Se desea realizar un programa que sea capaz de gestionar el manejo de mensajes. Un mensaje está definido por un código numérico y un texto que es el mensaje en sí. Tendremos dos tipos de mensajes: mensajes sin encriptar y mensajes encriptados. Mensajes sin encriptar: Los mensajes se almacenan tal y como los escribe el usuario. Mensajes encriptados: Los mensajes se almacenan encriptados de forma que cada carácter se almacena sumándole uno a su código ASCII, es decir, cada uno de los caracteres que forman el mensaje será sustituido por el carácter resultante de sumarle uno al char (c = c+1 siendo c de tipo char). Los mensajes encriptados implementarán métodos para encriptar y desencriptar los mensajes y para ello, implementarán la interfaz IEncriptable que define los métodos encriptar y desencriptar. Tendremos además una clase llamada Mensajería que tendrá el método main y, como atributo tendrá una lista de tamaño variable con todos los mensajes (encriptados y no encriptados). Al iniciar el programa se mostrará un menú al usuario con las siguientes opciones: 1. Listar todos los mensajes: muestra todos los mensajes de la lista. Esta opción nos dará la oportunidad de elegir si se quiere mostrar los mensajes por pantalla o guardarlos en un fichero de texto, cuyo nombre será pedido al usuario. 2. Mostrar mensajes normales: muestra todos los mensajes no encriptados de la lista 3. Mostrar mensajes encriptados: se mostraran los mensajes encriptados sin desencriptar, tal y como están almacenados 4. Buscar mensaje: buscará un mensaje a partir de un código, si no existe se lo indicará al usuario 5. Añadir mensaje: preguntará al usuario que tipo de mensaje desea añadir (encriptado o normal) y lo añadirá a la lista 6. Desencriptar mensaje: mostrará un mensaje, desencriptándolo primero. Si el mensaje no existe o no está encriptado lo indicará por pantalla. 7. Eliminar mensaje: eliminará un mensaje a partir de un código. Si no existe el mensaje lo indicará. 8. Salir: sale de la aplicación. Los mensajes deben tener como código la posición que ocupan dentro de la lista por lo que cada vez que eliminemos un mensaje deberemos reorganizar los códigos. Cuando arrancamos el programa se comprobará si existe el fichero mensajes.dat que contiene los mensajes. Si existe, los carga en la lista. Al salir del programa se guardará en el fichero mensajes.dat la lista de mensajes en formato binario

```
* 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 exordinario1819;

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

/**
 *
 * @author anusk
 */
public class Mensajeria {

    public static ArrayList<Mensaje> mensajes = new ArrayList<Mensaje>();
    public static String fMensajes = "mensajes.dat";



    public static int leerInt()
    {
        Scanner sc = new Scanner(System.in);
        while (true)
        {
            try
            {
                return sc.nextInt();
            }
            catch (Exception e)
```

```
        {
            sc.nextLine();
        }
    }
}

public static int menu()
{

    int op = 0;
    while (op<1 || op >8)
    {
        System.out.println("1. Listar todos los mensajes");
        System.out.println("2. Mostrar mensajes normales");
        System.out.println("3. Mostrar mensajes encriptados");
        System.out.println("4. Buscar mensaje");
        System.out.println("5. AÃ±adir mensaje");
        System.out.println("6. Desencriptar mensaje");
        System.out.println("7. Eliminar mensaje");
        System.out.println("8. Salir");

        op = leerInt();
    }
    return op;


}

public static void listarMensajes()
{
```

```
int op = 0;
while (op!=1 && op!=2)
{
   System.out.println("Â¿Desea mostrar los mensajes por pantalla o guardarlos en fichero de texto?");
   System.out.println("1. Por pantalla");
   System.out.println("2. Guardar en fichero");
   op = leerInt();
}
if (op == 1)
{
   Iterator <Mensaje> it = mensajes.iterator();
   while (it.hasNext())
   {
      System.out.println(it.next());
   }
}
else
{
   Scanner sc = new Scanner(System.in);
   String fichero="";
   System.out.println("Indica el nombre del fichero");
   fichero = sc.nextLine();
   BufferedWriter bw = null;
   try
   {
      bw = new BufferedWriter(new FileWriter(fichero));
      Iterator <Mensaje> it = mensajes.iterator();
      while (it.hasNext())
      {
         bw.write(it.next().toString());
```

```
                bw.newLine();
            }

        }
        catch (IOException e)
        {
            System.out.println("Error escribiendo el fichero de texto");
        }
        finally
        {
            try
            {
                if (bw!= null)
                    bw.close();
            }
            catch (IOException e)
            {
                System.out.println("Error cerrando el fichero");
            }
        }
    }

}

public static void listarMensajesNormales()
{
    Iterator <Mensaje> it = mensajes.iterator();
    while (it.hasNext())
    {
        Mensaje m = it.next();
        if (m instanceof MensajeSinEncriptar)
```

```
        {
            System.out.println(m);
        }

    }
}

public static void listarEncriptado()
{
    Iterator <Mensaje> it = mensajes.iterator();
    while (it.hasNext())
    {
        Mensaje m = it.next();
        if (m instanceof MensajeEncriptado)
        {
            System.out.println(m);
        }

    }
}

public static Mensaje buscar(int c)
{
    Iterator <Mensaje> it = mensajes.iterator();
    while (it.hasNext())
    {
        Mensaje m = it.next();
        if (m.getCodigo() == c)
        {
            return m;
        }
```

```
    }
    return null;
}

public static void buscarMensaje()
{
    System.out.print("CÃ³digo: ");
    int cod = leerInt();
    Mensaje m = buscar(cod);
    if (m == null)
        System.out.println("No existe ningÃºn mensaje con el cÃ³digo "+cod);
    else
        System.out.println(m);
}

public static void anadirMensaje()
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Mensaje: ");
    String texto = sc.nextLine();
    int op = 0;
    while (op!=1 && op!=2)
    {
        System.out.println("Â¿Desea encriptar su mensaje?\n1. SÃ-\n2.No");
        op = leerInt();
    }
    if (op == 1)
        mensajes.add(new MensajeEncriptado(mensajes.size(),texto));
    else
        mensajes.add(new MensajeSinEncriptar(mensajes.size(),texto));
}
```

```
public static void desencriptarMensaje()
{
   System.out.print("CÃ³digo del mensaje a desencriptar: ");
   int cod = leerInt();
   Mensaje m = buscar(cod);
   if (m == null)
      System.out.println("Mensaje no encontrado");
   else if (m instanceof MensajeEncriptado)
      ((MensajeEncriptado)m).mostrarDesencriptado();
   else
      System.out.println("El mensaje con cÃ³digo "+cod+" no es un mensaje encriptado");
}

public static void eliminarMensaje()
{
   System.out.print("CÃ³digo del mensaje a eliminar: ");
   int cod = leerInt();
   Mensaje m = buscar(cod);
   if (m == null)
      System.out.println("Mensaje no encontrado");
   else
   {
      mensajes.remove(cod);
      for (int i = cod; i < mensajes.size(); i++)
         mensajes.get(i).setCodigo(i);
   }
}

public static void guardarMensajes()
{
```

```
    ObjectOutputStream oos = null;
    try
    {
        oos = new ObjectOutputStream(new FileOutputStream(fMensajes));
        oos.writeObject(mensajes);
    }
    catch (IOException e)
    {
        System.out.println("Error guardando la lista de mensajes en el fichero");
    }
    finally
    {
        try
        {
            if (oos!=null)
                oos.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

public static void recuperarMensajes()
{
    ObjectInputStream ois = null;
    try
    {
        ois = new ObjectInputStream(new FileInputStream(fMensajes));
        mensajes = (ArrayList<Mensaje>)ois.readObject();
```

```
    }
    catch (ClassNotFoundException e)
    {
       System.out.println("Error en los datos del fichero");
    }
    catch (FileNotFoundException e)
    {

    }
    catch (IOException e)
    {
       System.out.println("Error leyendo el fichero");
    }
    catch (Exception e)
    {
       System.out.println("Error en los datos del fichero");
    }
    finally
    {
       try
       {
          if (ois!=null)
             ois.close();

       }
       catch (IOException e)
       {
          System.out.println("Error cerrando el fichero");
       }
    }
}
```

```
    public static void main(String[] args) {
        recuperarMensajes();
        int op = menu();
        while (op!=8)
        {
            switch (op)
            {
                case 1: listarMensajes(); break;
                case 2: listarMensajesNormales(); break;
                case 3: listarEncriptado(); break;
                case 4: buscarMensaje(); break;
                case 5: anadirMensaje(); break;
                case 6: desencriptarMensaje(); break;
                case 7: eliminarMensaje(); break;
            }
            op = menu();
        }
        guardarMensajes();
    }

}
```

