

4.2 Gaia. Beste datu egitura batzuk

eman ta zabal zazu

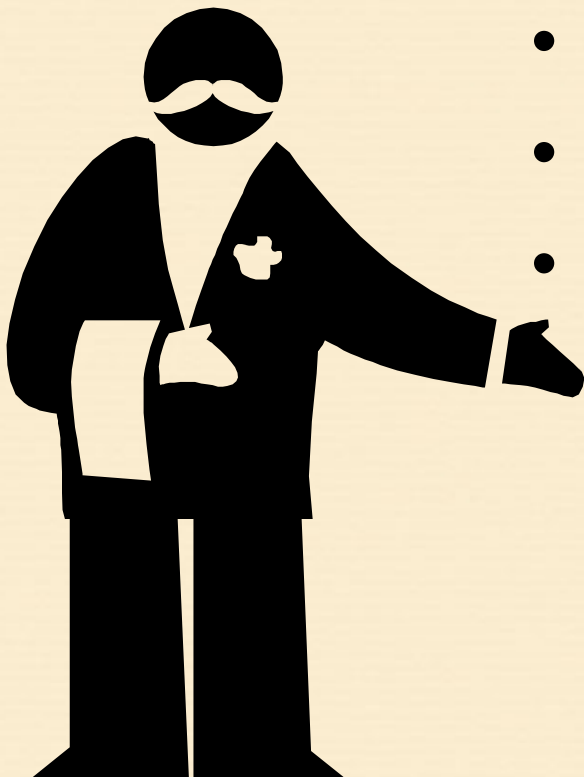


Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Aurkibidea

- **Gaiaren helburuak**
- Motibazioa
- Enumeratu motak
- Azpimotak
- Agregatuak



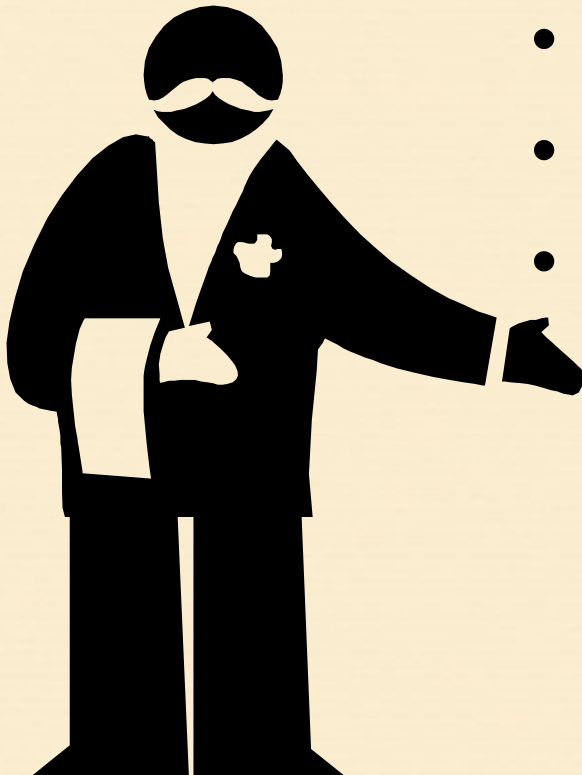
Gaiaren helburuak

- Bi datu mota gehigarri ezagutu, egituren diseinuan erabilgarriak izango direnak
 - Enumeratuak
 - Azpimotak
- Erregistroak eta bektoreak hasieratzeko agregatuen erabilera



