

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.mensajes;
```

```
/**
```

```
 *
```

```
 * @author brc-9
```

```
 */
```

```
public interface IEncriptable {
```

```
    public String encriptar(String mensaje);
```

```
    public String desEncriptar();
```

```
}
```

```
package Examen.mensajes;

import java.io.Serializable;

/**
 *
 * @author brc-9
 */
public abstract class Mensaje implements Serializable {
    protected int codigoNumerico;
    protected String mensaje;
    static int contador;

    public int getCodigoNumerico() {
        return codigoNumerico;
    }

    public void setCodigoNumerico(int codigoNumerico) {
        this.codigoNumerico = codigoNumerico;
    }

    public String getMensaje() {
        return mensaje;
    }

    public void setMensaje(String mensaje) {
        this.mensaje = mensaje;
    }

    public Mensaje(String mensaje) {
        contador++;
        this.codigoNumerico = contador;
        this.mensaje = mensaje;
    }
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Examen.mensajes;
```

```
public class MensajeEncriptado extends Mensaje implements IEncriptable {
```

```
    public MensajeEncriptado(String mensaje) {  
        super(mensaje);  
        this.mensaje = encriptar(mensaje);  
    }
```

```
    @Override  
    public String encriptar(String mensaje) {  
        String mensajeEncrip = "";  
        for (int i = 0; i < mensaje.length(); i++) {  
            mensajeEncrip += Character  
                .toString((char) (mensaje.codePointAt(i) + 1));  
        }  
        return mensajeEncrip;  
    }
```

```
    @Override  
    public String desEncriptar() {  
        String mensajeDesEncrip = "";  
        for (int i = 0; i < mensaje.length(); i++) {  
            mensajeDesEncrip += Character  
                .toString((char) (mensaje.codePointAt(i) - 1));  
        }  
        return mensajeDesEncrip;  
    }
```

```
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Examen.mensajes;
```

```
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.ObjectInputStream;  
import java.io.ObjectOutputStream;  
import java.util.ArrayList;
```

```
/**  
 *  
 * @author brc-9  
 */
```

```
public class Mensajeria {
```

```
    static String rutaFichero = "mensajes.dat";  
    static ArrayList<Mensaje> listaMensajes = null;
```

```
    public static void main(String[] args) {  
        leerFicheroBinario();  
        recuperarContador();
```

```
        int opcionMenu = menu();
```

```
        while (opcionMenu != 8) {  
            switch (opcionMenu) {  
                case 1 :  
                    listarTodosMensajes();  
                    break;  
                case 2 :  
                    listarNormales();  
                    break;  
                case 3 :  
                    listarExcriptados();  
                    break;  
                case 4 :  
                    buscarMensajes();  
                    break;  
                case 5 :  
                    anadirMensajes();  
                    break;
```

```

        case 6 :
            desenscriptarMensajes();
            break;
        case 7 :
            eliminarMensajes();
            break;
    }
    opcionMenu = menu();
}

escribirArchivoBinario();
}

private static void leerFicheroBinario() {
    File fichero = new File(rutaFichero);
    if (fichero.exists()) {
        ObjectInputStream objInputStream = null;
        try {
            objInputStream = new ObjectInputStream(
                new FileInputStream(fichero));
            listaMensajes = (ArrayList<Mensaje>) objInputStream
                .readObject();
        } catch (IOException e) {
            System.err.println("Error abriendo el fichero");
        } catch (ClassNotFoundException e) {
            System.err.println("Formato incorrecto de fichero");
        } finally {
            try {
                if (objInputStream != null)
                    objInputStream.close();
            } catch (IOException e) {
                System.err.println("Error cerrando el fichero");
            }
        }
    } else {
        listaMensajes = new ArrayList<Mensaje>();
    }
}

private static void escribirArchivoBinario() {
    File fichero = new File(rutaFichero);
    ObjectOutputStream objOutputStream = null;
    try {
        objOutputStream = new ObjectOutputStream(
            new FileOutputStream(fichero));
        objOutputStream.writeObject(listaMensajes);
    } catch (IOException e) {
        System.err.println("Error abriendo el fichero");
    } finally {

```

