

La empresa YoTeLoLLevo quiere implementar una aplicación para almacenar sus envíos de paquetería. De un envío debe almacenar la siguiente información: número de envío, nombre del remitente, dirección completa del remitente, código postal del remitente (formado por 5 dígitos y puede empezar por 0), nombre del destinatario, dirección completa del destinatario y código postal del destinatario (formado por 5 dígitos y puede empezar por 0). El número de envío será asignado por el programa de forma que al primer pedido se le asigna el 1, al segundo el 2 y así sucesivamente. La empresa desea almacenar dos tipos de envíos distintos: envíos ordinarios y envíos con seguimiento. Los envíos con seguimiento almacenarán también un código de seguimiento que estará formado por las letras PK seguidas de 10 dígitos, los 5 primeros serán el número de envío (se completará con ceros a la izquierda hasta completar los 5 dígitos) y los 5 siguientes serán el código postal del destinatario. Así, por ejemplo, un código de seguimiento podría ser PK0000128040. Para generar el código los envíos con seguimiento implementarán una interfaz llamada ISeguimiento con dos métodos: generarCodigo que devolverá el código de seguimiento y mostrarSeguimiento que mostrará un mensaje por pantalla con el texto "Mostrando seguimiento del envío PK0000128040", donde se mostrará el código de seguimiento que corresponda. También generará un fichero de texto con el mismo texto "Mostrando seguimiento del envío PK0000128040" y cuyo nombre será el código de seguimiento con la extensión txt. Se implementarán únicamente los constructores con parámetros y los gets y sets que sean necesarios. Se implementarán métodos toString que devolverá un String con la información de un envío donde aparecerá toda la información incluido el número de seguimiento en el caso de los envíos con seguimiento. Se pide: - Implementar las clases descritas ajustándose al esquema diseñado en el apartado anterior. Se implementarán los métodos que se consideren necesarios y se valorará el correcto uso de los modificadores. - Implementar un programa Mensajería que contenga un método main. Dicho programa creará un array de envíos cuyo tamaño se desconoce (un único array para todos ellos) y mostrará un menú con las siguientes opciones, cuya función se describe a continuación de cada una de ellas: 1. Nuevo envío: si se elige esta opción se comprobará si hay espacio para nuevos envíos en el array y, si lo hay, pedirá al usuario todos los datos del envío y posteriormente a eso se preguntará si se desea que el envío se haga con seguimiento o no. Se añadirá el envío al array si se trata de un envío con seguimiento se mostrará el texto "Se ha generado un envío con código de seguimiento PX0000128040" sustituyendo el código por el que corresponda. Si no hay espacio en el array se mostrará un mensaje indicándolo. 2. Envío entregado: se pedirá un número de envío y, si éste se encuentra en el array, lo elimina. Si no se encuentra muestra un mensaje indicándolo. 3. Listado de envíos: mostrará un listado con la información de todos los envíos. 4. Mostrar seguimiento: pedirá un código de seguimiento de un envío y si el envío se encuentra en el array, llamará a su método mostrarSeguimiento. 5. Salir: el programa finalizará al seleccionar esta opción. Página 2 de 2 A no ser que se elija la opción salir después de cada acción se volverá a mostrar el menú. Cualquier decisión que se tome por no estar especificada en el enunciado y que no modifique ninguna de las especificaciones deberá ser explicada razonadamente. Para evitar que al salir del programa se pierdan todos los datos se deberá guardar todos los envíos del array en un fichero binario antes de terminar la ejecución del programa. Al arrancar el programa se comprobará si el fichero binario de los envíos existe, si es así, rellena el array con los datos del fichero Se deberá controlar cualquier posible excepción que pudiera darse. Para aprobar el examen será imprescindible que éste no tenga errores de compilación.

```
package pendientes2022;
```

```
import java.io.*;
```

```
import java.util.ArrayList;
```

```
import java.util.Iterator;
```

