

Una agencia de viajes ofrece a sus clientes paquetes vacacionales. Un paquete vacacional está formado por un identificador numérico asignado por el programa automáticamente de forma que no haya repetidos, ciudad de origen, ciudad de destino, transporte, alojamiento y una lista de extras, además de la fecha (será un String y no será necesario comprobar que sea correcta) de salida y de la duración del viaje expresada en días. De un transporte debemos conocer la duración del trayecto (se asume que la duración es la misma que la de la vuelta) expresada en minutos y el precio. Tendremos dos tipos de transporte: avión o tren. En el caso del transporte en avión almacenaremos si el vuelo es directo o no. Del alojamiento se guardará el nombre, distancia en metros al centro de la ciudad y precio por día. Los extras almacenarán una descripción y un precio. Los paquetes vacacionales tendrán un método precioTotal que nos indicará el precio total del vuelo+alojamiento+extras. Se desea implementar un programa que gestione una lista de paquetes vacacionales y que nos permita las siguientes operaciones:

1. Crear paquete vacacional: Creará un nuevo paquete vacacional pidiendo los datos necesarios y preguntando por todos los extras que se quiera añadir al paquete.
2. Mostrar paquetes vacacionales: Mostrará por pantalla todos los paquetes vacacionales creados con el precio total de cada uno de ellos y también guardará en un fichero de texto llamado paquetes.txt ese mismo listado.
3. Mostrar paquetes con filtro: Mostrar aquellos paquetes cuya duración por trayecto en el transporte no sea superior a la indicada por el usuario y que además no incluyan vuelos con escalas.
4. Eliminar paquete vacacional: eliminará del listado el paquete cuyo identificador indique el usuario (si el identificador dado no existe lo indicará mediante un mensaje).
5. Salir Al salir del programa se almacenará la situación actual del programa de forma que al volver a iniciar el programa se recupere los paquetes vacacionales creados. Se deberá controlar todas las posibles excepciones.

```
package examenord2223;
```

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

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

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

```
public class ExamenOrd2223 {
```

```
    public static ArrayList<Paquete> paquetes = new ArrayList<Paquete>();
```

```
    public static String fDatos = "paquetes.bin";
```

```
    public static Extra pedirExtra()
```

```
    {
```

```
        String desc = Lector.leerLinea("Descripción: ");
```

```
        double precio = Lector.leerDouble("Precio: ", "Precio inválido, escríbalo de nuevo");
```

```
        return new Extra(desc, precio);
```

```
    }
```

```
    public static Alojamiento pedirAlojamiento()
```

```
    {
```

```
        String nombre = Lector.leerLinea("Nombre: ");
```

```
        int dist = Lector.leerInt("Distancia en metros al centro: ", "Distancia inválida");
```

```
        double precio = Lector.leerDouble("Precio: ", "Precio inválido, escríbalo de nuevo");
```

```
        return new Alojamiento(nombre, dist, precio);
```

```
    }
```

```
    public static int menuTransporte()
```

```
    {
```

```
        int op = 0;
```

```

while (op<1 || op >2)
{
    System.out.println("1. Avión");
    System.out.println("2. Tren");
    op = Lector.leerInt();
}
return op;
}

```

```

public static int menuAvion()
{
    int op = 0;
    while (op<1 || op >2)
    {
        System.out.println("1. Vuelo directo");
        System.out.println("2. Vuelo con escalas");
        op = Lector.leerInt();
    }
    return op;
}

```

```

public static Transporte pedirTransporte()
{
    int tipo = menuTransporte();
    int dur = Lector.leerInt("Duración: ");
    double precio = Lector.leerInt("Precio: ");
    if (tipo == 1)
    {
        int dir = menuAvion();
        return new Avion(dur,precio,dir==1);
    }
}

```

```

else
    return new Tren(dur,precio);
}

```

```

public static void nuevoPaquete()
{
    String or = Lector.leerLinea("Origen: ");
    String dest = Lector.leerLinea("Destino: ");
    Transporte t = pedirTransporte();
    Alojamiento al = pedirAlojamiento();
    ArrayList<Extra> extras = new ArrayList<Extra>();
    boolean seguir = Lector.leerSINO("¿Desea añadir algún extra? (sS/nN): ", true);
    while (seguir)
    {
        extras.add(pedirExtra());
        seguir = Lector.leerSINO("¿Desea añadir algún extra más? (sS/nN): ", true);
    }

    String fecha = Lector.leerLinea("Fecha: ");
    int dur = Lector.leerInt("Duración del viaje: ");

    paquetes.add(new Paquete(or,dest,t,al,extras,fecha,dur));
}

```

```

public static void mostrarPaquetes()
{
    String fpaquetes = "paquetes.txt";
    BufferedWriter bw = null;
    try

```

```

{
    bw = new BufferedWriter(new FileWriter(fpaquetes));

    Iterator <Paquete> it = paquetes.iterator();
    while (it.hasNext())
    {
        Paquete p = it.next();
        System.out.println(p+"\n");
        bw.write(p+"\n\n");
    }
}
catch (IOException e)
{
    System.err.println("Error en el fichero de texto de paquetes");
}
finally
{
    try
    {
        if (bw!=null)
            bw.close();
    }
    catch (IOException e)
    {
        System.err.println("Error cerrando el fichero de texto");
    }
}
}

```

```

public static void mostrarConFiltro()

```

```

{
    int dur = Lector.leerInt("Indique la duraci3n m3xima por trayecto en minutos: ");
    Iterator<Paquete> it = paquetes.iterator();
    while (it.hasNext())
    {
        Paquete p = it.next();
        if (p.durTransporte()<=dur &&
            ((p.getTransporte() instanceof Avion && ((Avion)p.getTransporte()).esDirecto()) ||
             (p.getTransporte() instanceof Tren)))
            System.out.println(p+"\n");
    }
}

