádele el código necesario para serializar en un fichero los datos que el usuario introduce.

```
public static void main(String[] args) {
    Vehiculo vhl[] = new Vehiculo [5];
    Scanner sc = new Scanner(System.in);
    for (int x = 0; x < 5; x++) {
        System.out.print("1.Moto o 2.Coche? ");
        int opcion = sc.nextInt();
        if (opcion == 1){
            Moto mimoto = new Moto();
            vhl[x] = (Vehiculo)mimoto;
        } else{
            Coche micoche = new Coche();
            vhl[x] = (Vehiculo)micoche;
        }
    }
    try{
        String archivo = "C:\\prueba.txt";
        FileOutputStream fileo = new FileOutputStream(archivo);
        ObjectOutputStream oos = new ObjectOutputStream(fileo);
        for (int x = 0; vhl.length > x; x++)
            oos.writeObject(vhl[x]);
        System.out.println("El vector de vehiculos se ha serializado en el archivo " + archivo);
        oos.close();
    }catch(IOException ex){
        System.out.print(ex.getMessage());
    }
}
```

```
public class Vehiculo implements Serializable {
    public String color;
    public int caballos;
    public String marca;
    public String modelo;

    public Vehiculo () {
        this.color="rojo";
        this.caballos=300;
        this.marca="Ferrari";
        this.modelo="F15";
    }
}
```

Problema 8

En un proyecto diferente al del ejercicio 7, lee los datos serializados por el ejercicio anterior y muéstralos por pantalla.

NOTA: El único requisito para deserializar un objeto desde un archivo, consiste en utilizar el mismo nombre del package del programa con el que se serializaron los datos, puesto que este nombre queda grabado junto con la información serializada en el fichero.

```
public static void main(String[] args) {
    try{
        String archivo = "C:\\prueba.txt";
        FileInputStream arch = new FileInputStream(archivo);
        ObjectInputStream ois = new ObjectInputStream(arch);
        Vehiculo pepe[] = new Vehiculo [5];
        System.out.println("Deserializamos la información del archivo " + archivo);

        for (int x = 0; x<pepe.length; x++) {
            pepe[x] = (Vehiculo)ois.readObject();
        }
        ois.close();
        arch.close();
        for (int x = 0; x<pepe.length; x++) {
            if (pepe[x] instanceof Moto)
                System.out.println("Objeto deserializado " + x + " se trata de una moto");
            else
                System.out.println("Objeto deserializado " + x + " se trata de un coche");
        }
    }catch(IOException ioex){
        System.out.print(ioex.getMessage());
    }catch(ClassNotFoundException curro){
        System.out.print(curro.getMessage());
    }
}
```

Problema 9

Haz un programa que lea varios ficheros de texto, y cree un fichero resultado con el contenido de todos los anteriores. El usuario debe poder especificar qué ficheros desea incluir, y cómo se llamará el fichero final.

```
public static void main(String[] args) {
    BufferedReader miArchivo;
    java.util.Scanner sc = new java.util.Scanner(System.in);
    FileWriter miSalida;
    String path = "C:\\\\";
    StringBuilder sb = new StringBuilder();
    try {
        System.out.print("Indica un fichero a agregar (-1 para finalizar): ");
        String fitxer = sc.next();
        while (!fitxer.equals("-1")) {
            fitxer = path + fitxer;
            miArchivo = new BufferedReader(new FileReader(fitxer));
            sb.append(miArchivo.readLine());
            sb.append(" ");
            System.out.print("Indica un fichero a agregar (-1 para finalizar): ");
            fitxer = sc.next();
        }
        System.out.println("Información leída: \n" + sb.toString());
        System.out.print("Indica el nombre del fichero donde guardar la información: ");
        String respuesta = sc.next();
        miSalida = new FileWriter(path + respuesta);
        miSalida.write(sb.toString());
        miSalida.close();
    } catch (IOException e) {
        System.err.println("Error " + e.getMessage());
    }
}
```