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

```
/**
```

```
 *
```

```
 * @author anusk
```

```
 */
```

```
public class Mensajeria {
```

```
    public static ArrayList<Envio> envios = new ArrayList<Envio>();
```

```
    public static final String FDATOS = "envios.bin";
```

```
    public static int leerInt()
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        while (true)
```

```
        {
```

```
            try
```

```
            {
```

```
                return sc.nextInt();
```

```
            }
```

```
            catch (Exception e)
```

```
            {
```

```
                System.out.println("Número no válido");
```

```
                sc.nextLine();
```

```
            }
```

```
}  
}
```

```
public static int menu()  
{  
    int op = 0;  
    while (op<1 || op>5)  
    {  
        System.out.println("1. Nuevo envÃ-o");  
        System.out.println("2. EnvÃ-o entregado");  
        System.out.println("3. Listado de envÃ-os");  
        System.out.println("4. Mostrar seguimiento");  
        System.out.println("5. Salir");  
        op = leerInt();  
    }  
    return op;  
}
```

```
public static void nuevoEnvio()  
{  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Nombre del remitente: ");  
    String nr = sc.nextLine();  
    System.out.print("DirecciÃ³n del remitente: ");  
    String dr = sc.nextLine();  
    String cr="";  
    while (!CodigoPostal.esValido(cr))  
    {  
        System.out.print("CÃ³digo postal del remitente: ");  
        cr = sc.nextLine();  
    }  
}
```

```

System.out.print("Nombre del destinatario: ");

String nd = sc.nextLine();

System.out.print("Dirección del destinatario: ");

String dd = sc.nextLine();

String cd="";

while (!CodigoPostal.esValido(cd))

{

    System.out.print("Código postal del destinatario: ");

    cd = sc.nextLine();

}

int op = 0;

while (op!=1 && op!=2)

{

    System.out.println("¿Qué tipo de envío desea realizar?\n1. Ordinario\n2. Con seguimiento");

    op = leerInt();

}

try

{

    if (op==1)

    {

        envios.add(new EOrdinario(nr,dr,new CodigoPostal(cr),nd,dr,new CodigoPostal(cd)));

        System.out.println("Se ha añadido el envío");

    }

    else

    {

        ESeguimiento e = new ESeguimiento(nr,dr,new CodigoPostal(cr),nd,dr,new CodigoPostal(cd));

        envios.add(e);

        System.out.println("Se ha generado un envío con código de seguimiento "+e.generarCodigo());

    }

}

```

```

    }
    catch (CPEException e)
    {
        System.out.println("No se ha podido añadir el envío por un error en el código postal");
    }
}

```

```

public static Envio buscar(int n)
{
    Iterator<Envio> it = envios.iterator();
    while (it.hasNext())
    {
        Envio e = it.next();
        if (e.getnEnvio()==n)
            return e;
    }
    return null;
}

```

```

public static void envioEntregado()
{
    System.out.print("Número de envío: ");
    int n = leerInt();
    Envio e = buscar(n);
    if (e!=null)
    {
        envios.remove(e);
        System.out.println("Se ha entregado el envío "+n);
    }
}

```

```

else
    System.out.println("No existe el envÃo "+n);
}

public static void listadoEnvios()
{
    Iterator<Envio> it = envios.iterator();
    while (it.hasNext())
    {
        System.out.println(it.next());
    }
}

public static void mostrarSeguimiento()
{
    System.out.print("NÃmero de envÃo: ");
    int n = leerInt();
    Envio e = buscar(n);
    if (e!=null)
    {
        if (e instanceof ESeguimiento)
            ((ESeguimiento) e).mostrarSeguimiento();
        else
            System.out.println("El envÃo "+n+" no tiene seguimiento");
    }
    else
        System.out.println("No existe el envÃo "+n);
}

public static void guardarDatos()

```

```

{
    ObjectOutputStream oos = null;
    try
    {
        oos = new ObjectOutputStream(new FileOutputStream(FDATOS));
        oos.writeObject(envios);
        oos.writeInt(Envio.getContador());
    }
    catch (IOException e)
    {
        System.out.println("Error guardando datos");
    }
    finally
    {
        try
        {
            if (oos!=null)
            {
                oos.close();
            }
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

```

```

public static void recuperarDatos()

```

```

{
    ObjectInputStream ois = null;

```

```

try
{
    ois = new ObjectInputStream(new FileInputStream(FDATOS));
    envios = (ArrayList<Envio>)ois.readObject();
    Envio.setContador(ois.readInt());
}
catch (ClassNotFoundException e)
{
    System.out.println("Error en los datos");
}
catch (FileNotFoundException e)
{
}

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

```



```
        System.out.println("Error cerrando el fichero");
    }
}
}
```