```

        try {
            if (objOutputStream != null)
                objOutputStream.close();
        } catch (IOException e) {
            System.err.println("Error cerrando el fichero");
        }
    }
}

private static int menu() {
    String textoMenu = "1. Listar todos los mensajes \n"
        + "2. Mostramos mensajes normales \n"
        + "3. Mostramos mensajes encriptados \n" + "4. Buscar mensaje \n"
        + "5. Añadir mensaje \n" + "6. Desencriptar mensaje \n"
        + "7. Eliminar mensaje \n" + "8. Salir \n";
    int opcionSeleccionada;
    opcionSeleccionada = Teclado.leerInt(textoMenu, 1, 8);
    return opcionSeleccionada;
}

private static void recuperarContador() {
    if (!listaMensajes.isEmpty()) {
        Mensaje.contador = listaMensajes.get(listaMensajes.size() - 1)
            .getCodigoNumerico();
    }
}

private static void imprimirMensaje() {
    String rutaFicheroTexto;
    rutaFicheroTexto = Teclado.leerTexto("Indica el nombre del fichero");
    rutaFicheroTexto += ".txt";

    File ficheroTexto = new File(rutaFicheroTexto);
    BufferedWriter buffWriter = null;
    try {
        buffWriter = new BufferedWriter(new FileWriter(ficheroTexto));
        buffWriter.write(listarMensajes());
        System.out.println("Fichero de texto generado correctamente");
    } catch (IOException e) {
        System.err.println("Error escribiendo el fichero " + rutaFichero);
    } finally {
        try {
            if (buffWriter != null)
                buffWriter.close();
        } catch (IOException e) {
            System.err.println("Error cerrando el fichero.");
        }
    }
}

```

```

}
private static String listarMensajes() {
    String mensajes = "";
    for (Mensaje mens : listaMensajes) {
        mensajes += "*****\nCodigo: " + mens.getCodigoNumerico()
            + "\nMensaje: " + mens.getMensaje() + "\n*****";
    }
    return mensajes;
}

```

```

private static void listarTodosMensajes() {
    String opcionElegida = "";
    boolean opcionElegidaValida = false;
    String mensaje;

    while (!opcionElegidaValida) {
        opcionElegida = Teclado.leerTexto(
            "¿Que quieres hacer? \n Listar: 'l' \nImprimir: 'i'");
        if (opcionElegida.equalsIgnoreCase("l")
            || opcionElegida.equalsIgnoreCase("i")) {
            opcionElegidaValida = true;
        }
    }
    if (opcionElegida.equalsIgnoreCase("l")) {
        for (Mensaje mens : listaMensajes) {
            System.out.println("*****\nCodigo: "
                + mens.getCodigoNumerico() + "\nMensaje: "
                + mens.getMensaje() + "\n*****");
        }
    } else if (opcionElegida.equalsIgnoreCase("i")) {
        imprimirMensaje();
    }
}

```

```

private static void listarNormales() {
    for (Mensaje mensaje : listaMensajes) {
        if (mensaje instanceof MensajeSinEncriptar) {
            System.out.println("*****\nCodigo: "
                + mensaje.getCodigoNumerico() + "\nMensaje: "
                + mensaje.getMensaje() + "\n*****");
        }
    }
}

```

```

private static void listarExcriptados() {
    for (Mensaje mensaje : listaMensajes) {
        if (mensaje instanceof MensajeEncriptado) {
            System.out.println("*****\nCodigo: "
                + mensaje.getCodigoNumerico() + "\nMensaje: "

```

```

        + mensaje.getMensaje() + "\n*****");
    }
}

private static void buscarMensajes() {
    int codigoBuscado;
    codigoBuscado = Teclado.leerInt("Introduce el codigo a buscar");

    for (int i = 0; i < listaMensajes.size(); i++) {
        if (listaMensajes.get(i).getCodigoNumerico() == codigoBuscado) {
            System.out.println("*****\nCodigo: "
                + listaMensajes.get(i).getCodigoNumerico()
                + "\nMensaje: " + listaMensajes.get(i).getMensaje()
                + "\n*****");
            break;
        } else if (i == listaMensajes.size() - 1) {
            System.out.println("El codigo no existe");
        }
    }
}

/*
 * for (Mensaje mensaje : listaMensajes) {
 * if(mensaje.getCodigoNumerico() == codigoBuscado){
 * System.out.println("*****\nCodigo: " +
 * mensaje.getCodigoNumerico() + "\nMensaje: " + mensaje.getMensaje() +
 * "\n*****"); break; }else{
 * System.out.println("El codigo no existe"); }
 * }
 */
}

```