Aurkibidea

- Gaiaren helburuak
- **Motibazioa**
- Enumeratu motak
- Azpimotak
- Agregatuak



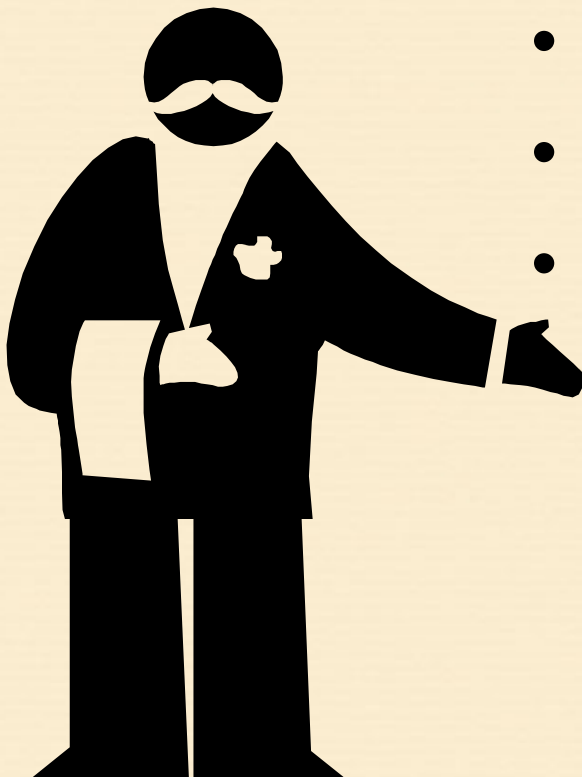
Motibazioa

- 4.2 gaian ikusitakoa bukatzen du, eta ondorioz ADA lengoaia bakarrik ikusiko dugu



Aurkibidea

- Gaiaren helburuak
- Motibazioa
- **Enumeratu motak**
- Azpimotak
- Agregatuak



Enumeratu motak

- Aldagaiek izan ditzaketen balioak (bakarrak) zerrendatuz definitzen dira
 - Boolean aurredefinitutako enumeratu mota bat da

```
type T_asteko_eguna is (Astelehena, Asteartea, Asteazkena, Osteguna, Ostirala,  
Larunbata, Igandea);  
type T_kolorea is (Urdina, Gorria, Berdea, Horia);  
type T_egoera is (Piztuta, Martxan, Itzaltzen, Itzalita);  
type T_jokalari_pos is (Antolatzailea, Eskolta, Hegalekoa, Hegaleko_pibota, Pibota);
```



Enumeratuen erabilera

- Orain arte ikusitakoen moduan erabiltzen dira enumeratuak (adibidez boolearrak)

```
-- Aldagaien erazagupena
Eguna: T_asteko_eguna;
Egoera: T_egoera;

-- Balio literalen esleipena
Eguna:= Osteguna;
Egoera:= Itzalita;

-- Aldagaien erabilera
while Egoera /= Martxan loop...
if Eguna = Osteguna then...
```



Atributuak (I)

- Enumeratuaren lehen eta azken balioak
 - T_asteko_eguna'first → Astelehena
 - T_asteko_eguna'last → Igandea
- Aurrekoa eta ondorengoa
 - T_asteko_eguna'pred(Igandea) → Larunbata
 - T_asteko_eguna'succ(Osteguna) → Ostirala
 - T_asteko_eguna'succ(Igandea) → ERROREA



Atributuak (eta II)

- Osokoetatik eta osokoetara bihurketa (0n hasten da)
 - T_asteko_eguna'pos(Asteartea) → 1
 - T_asteko_eguna'val(3) → Osteguna
- Stringetatik eta stringetara bihurketa
 - T_asteko_eguna'image(Asteartea) → “Asteartea”
 - T_asteko_eguna'value(“Osteguna”) → Osteguna



Atributuak (eta II)

```
with datuak;  
use datuak;  
with ada.text_io;  
use ada.text_io;  
procedure nagusia is  
    egunak: T_asteko_eguna;  
    package nire_IO is new Ada.Text_IO Enumeration_IO(T_asteko_eguna);  
begin  
    if (T_asteko_eguna'image(Asteartea)="ASTEARTEA") then      – STRING konparaketa  
        put(T_asteko_eguna'image(Asteartea));  
    end if;  
    if (T_asteko_eguna'value("Osteguna")=ASTELEHENA) then      – ENUMERATU konparaketa  
        nire_IO.put(T_asteko_eguna'value("Osteguna"));  
    end if;  
end nagusia;
```



Enumeratuen balioak inprimatu

- Hiru modu ikusiko ditugu enumeratuak korritu eta euren balioak pantailatik inprimatzeko

```
with Ada.Text_IO;  
use Ada.Text_IO;  
procedure inprimatu1 is  
  type T_jokalari_pos is (Antolatzailea, Eskolta, Hegalekoa,  
    Hegaleko-pibota, Pibota);  
  package tjp_IO is new Ada.Text_IO Enumeration_IO(T_jokalari_pos);  
begin  
  for Ind in 0 .. T_jokalari_pos'pos(T_jokalari_pos'last) loop  
    tjp_IO.put(T_jokalari_pos'val(Ind));  
    new_line;  
  end loop;  
end inprimatu1;
```