```

```

public static Paquete buscar(int id)
{
    Iterator<Paquete> it = paquetes.iterator();
    while (it.hasNext())
    {
        Paquete p = it.next();
        if (p.getId()==id)
            return p;
    }
    return null;
}

```

```

public static void eliminar()
{
    int id = Lector.leerInt("Identificador: ");
    Paquete p = buscar(id);
    if (p!=null)
        paquetes.remove(p);
}

```

```
else  
    System.out.println("Identificador no encontrado");  
}
```

```
public static int menu()  
{  
    int op = 0;  
    while (op<1 || op >5)  
    {  
        System.out.println("1. Crear paquete vacacional");  
        System.out.println("2. Mostrar y exportar paquetes");  
        System.out.println("3. Mostrar paquetes filtrados");  
        System.out.println("4. Eliminar paquete");  
        System.out.println("5. Salir");  
        op = Lector.leerInt();  
    }  
    return op;  
}
```

```
public static void guardar()  
{  
    ObjectOutputStream oos = null;  
    try  
    {  
        oos = new ObjectOutputStream(new FileOutputStream(fDatos));  
        oos.writeObject(paquetes);  
        oos.writeInt(Paquete.getContador());  
    }  
    catch (IOException e)  
    {  
        System.err.println("Error guardando datos");  
    }  
}
```

```

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

```

```

public static void recuperar()
{
    ObjectInputStream ois = null;
    try
    {
        ois = new ObjectInputStream(new FileInputStream(fDatos));
        paquetes = (ArrayList<Paquete>)ois.readObject();
        Paquete.setContador(ois.readInt());
    }
    catch (FileNotFoundException e)
    {}

    catch (Exception e)
    {
        System.err.println("Error recuperando datos");
    }
}

```



```

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

```

```

public static void main(String[] args) {

```

```

    recuperar();
    int op = menu();
    while (op!=5)
    {
        switch (op)
        {
            case 1: nuevoPaquete(); break;
            case 2: mostrarPaquetes(); break;
            case 3: mostrarConFiltro(); break;
            case 4: eliminar(); break;
        }
        op = menu();
    }
    guardar();
}

```

}

}

```

package examenord2223;

import java.io.Serializable;

/**
 *
 * @author anusk
 */
public class Alojamiento implements Serializable{
    private String nombre;
    private int dist;
    private double precio;

    public Alojamiento(String n, int d, double p)
    {
        nombre = n;
        dist = d;
        precio = p;
    }

    public double getPrecio()
    {
        return precio;
    }

    @Override
    public String toString()
    {
        return nombre+"\n"+dist+" metros hasta el centro\n"+precio+"â,~/dia";
    }
}

```

```

package examenord2223;

import java.io.Serializable;

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

    protected int dur;
    protected double precio;

    public Transporte(int d, double p)
    {
        dur = d;
        precio = p;
    }

    public double getPrecio()
    {
        return precio;
    }

    public int getDuracion()
    {
        return dur;
    }

    @Override
    public String toString()
    {
        return dur+" min \t"+precio+"â,-";
    }

}

```

```
package examenord2223;

import java.io.Serializable;

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

    private String desc;
    private double precio;

    public Extra (String d, double p)
    {
        desc = d;
        precio = p;
    }

    public double getPrecio()
    {
        return precio;
    }

    @Override
    public String toString()
    {
        return desc+"\t"+precio+"â¬“";
    }
}
```

```

public class Paquete implements Serializable{

    private static int contador = 1;
    private int id;
    private String origen;
    private String destino;
    private Transporte trans;
    private Alojamiento aloj;
    private ArrayList<Extra> extras;
    private String fecha;
    private int dias;

    public Paquete(String or, String dest, Transporte trans, Alojamiento
aloj,
        ArrayList<Extra> ex, String fech, int d)
    {
        id = contador++;
        origen = or;
        destino = dest;
        this.trans = trans;
        this.aloj = aloj;
        extras = ex;
        fecha = fech;
        dias = d;
    }

    public int getId()
    {
        return id;
    }

    public static int getContador()
    {
        return contador;
    }

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

    public double precioTotal()
    {
        double precio = trans.getPrecio()+aloj.getPrecio()*dias;
        Iterator <Extra> it = extras.iterator();
        while (it.hasNext())
            precio+=it.next().getPrecio();
        return precio;
    }

    public int durTransporte()
    {
        return trans.getDuracion();
    }

    public Transporte getTransporte()
    {
        return trans;
    }

```

```
}

@Override
public String toString()
{
    String texto = "ID: "+id+"\nORIGEN: "+origen+"\nDESTINO: "+destino+
        "\nTRANSPORTE: "+trans+"\nALOJAMIENTO: "+aloj+
        "\nEXTRAS:\n";
    Iterator<Extra> it = extras.iterator();
    while (it.hasNext())
        texto+= "\t"+it.next().toString()+"\n";
    texto+= "FECHA: "+fecha+"\nDIAS: "+dias;
    return texto;
}

}
```

```
public class Avion extends Transporte{

    private boolean directo;

    public Avion(int d, double p, boolean dir)
    {
        super(d,p);
        directo = dir;
    }

    public boolean esDirecto()
    {
        return directo;
    }

    @Override
    public String toString()
    {
        return "AVION: "+ super.toString()+" "+ ((directo)?"Vuelo
directo":"Vuelo con escalas");
    }
}
```



```
package examenord2223;

/**
 *
 * @author anusk
 */
public class Tren extends Transporte{

    public Tren(int d, double p)
    {
        super(d,p);
    }

    @Override
    public String toString()
    {
        return "TREN: "+ super.toString();
    }
}
```