

## micros

Unit micros;

{Trabajo desarrollado en el primer cuatrimestre 2013 por los alumnos Arcuri, Santos y Rios}

Interface

TYPE

```
texto= string [30];
dias= 1..31;
meses= 1..12;
fechas = RECORD
    dia: dias;
    mes: meses;
    ano: integer;
end;
asientos= array [1..30] of boolean;
micro = RECORD
    destino: texto;
    fecha: fechas;
    a: asientos;
    precio: real;
end;
```

```
PROCEDURE crearmicro (VAR m:micro; dest:texto; partida:fechas;prec:real);
    {crea un micro}
PROCEDURE ocupar (VAR m: micro; num: integer);
    {ocupa un asiento a partir de un número}
FUNCTION cantlibres(m:micro): integer;
    {devuelve la cantidad de asientos no ocupados}
PROCEDURE liberar (VAR m:micro; num:integer);
    {libera un asiento a partir de un número dado}
FUNCTION verificar (m:micro; num:integer): boolean;
    {retorna true si esta ocupado y false si esta libre}
FUNCTION verlibre (m:micro): integer;
    {devuelve un número de asiento libre}
FUNCTION pertenece (num: integer) :boolean;
    {devuelve true si el número está dentro del rango, y falso en lo contrario}
FUNCTION haylibre (m:micro): boolean;
    {devuelve si hay un asiento disponible en el micro}
PROCEDURE asignar (m1: micro; VAR m2:micro);
    {asigna los datos de un micro a otro}
FUNCTION verD (m:micro): texto;
    {devuelve el nombre de un colectivo}
PROCEDURE verF (mic:micro;VAR d:dias; VAR m:meses; VAR a:integer);
    {devuelve la fecha del micro}
FUNCTION verP(m:micro):real;
    {devuelve el precio de cada asiento del micro}
implementation
```

```
PROCEDURE crearmicro (VAR m:micro; dest:texto; partida:fechas; prec:real);
var i:integer;
begin
m.destino:=dest;
m.precio:=prec;
for i:=1 to 30 do
m.a[i]:=false;
with m.fecha do begin
dia:=partida.dia;
mes:=partida.mes;
ano:=partida.ano;
end;
end;
```

```
PROCEDURE ocupar (VAR m: micro; num: integer);
```

## micros

```
begin
m.a[num]:=true;
end;

FUNCTION cantlibres(m:micro): integer;
var
cant,i:integer;
begin
cant:=0;
for i:=1 to 30 do begin
    if(m.a[i]=false) then
        cant:=cant+1;
end;
cantlibres:=cant;
end;

PROCEDURE liberar (VAR m:micro; num:integer);
begin
m.a[num]:=false;
end;

FUNCTION verificar (m:micro; num: integer):boolean;
begin
if (m.a[num]= false) then
verificar:=true
else
    verificar:=false;
end;

FUNCTION verlibre(m:micro):integer;
var
encontre:boolean;
i:integer;
begin
encontre:=false;
i:=1;
while(encontre=false) and (i<=30) do begin
if (m.a[i]= false) then
    encontre:=true
else
    i:=i+1;
end;
verlibre:=i;
end;

FUNCTION pertenece (num: integer) :boolean;
Begin
    if(num>=1) and (num<=30) then
        pertenece:= true
    else
        pertenece:= false;
end;

FUNCTION haylibre (m:micro): boolean;
VAR
    i:integer;
    ok:boolean;
Begin
    ok:=false;
    i:=1;
```

```

                                micros
while(i<=30) and NOT(ok)do
  Begin
    if(m.a[i]=false)then
      ok:=true
    else
      i:=i+1;
    end;
  haylibre:=ok;
end;
PROCEDURE asignar (m1: micro; VAR m2:micro);
Begin
m2:=m1;
end;
FUNCTION verD (m:micro): texto;
  Begin
    verD:= m.destino;
  end;
PROCEDURE verF (mic:micro; VAR d:dias; VAR m:meses; VAR a:integer);
  Begin
    d:=mic.fecha.dia;
    m:=mic.fecha.mes;
    a:=mic.fecha.ano;
  end;
FUNCTION verP(m:micro):real;
  Begin
    verP:=m.precio;
  end;
end.

```