| {Ejercicio 4. Una agencia de alquiler de autos a nivel nacional posee 10 sucursales. Cada sucursal almacena la información de todos sus alquileres ordenada por marca y modelo (un solo campo). De cada alquiler se conoce fecha de alquiler, marca y modelo del auto, cantidad de días del alquiler y precio por día.  1) Haga un módulo que cargue todos los alquileres de las 10 sucursales. Las lecturas de los alquileres no tienen orden alguno. La lectura finaliza al leer un número de sucursal 0. La información debe retornarse en una estructura de datos adecuada.  2) Haga un módulo que reciba la estructura de datos devuelta por el módulo anterior y retorne un resumen de ventas para cada marca y modelo distinto, ordenada de manera descendente por cantidad de alquileres.  3) Haga un módulo que reciba la estructura de datos devuelta por el módulo implementado en el punto 2 e imprima la marca y modelo con mayor cantidad de alquileres a nivel nacional.  4) Escriba un programa que invoque a los tres módulos implementados y compruebe el correcto funcionamiento del mismo.}   program ejercicio4; uses genericlinkedlist;  const suc=10;  //----------------------------- DECLARACIONES --------------------------------------- type fecha=record dia:integer; mes:integer; anio:integer; end;  alquiler=record fechas:fecha; marcamodelo:string;  cantdias:integer; preciodia:real; end;  acumulado=record marcamodelo:string; cantacum:integer; montoacum:real; end;  lista=specialize linkedlist <alquiler>; listaacum=specialize linkedlist <acumulado>; vector=array[1..suc] of lista;   {1) Haga un módulo que cargue todos los alquileres de las 10 sucursales.  \* Las lecturas de los alquileres no tienen orden alguno. La lectura finaliza al leer un número de sucursal 0.   \* La información debe retornarse en una estructura de datos adecuada.}   //------------------------- MODULOS ---------------------------------------------------  PROCEDURE leeralquiler(var a:alquiler); var vmarcamodelo:array[1..12] of string=('Ford-Fiesta','Fiat-Palio','Chevrolet-Onix','Nissan-Sentra','Audi-A3','Toyota-Corolla','Ferrari-448','Renault-Clio','Seat-ibiza','Citroen-c3','Peugeot-100','Porsche-Victory'); begin  a.fechas.dia:=random(31)+1;  a.fechas.mes:=random(12)+1;  a.fechas.anio:=random(2024-2010+1)+2010; a.marcamodelo:=vmarcamodelo[random(12)+1]; a.cantdias:=random(60)+1; a.preciodia:=random(5000-1000+1)+1000; end;  //-------------------------------------------------------------------------------  procedure insertarordenado (var l:lista;a:alquiler); var seguir:boolean;  begin seguir:=True; l.reset();  while not l.eol() and seguir do  begin  if a.marcamodelo<l.current().marcamodelo then  seguir:=false  else   l.next();  end; l.insertcurrent(a); end;  //------------------------------------------------------------------------------  PROCEDURE cargar(var v:vector); var a:alquiler; sucursal:integer; i:integer;  begin for i:=1 to suc do begin v[i]:=lista.create(); end;   sucursal:=random(11); leeralquiler(a); while sucursal<>0 do begin insertarordenado(v[sucursal],a); sucursal:=random(11); leeralquiler(a);  end; end;  procedure imprimirAlquileres(v: vector); var  i: Integer;  a: alquiler; begin  writeln('======================================================================');  writeln(' Listado de Alquileres por Sucursal ');  writeln('======================================================================');  writeln('| Marca y Modelo | Fecha | Dias | Precio/día |');  for i := 1 to suc do  begin  writeln('Sucursal ', i, ':');    v[i].reset;  while not v[i].eol do  begin  a := v[i].current();    // Alineamos las columnas de forma adecuada para mejorar la presentación  write('| ', a.marcamodelo:18); // Marca y modelo alineados a la izquierda, 18 caracteres  write(' | ', a.fechas.dia:2, '/', a.fechas.mes:2, '/', a.fechas.anio:4); // Fecha en formato DD/MM/AAAA  write(' | ', a.cantdias:5); // Días del alquiler, alineados a la derecha con 5 caracteres  writeln(' | ', a.preciodia:10:2, ' |'); // Precio por día, alineado a la derecha con 10 caracteres y 2 decimales    v[i].next();  end;  writeln('');  end;  writeln('') end;     {2) Haga un módulo que reciba la estructura de datos devuelta por el módulo anterior y retorne un resumen de ventas para cada marca y modelo distinto, ordenada de manera descendente por cantidad de alquileres. }  procedure minimo(var v:vector;var min:alquiler;var pos:integer); var i:integer; begin  min.marcamodelo:='zzz';  for i:=1 to suc do BEGIN  if not (v[i].eol()) then begin  if v[i].current.marcamodelo<min.marcamodelo then  begin  min:=v[i].current();  pos:=i;  end;  end;  end; if min.marcamodelo<>'zzz' then  begin  v[pos].next(); end end;   procedure mergeacumulador (v:vector;var l2:listaacum); var  min:alquiler; a:acumulado;  i,pos:integer; begin  l2:=listaacum.create();  for i:=1 to suc do begin v[i].reset(); end;  minimo(v,min,pos); while min.marcamodelo<>'zzz' do begin  a.marcamodelo:=min.marcamodelo; a.cantacum:=0; a.montoacum:=0; while min.marcamodelo=a.marcamodelo do begin a.cantacum:=a.cantacum + 1; a.montoacum:=a.montoacum+(min.preciodia\*min.cantdias); minimo(v,min,pos); end; l2.add(a); end;   end;  procedure imprimirlista(l2: listaacum); begin  l2.reset();  writeln('-----------------------------------------------------');  writeln('| Marca y Modelo | Cantidad | Monto Total |');  writeln('-----------------------------------------------------');    while not l2.eol do  begin  write('| ', l2.current().marcamodelo:20);   write(' | ', l2.current().cantacum:8);   writeln(' | ', l2.current().montoacum:12:2, ' |');   l2.next();  end; end;  {3) Haga un módulo que reciba la estructura de datos devuelta por el módulo implementado en el punto 2 e imprima la marca y modelo con mayor cantidad de alquileres a nivel nacional.}  PROCEDURE MAXIMO (l2:listaacum); var maximo:integer; marcamax:string; begin maximo:=-999; marcamax:='zzz'; l2.reset();  while not l2.eol() do begin  if l2.current().cantacum>maximo then   begin  marcamax:=l2.current().marcamodelo;  maximo:=l2.current.cantacum;  end;  l2.next();  end; writeln('La marca con mas cantidad de alquileres es ', marcamax, ' con un total de ',maximo, ' alquileres vendidos'); end;  {4) Escriba un programa que invoque a los tres módulos implementados y compruebe el correcto funcionamiento del mismo.} Var v:vector; l2:listaacum; begin // punto 1 randomize; cargar(v); imprimirAlquileres(v); // punto 2 mergeacumulador(v,l2); WRITELN('.....................................................'); writeln ('Lista Acumulada'); imprimirlista(l2); // punto 3 WRITELN('.....................................................'); WRITELN(''); maximo(l2);  end. |
| --- |