Enumeratuen balioak inprimatu

```
with Ada.Text_IO;  
use Ada.Text_IO;  
procedure inprimatu2 is  
  type T_jokalari_pos is (Antolatzailea, Eskolta, Hegalekoa,  
    Hegaleko-pibota, Pibota);  
  package tjp_IO is new Ada.Text_IO Enumeration_IO(T_jokalari_pos);  
begin  
  for Ind in T_jokalari_pos loop  
    tjp_IO.put(Ind);  
    new_line;  
  end loop;  
end inprimatu2;
```



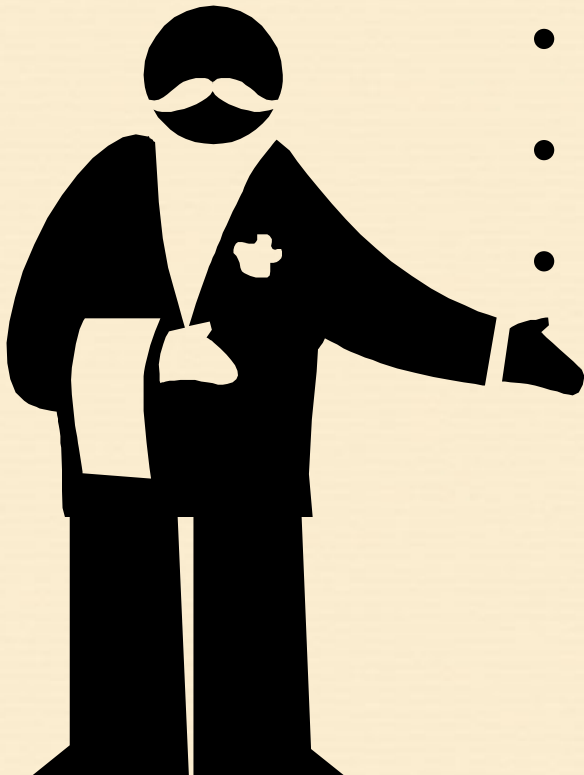
Enumeratuen balioak inprimatu

```
with Ada.Text_IO;  
use Ada.Text_IO;  
procedure inprimatu3 is  
  type T_jokalari_pos is (Antolatzailea, Eskolta, Hegalekoa,  
Hegaleko-pibota, Pibota);  
  begin  
    for Ind in T_jokalari_pos'first..T_jokalari_pos'last loop  
      put_line(T_jokalari_pos'image(Ind));  
    end loop;  
  end inprimatu3;
```



Aurkibidea

- Gaiaren helburuak
- Motibazioa
- Enumeratu motak
- **Azpimotak**
- Agregatuak



Azpimotak

- Beste mota bateko balioen azpimultzo gisa definitzen dira
 - Natural eta Positive Integer-en azpimotak dira
- Azpimota eta erlazionatutako mota bateragarriak dira
 - Hau da, ez dira mota ezberdinak

```
subtype T_talde_tartea is Character range 'A'..'D';  
subtype T_nota is Integer range 0..10;  
  
type T_asteko_eguna is (Astelehena, Asteartea, Asteazkena, Osteguna, Ostirala,  
Larunbata, Igandea);  
subtype T_laneguna is T_asteko_eguna range Astelehena..Ostirala;
```



Azpimoten erabilera

- Azpimotak ere orain arte ikusitakoen moduan erabiltzen dira

```
-- Aldagaien erazagupena
Laneguna: T_laneguna;
Taldea: T_talde_tartea;
Nota_OP: T_nota;

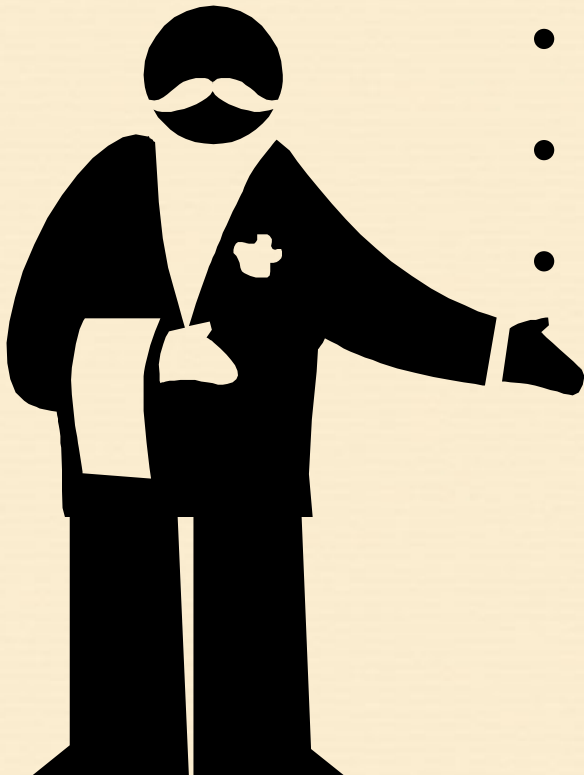
-- Balio literalen esleipena
Laneguna:= Osteguna;
Taldea:= 'A';
Nota_OP:= 7;

-- Aldagaien erabilera
while Laneguna /= Ostirala loop...
    if Nota < 5 then...
```



