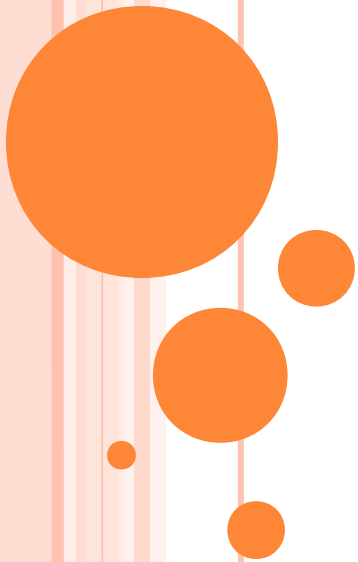
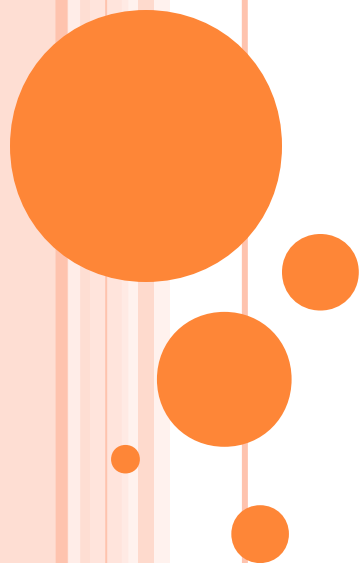


BESTE DATU MOTA BATZUK: ENUMERATUAK ETA AZPIMOTAK



ENUMERATUAK



ENUMERATU MOTA

- Mota honen definizio batzuk adibide gisa:
 - **Type** t_asteko_egunak **is** (astelehena, asteartea, asteazkena, osteguna, ostirala, larunbata, igandea);
 - **Type** t_egoera **is** (piztuta, martxan, itzalita)
 - **Type** t_koloreak **is** (gorria, urdina, beltza, orlegia)



Type t_asteko_egunak **is** (astelehena, asteartea, asteazkena, osteguna, ostirala, larunbata, igandea);

Type t_egoera **is** (piztuta, martxan, itzalita)

Type t_koloreak **is** (gorria, urdina, beltza, orlegia)

- Aldagai batzuen deklarazioa:

- Eguna: t_asteko_eguna;
- Egoera: t_egoera;

- Esleipena

- Eguna:= ostirala;
- Egoera:= piztuta;



○ Egikariketak


- If eguna = osteguna then ...
- If egoera /= martxan then ...

○ Atributuak

- T'pred, T'succ, T'first, T'last, T'pos, T'val, T'image, T'value, T'min, T'max

Non T enumeratu motatakoren bat izango den, adibidez:

T_asteko_egunak'pos(astelehena) = 0 → 0 posizioan dago?
T_asteko_egunak'pred(igandea) → larunbata itzuliko luke!
T_asteko_egunak'succ(igandea) → ERROREA!
T_asteko_egunak'first → astelehena itzuliko luke!
T_asteko_egunak'pos(igandea) → 6 itzuliko luke (integer)
T_asteko_egunak'min(igandea,astelehena) → astelehena itzuliko luke



- Hurrengo deklarazioak emanda:
 - `Kol:t_koloreak;`
 - `Eguna:t_asteko_egunak;`
- Ez da onartzen:
 - `Kol:= astelehena; --Errorea`
 - `Eguna:= t_asteko_egunak'pred(astelehena);`
 - `Eguna:= t_asteko_egunak'succ(igandea);`



Zeharkatzeko modu bat

```
with Ada.Text_IO;  
use Ada.Text_IO;  
procedure A is  
    type asteko_egunak is  
        (Astelehena, Asteartea, Asteazkena, Osteguna, Ostirala, Larunbata, Igandea);  
  
    package asteEnumeration_IO is Ada.Text_IO Enumeration_IO(asteko_egunak);  
  
    i:integer;  
    begin  
        i:=0;  
        loop exit when i > 6;  
            asteEnumeration_IO.Put(asteko_egunak'Val(i));  
            new_line;  
            i:=i+1;  
        end loop;  
    end;
```



Nola zeharkatzen dut nik aldagai bat bere mota enumeratua denean, adibidez:

```
type asteko_egunak is (Astelehena,Asteartea,Asteazkena,Osteguna,Ostirala,Larunbata,Igandea);

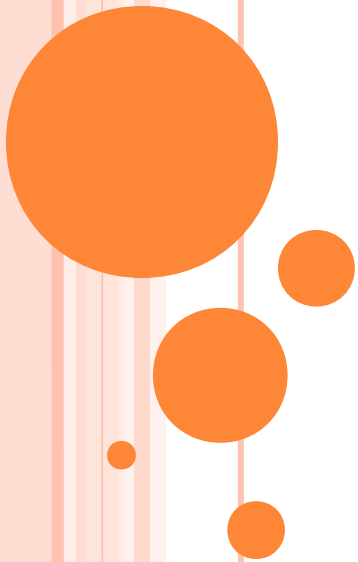
with Ada.Text_IO;
use Ada.Text_IO;
procedure enumeratu_bat_inprimatzeko is
begin
  for i in asteko_egunak'first .. asteko_egunak'last loop
    Put_Line(asteko_egunak'Image(i));
  end loop;
end;
```

Inprimatzeko beste modu bat:

```
with Ada.Text_IO;
use Ada.Text_IO;
procedure A is
  type asteko_egunak is (Astelehena,Asteartea,Asteazkena,Osteguna,Ostirala,Larunbata,Igandea);
  package asteEnumeration_IO is new Ada.Text_IO Enumeration_IO(asteko_egunak);
begin
  for i in asteko_egunak loop
    asteEnumeration_IO.Put(i);
    new_line;
  end loop;
end;
```



AZPIMOTAK



- Azpimota baten definizioa

- **Subtype** t_talde_rangoa **is** Character **range** 'A'..'D'
- **Subtype** t_nota-rangoa **is** Integer **range** 0..10;
- **Subtype** t_lan_egunak **is** t_asteko_egunak **range** astelehena..ostirala;



- Aldagaien deklarazioa:
 - Talde1: t_talde_rangoa;
 - Lan_eguna1: t_lan_egunak;
 - Nota1: t_nota_rangoa;

- Esleipena:
 - Talde1:= 'A';
 - Nota:= 7;
 - Lan_eguna1 := asteazkena;



ERREGISTRO BATEN AGREGATUAK

Erregistro motatako aldagai bati balioak emateko balio dute:

```
Type info_ikasle is record  
    expe_zenb: natural;  
    izena, abizena: string(1..30);  
    Kurtsoa: natural;  
    taldea: character;  
End record;
```

- Est1: info_ikasle:= (1150, "Iñaki", "Zuñiga", 1, 'D');
- Edo eremu identifikadoreak erabilita:
 est2:= (exped_zenb=>1520,
 izena=>"Luis",
 abizena => "Arregi",
 kurtso=>2, taldea=>'D');



ARRAYEN AGREGATUAK

- **type** t_osokoak **is array**(1..10) **of** Integer;

zenbakiak: t_osokoak;

zenbakiak:= (0, 1, 0, 1, 0, 1, 0, 1, 0, 1);

zenbakiak:= (1 => 0, 2 => 1, 3 => 0, others => 0);