```

private static void anadirMensajes() {
    String tipoMensaje = "";
    boolean tipoMensajeValido = false;
    String mensaje;
    Mensaje nuevoMensaje = null;
    while (!tipoMensajeValido) {
        tipoMensaje = Teclado.leerTexto(
            "¿Que tipo de mensaje quieres crear? \n Normal: 'n'
\nEncriptado: 'e'");
        if (tipoMensaje.equalsIgnoreCase("n")
            || tipoMensaje.equalsIgnoreCase("e")) {
            tipoMensajeValido = true;
        }
    }
    if (tipoMensaje.equalsIgnoreCase("n")) {
        mensaje = Teclado.leerTexto("Introduce el mensaje");
        nuevoMensaje = new MensajeSinEncriptar(mensaje);
    }
}

```



```

        listaMensajes.add(nuevoMensaje);
    } else if (tipoMensaje.equalsIgnoreCase("e")) {
        mensaje = Teclado.leerTexto("Introduce el mensaje");
        nuevoMensaje = new MensajeEncriptado(mensaje);
        listaMensajes.add(nuevoMensaje);
    }
}

private static void desencriptarMensajes() {
    int codigoBuscado;
    codigoBuscado = Teclado.leerInt("Introduce el codigo a buscar");

    for (int i = 0; i < listaMensajes.size(); i++) {
        if (listaMensajes.get(i).getCodigoNumerico() == codigoBuscado
            && listaMensajes.get(i) instanceof MensajeEncriptado) {
            System.out.println("*****" + "\nCodigo: "
                + listaMensajes.get(i).getCodigoNumerico()
                + "\nMensaje: " + listaMensajes.get(i).getMensaje()
                + "\nMensaje desencriptado: "
                + ((MensajeEncriptado) listaMensajes.get(i))
                    .desEncriptar()
                + "\n*****");
            break;
        } else if (i == listaMensajes.size() - 1) {
            System.out.println(
                "El codigo no existe o no es un mensaje encriptado.");
        }
    }
}

private static void eliminarMensajes() {
    int codigoBuscado;
    codigoBuscado = Teclado.leerInt("Introduce el codigo a eliminar");
    boolean codigoBorrado = false;
    int indiceBorrado = 0;

    for (int i = 0; i < listaMensajes.size(); i++) {
        if (listaMensajes.get(i).getCodigoNumerico() == codigoBuscado) {
            listaMensajes.remove(i);

            indiceBorrado = i;
            codigoBorrado = true;
        } else if (i == listaMensajes.size() - 1) {
            System.out.println("El codigo no existe.");
        }
    }
    System.out.println(listaMensajes.size());
    if (codigoBorrado) {

```

```
for (int j = indiceBorrado; j < listaMensajes.size(); j++) {  
    listaMensajes.get(j).setCodigoNumerico(  
        listaMensajes.get(j).getCodigoNumerico() - 1);  
}  
}  
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Examen.mensajes;

/**
 *
 * @author brc-9
 */
public class MensajeSinEncriptar extends Mensaje {

    public MensajeSinEncriptar(String mensaje) {
        super(mensaje);
    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.mensajes;
```

```
import java.util.Scanner;
```

```
/**
```

```
*
```

```
* @author brc-9
```

```
*/
```

```
public class Teclado {
    static Scanner sc = new Scanner(System.in);

    public static int leerInt(String mensaje, int min, int max) {
        int numero = 0;
        boolean respuestValida = true;
        System.out.println(mensaje);
        while (respuestValida) {
            try {
                numero = sc.nextInt();
                if (numero >= min && numero <= max) {
                    respuestValida = false;
                } else {
                    System.err.println(
                        "Introduce un numero entre " + min + " y " + max);
                }
            } catch (Exception e) {
                sc.nextLine();
                System.err.println("Introduce un numero entero");
            }
        }
        return numero;
    }
}
```

```
public static int leerInt(String mensaje) {
    int numero = 0;
    boolean respuestValida = true;
    System.out.println(mensaje);
    while (respuestValida) {
        try {
            numero = sc.nextInt();
            respuestValida = false;
        } catch (Exception e) {
            sc.nextLine();
            System.err.println("Introduce un numero entero");
        }
    }
}
```

```
        }  
    }  
    return numero;  
}  
  
public static String leerTexto(String mensaje) {  
    String texto = "";  
    System.out.println(mensaje);  
    texto = sc.next();  
    return texto;  
}  
  
}
```