***Ejercicio 1***

Realizar un programa que gestiona una agenda. Para ello se declara:

la clase contacto con los atributos (nombre, apellidos, mail, teléfono, fechaNacimiento).

El programa utilizará una estructura (ListList<Contacto> )para almacenar los contactos permitirá:

* 1. Añadir contacto
  2. Eliminar Contato
  3. Modificar contacto
  4. Buscar Contacto por nombre y apellidos
  5. Listar ordenadamente todos los contactos

***Ejercicio 2***

La concesionaria de autopistas Viaje Feliz, está implementando un sistema de facturación para sus clientes. Para la implementación del software se ha sostenido una reunión de coordinación con el área técnica encargada de la operación de los pórticos que registran el paso de vehículos. Uno de los acuerdos de esta reunión es que el área técnica entregará un archivo de texto con los detalle de tráfico de un mes para poder realizar la facturación correspondiente. El formato de archivo acordado es el siguiente: una línea por cada vehiculo que atraviesa un pórtico, detallando fecha, patente e identificador del pórtico (“P1” a “P9”). Ejemplo 25/06/2006 RA3728 P4

25/06/2006 TR7865 P1

25/06/2006 HK7827 P1

26/06/2006 RA3728 P5

26/06/2006 HK7827 P2

El equipo implementador del software de facturación ha decido manejar cada una de las líneas como un objeto de la clase Evento. Cada evento representa el paso de un vehículo por uno de los pórticos de la autopista:

Para el procesamiento de los eventos se ha creado una clase llamada Facturador, que tiene un atributo ya definido para almacenar en memoria el detalle del archivo. Este atributo corresponde a un array de objetos de la clase Evento

public class Facturador {

List<Evento> traficoMensual; … // resto de la definición de la clase }

Se pide que se implemente los siguientes métodos para la clase Facturador:

a) public void cargarDatos( ) Carga el detalle de eventos almacenados en un archivo. Los datos deben ser almacenado en el atributo traficoMensual

.

b) public void facturacion(String Patente) Recibe como parámetro un String con una Matricula particular y genera una salida con el detalle de tráfico del vehiculo correspondiente a la matricula. Al final del archivo se indica además el total a pagar para este vehículo. Puede suponerse que el método facturación siempre será llamado después de cargar\_archivo.

***Ejercicio 2***

La concesionaria de autopistas Viaje Feliz, está implementando un sistema de facturación para sus clientes. Para la implementación del software se ha sostenido una reunión de coordinación con el área técnica encargada de la operación de los pórticos que registran el paso de vehículos. Uno de los acuerdos de esta reunión es que el área técnica entregará un archivo de texto con los detalle de tráfico de un mes para poder realizar la facturación correspondiente. El formato de archivo acordado es el siguiente: una línea por cada vehiculo que atraviesa un pórtico, detallando fecha, patente e identificador del pórtico (“P1” a “P9”). Ejemplo 25/06/2006 RA3728 P4

25/06/2006 TR7865 P1

25/06/2006 HK7827 P1

26/06/2006 RA3728 P5

26/06/2006 HK7827 P2

El equipo implementador del software de facturación ha decido manejar cada una de las líneas como un objeto de la clase Evento. Cada evento representa el paso de un vehículo por uno de los pórticos de la autopista:

Para el procesamiento de los eventos se ha creado una clase llamada Facturador, que tiene un atributo ya definido para almacenar en memoria el detalle del archivo. Este atributo corresponde a un array de objetos de la clase Evento

public class Facturador {

ArrayList>Evento> traficoMensual; … // resto de la definición de la clase }

Se pide que se implemente los siguientes métodos para la clase Facturador:

a) public void cargarDatos( ) Carga el detalle de eventos almacenados en un archivo. Los datos deben ser almacenado en el atributo traficoMensual

.

b) public void facturacion(String Patente) Recibe como parámetro un String con una Matricula particular y genera una salida con el detalle de tráfico del vehiculo correspondiente a la matricula. Al final del archivo se indica además el total a pagar para este vehículo. Puede suponerse que el método facturación siempre será llamado después de cargar\_archivo.

***Ejercicio 3***

Elección de delegado.

Realizar un programa que lea las votaciones realizadas en un aula para la elección de delegado hasta recibir la palabra “Fin”.

* Declarar la clase Alumno con las propiedades (nombre, votos )
* Declarar un List <Alumno>
* Cada alumno votado se incorporará al array de alumnos si no existe y sus votos se inicializará a uno y si existe solo se incrementarán sus votos

Una vez realizado del recuento se mostrará:

* la lista de alumnos ordenada de mayor a menor votación
* El delegado y subdelegado; asignado como delegado el primero y subdelegado el segundo más votado:

Ejemplo:

Entrada: Juan Luis Juan Luis Ana Luis Ana Juan Juan Fin

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Juan | 4 | | |  |  | | --- | --- | | Luis | 3 | | |  |  | | --- | --- | | Ana | 2 | |  | …….. |  |