Aurkibidea

- Gaiaren helburuak
- Motibazioa
- Enumeratu motak
- Azpimotak
- **Agregatuak**



Agregatuak

- Ez dira datu egiturak, baizik eta erregistro zein bektoreei balioak esleitzeko tresnak dira
 - Alternatibean, begiztak edo banakako esleipen indibidualen bidez egin beharko litzateke



Erregistroen agregatuak

```
type Ikaslea is record  
    Esp_zenb: Positive;  
    Izena, Abizena: String(1..30);  
    Kurtsoa: Natural;  
    Taldea: T_talde_tartea;  
end record;
```

```
Ikas1, Ikas2: Ikaslea;  
  
Ikas1:= (1150, "Iñaki      ", "Zuñiga   ", 1, 'D');  
  
-- Eremuen identifikadoreak erabiliz  
Ikas2:= (Esp_zenb => 1520, Izena => "Luis      ",  
         Abizena => "Arregi    ", Kurtsoa => 2, Taldea => 'A');
```



Arrayen agregatuak

```
type Osokoen_sekuentzia is array(1..10) of Integer;
```

```
Zenbakiak, Zenbaki_berdinak, Denak_bat: Osokoen_sekuentzia;  
  
Zenbakiak:= (1,0,1,0,0,0,0,0,1);  
Zenbaki_berdinak:= (1..1=>1, 3..3=>1, 10..10=>1, others =>0);  
Denak_bat:= (others=>1);
```

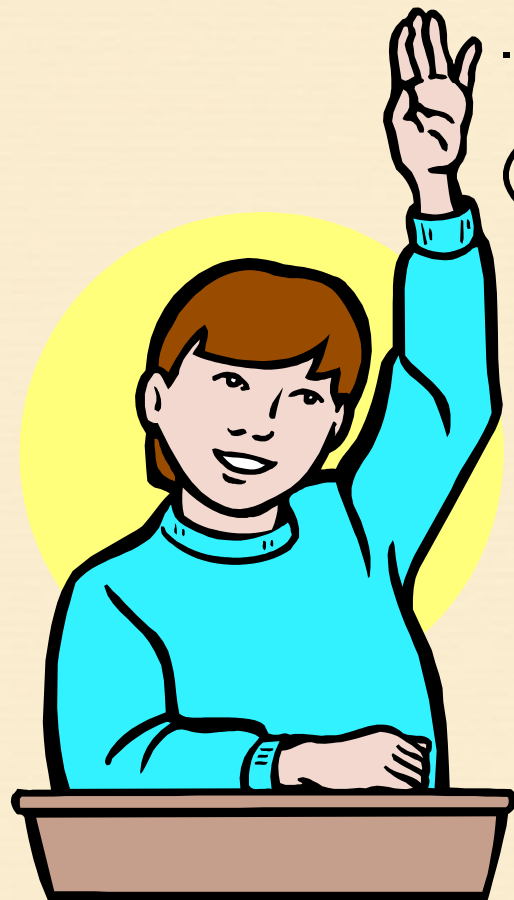
```
-- Slice-ak  
Zenbakiak(1..3):= (0,1,2);  
Zenbakiak(4..6):= Zenbaki_berdinak(1..3);  
  
KONTUZ: Zenbakiak(7..10) ez dute baliorik!
```

Arrayen arrayak agregatuak

```
type Osokoen_sekuentzia is array(1..10) of Integer;  
type Osokoen_matrizea is array(1..10) of Osokoen_sekuentzia;
```

```
matrizea: Osokoen_matrizea;  
  
for i in 1..10 loop  
    matrizea(i) := (others => 0);  
end loop;  
matrizea(2) := (2, 1, 2, others => 5);
```

KONTUZ! Soilik arrayen arrayekin!



Galderarik?