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

```
    recuperarDatos();
    int op = menu();
    while (op!=5)
    {
        switch (op)
        {
            case 1: nuevoEnvio(); break;
            case 2: envioEntregado(); break;
            case 3: listadoEnvios(); break;
            case 4: mostrarSeguimiento(); break;
        }
        op=menu();
    }
    guardarDatos();

}

}
```


ENVIO.JAVA

```
package pendientes2022;

import java.io.Serializable;

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

    protected int nEnvio;
    protected String nombreR;
    protected String dirR;
    protectedCodigoPostal cpR;
    protected String nombreD;
    protected String dirD;
    protectedCodigoPostal cpD;

    private static int contador = 1;

    public Envio(String nr, String dr, CodigoPostal cr, String nd, String dd,
CodigoPostal cd)
    {
        nEnvio = contador++;
        nombreR = nr;
        dirR = dr;
        cpR = cr;
        nombreD = nd;
        dirD = dd;
        cpD = cd;
    }

    public static int getContador() {
        return contador;
    }

    public static void setContador(int contador) {
        Envio.contador = contador;
    }

    public int getnEnvio() {
        return nEnvio;
    }

    public String toString()
    {
        return nEnvio+"\tRemitente:
"+nombreR+"\t"+dirR+"\t"+cpR.getCodigoPostal()+
        "\tDestinatario:
"+nombreD+"\t"+dirD+"\t"+cpD.getCodigoPostal();
    }
}
```

```

CODIGOPOSTAL.JAVA
package pendientes2022;

import java.io.Serializable;

/**
 *
 * @author anusk
 */
public class CodigoPostal implements Serializable{

    private String cp;

    public CodigoPostal(String cp) throws CPEException
    {
        if (esValido(cp))
            this.cp = cp;
        else
            throw new CPEException();
    }

    public static boolean esValido(String cp)
    {
        if (cp.length()!=5) return false;
        try
        {
            int numero = Integer.valueOf(cp);
            if (numero>=0 && numero<=99999)
                return true;
            else
                return false;
        }
        catch (NumberFormatException e)
        {
            return false;
        }
    }

    public String getCodigoPostal()
    {
        return cp;
    }

}

```

CPEXCEPTION

```
package pendientes2022;

/**
 *
 * @author anusk
 */
public class CPEXception extends Exception{

    public CPEXception()
    {
    }
    public CPEXception(String m)
    {
        super(m);
    }

}
```

EORDINARIO

```
package pendientes2022;

/**
 *
 * @author anusk
 */
public class EOrdinario extends Envio{

    public EOrdinario(String nr, String dr, CodigoPostal cr, String nd,
String dd, CodigoPostal cd)
    {
        super(nr,dr,cr,nd,dd,cd);
    }

}
```

ESEGUIMINETO

```
package pendientes2022;
```

```
import java.io.*;
```

```
/**
```

```
 *
```

```
 * @author anusk
```

```
 */
```

```
public class ESeguimiento extends Envio implements ISeguimiento{
```

```
    public ESeguimiento(String nr, String dr, CodigoPostal cr, String nd,  
String dd, CodigoPostal cd)
```

```
    {
```

```
        super(nr,dr,cr,nd,dd,cd);
```

```
    }
```

```
@Override
```

```
public String generarCodigo() {
```

```
    String ne = String.valueOf(nEnvio);
```

```
    while (ne.length()<5)
```

```
        ne="0"+ne;
```

```
    return "PK"+ne+cpD.getCodigoPostal();
```

```
}
```

```
@Override
```

```
public void mostrarSeguimiento() {
```

```
    System.out.println("Mostrando seguimiento del envÃ-o
```

```
" + generarCodigo());
```

```
    BufferedWriter bw = null;
```

```
    try
```

```
    {
```

```
        bw = new BufferedWriter(new FileWriter(generarCodigo()+".txt"));
```

```
        bw.write("Mostrando seguimiento del envÃ-o " + generarCodigo());
```

```
    }
```

```
    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 de texto");
```

```
        }
```

```
    }
```

```
}
```

```
public String toString()
```

```
{
```

```
    return super.toString()+"\t"+generarCodigo();
```

```
ISEGUIMIENTO
package pendientes2022;

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

    public String generarCodigo();
    public void mostrarSeguimiento();

}
```