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 exordinario1819;

import java.io.Serializable;

/**
 *
 * @author anusk
 */
public abstract class Mensaje implements Serializable{

    protected int codigo;
    protected String texto;

    public Mensaje(int c, String t)
    {
        codigo = c;
        texto = t;
    }

    @Override
    public String toString()
    {
        return codigo+" "+texto;
    }

    public int getCodigo() {
```

```
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }


}
```

IENCRIPTABLE

```
package exordinario1819;

/**
 *
 * @author anusk
 */
public interface IEncriptable {

    public void encriptar();
    public String desencriptar();

}
```

MENSAJE SIN ENCRIPTAR

```
* 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 exordinario1819;


/**
 *
 * @author anusk
 */
public class MensajeSinEncriptar extends Mensaje{


   public MensajeSinEncriptar(int c, String t)
   {
      super(c,t);
   }
```

ENCRIPTADO

```
* 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 exordinario1819;

/**
 *
 * @author anusk
 */
public class MensajeEncriptado extends Mensaje implements IEncriptable{

    public MensajeEncriptado(int c, String t)
    {
        super(c, t);
        encriptar();
    }

    public void encriptar()
    {
        String tEncriptado = "";
        for (int i = 0;i<texto.length();i++)
        {
            tEncriptado+=(char)(texto.charAt(i)+1);
        }
        texto = tEncriptado;
    }

    public String desencriptar()
    {
```

```
        String tDesencriptado = "";
        for (int i = 0;i<texto.length();i++)
        {
            tDesencriptado+=(char)(texto.charAt(i)-1);
        }
        return tDesencriptado;
    }

    public void mostrarDesencriptado()
    {
        System.out.println(codigo+" "+desencriptar());
    }
}
```