class Program

{

class Persona

{

// Estructura de Datos

// Unidad 3 Estructuras lineales

// 16210519 Escobedo Díaz Jesús Eduardo

// Programa 1

int boleto;

string nombre, sexo;

public Persona siguiente;

public Persona(int boleto, string nombre, string sexo)

{

this.boleto = boleto;

this.nombre = nombre;

this.sexo = sexo;

siguiente = null;

}

public int Boleto

{

get { return boleto; }

}

public string Nombre

{

get { return nombre; }

}

public string Sexo

{

get { return sexo; }

}

public void Mostrar()

{

Console.Write("Boleto: {0} | Nombre: {1} | Sexo: {2}\n", boleto, nombre, sexo);

}

}

class Vagon

{

int codigo;

Persona first, last;

public Vagon siguiente;

public Vagon(int codigo)

{

this.codigo = codigo;

first = last = null;

siguiente = null;

}

public int Codigo

{

get { return codigo; }

}

public void Insertar(Persona nuevo)

{

if (first == null)

first = nuevo;

else

last.siguiente = nuevo;

last = nuevo;

}

public void Mostrar()

{

if (first != null)

{

Persona actual = first;

while (actual != null)

{

actual.Mostrar();

actual = actual.siguiente;

}

}

else

Console.WriteLine("\nEl Vagón {0} Está Vacío", codigo);

}

public void MostrarVagon()

{

Console.Write("N° Vagon: {0}\n", codigo);

}

public Persona Buscar(int n)

{

Persona actual = first;

while (actual != null)

{

if (actual.Boleto == n)

return actual;

actual = actual.siguiente;

}

return null;

}

public void Eliminar(int n)

{

if (first != null)

{

Persona actual, padre;

padre = BuscarPadre(n);

if (padre == last)

{

Console.WriteLine("\nPasajero No Encontrado");

return;

}

if (padre == null)

{

actual = first;

first = first.siguiente;

padre = first;

}

else

{

actual = padre.siguiente;

padre.siguiente = actual.siguiente;

}

actual = null;

if (padre == null || padre.siguiente == null)

last = padre;

Console.WriteLine("Pasajero Bajado Del Vagón");

}

else

Console.WriteLine("\nNo Existe Ningún Pasajero");

}

private Persona BuscarPadre(int n)

{

Persona padre, actual;

actual = first;

padre = null;

while (actual != null)

{

if (actual.Boleto == n)

break;

padre = actual;

actual = actual.siguiente;

}

return padre;

}

}

class Tren

{

string destino;

Vagon first, last;

public Tren siguiente;

public Tren(string destino)

{

this.destino = destino;

first = last = null;

siguiente = null;

}

public string Destino

{

get { return destino; }

}

public void Insertar(Vagon nuevo)

{

if (first == null)

first = nuevo;

else

last.siguiente = nuevo;

last = nuevo;

}

public void Mostrar()

{

if (first != null)

{

Vagon actual = first;

while (actual != null)

{

Console.Write("\nTren Destino: {0}\n", destino);

actual.Mostrar();

actual.MostrarVagon();

actual = actual.siguiente;

}

}

else

Console.WriteLine("\nEl Tren Está Vacío");

}

public Vagon Buscar(int n)

{

Vagon actual = first;

while (actual != null)

{

if (actual.Codigo == n)

return actual;

actual = actual.siguiente;

}

return null;

}

}

class Estacion

{

Tren first, last;

public Estacion()

{

first = last = null;

}

public void Insertar(Tren nuevo)

{

if (first == null)

first = nuevo;

else

last.siguiente = nuevo;

last = nuevo;

}

public void Mostrar()

{

if (first != null)

{

Tren actual = first;

while (actual != null)

{

actual.Mostrar();

actual = actual.siguiente;

}

}

else

Console.WriteLine("\nLa Estación Está Vacía");

}

public Tren Buscar(string n)

{

Tren actual = first;

while (actual != null)

{

if (actual.Destino == n)

return actual;

actual = actual.siguiente;

}

return null;

}

}

static void Main(string[] args)

{

short op;

int bol = 0;

Estacion e = new Estacion();

Tren t; Vagon v; Persona p;

do

{

Console.WriteLine("\nESTACION DE TREN\n ");

Console.WriteLine("1. Ingresar Pasajero");

Console.WriteLine("2. Mostrar Todos Los Pasajeros");

Console.WriteLine("3. Buscar Pasajero");

Console.WriteLine("4. Bajar Pasajero");

Console.WriteLine("5. Salir");

Console.Write("Ingrese Opción: ");

op = short.Parse(Console.ReadLine());

switch (op)

{

case 1:

Console.Write("\nIngrese Destino Del Viaje: ");

string dv = Console.ReadLine();

t = e.Buscar(dv);

if (t == null)

{

Console.Write("\nTren Inexistente Con Este Destino...\nSe Creara Nuevo Destino - {0}\n", dv);

t = new Tren(dv);

e.Insertar(t);

}

Console.Write("\nIngrese Número De Vagon: ");

int nv = int.Parse(Console.ReadLine());

v = t.Buscar(nv);

if (v == null)

{

v = new Vagon(nv);

t.Insertar(v);

}

Console.Write("Ingrese Nombre: ");

string n = Console.ReadLine();

Console.Write("Ingrese Sexo: ");

string s = Console.ReadLine();

p = new Persona(bol, n, s);

v.Insertar(p);

bol++;

break;

case 2:

e.Mostrar();

break;

case 3:

Console.Write("\nIngrese Destino Del Tren: ");

string d = Console.ReadLine();

t = e.Buscar(d);

if (t != null)

{

Console.Write("Ingrese Codigo de Vagón: ");

int c = int.Parse(Console.ReadLine());

v = t.Buscar(c);

if (v != null)

{

Console.Write("Ingrese N° De Boleto: ");

int b = int.Parse(Console.ReadLine());

p = v.Buscar(b);

if (p != null)

{

Console.Write("\n");

p.Mostrar();

}

else

Console.Write("\nPasajero No Existe\n");

}

else

Console.Write("\nVagón No Existe\n");

}

else

Console.Write("\nTren No Existe\n");

break;

case 4:

Console.Write("Ingrese Destino De Tren Para Bajar: ");

d = Console.ReadLine();

t = e.Buscar(d);

if (t != null)

{

Console.Write("Ingrese Codigo de Vagón: ");

int c = int.Parse(Console.ReadLine());

v = t.Buscar(c);

if (v != null)

{

Console.Write("Ingrese N° De Boleto:");

int b = int.Parse(Console.ReadLine());

p = v.Buscar(b);

if (p != null)

{

Console.Write("\n");

v.Eliminar(b);

}

else

Console.Write("\nPasajero No Existe\n");

}

else

Console.Write("\nVagón No Existe\n");

}

else

Console.Write("\nTren No Existe\n");

break;

case 5:

Console.WriteLine("\nFin del Programa");

Console.ReadLine();

break;

default:

Console.WriteLine("\nOpción No Valida");

break;

}

} while (op != 5);

}

}