DATU EGITURAK

Erregistroak

- Datu mota hau erabiltzen da izen baten pean mota desberdineko datuak lotzeko.

- Erregistro baten definizioa:

```
type Ikasle is record
        exp_zenb: Integer;
        izen, abizena: String(1..30);
        kurtsoa: Integer;
        taldea: Character;
end record;
```

Eragiketak erregistroekin

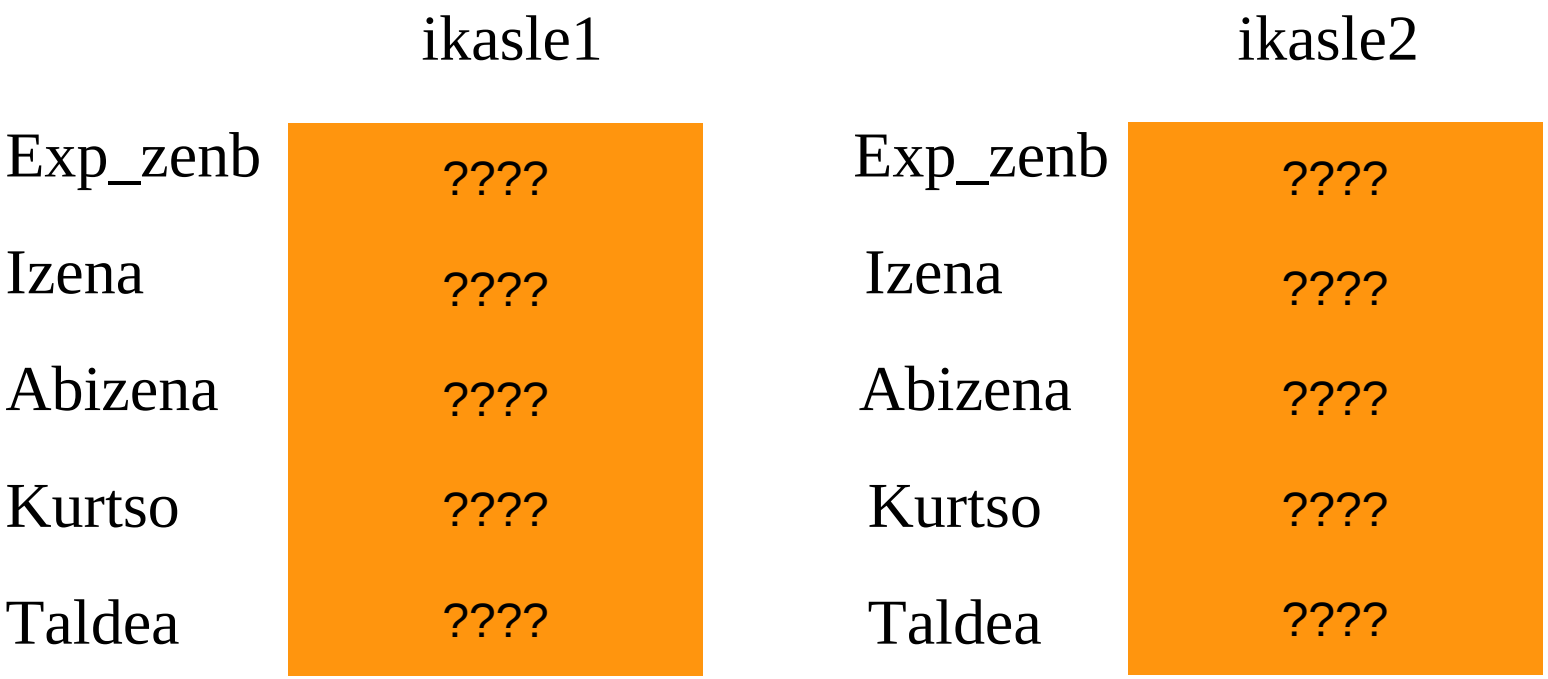
- Erregistro motako aldagai batek, mota desberdinetako datuak barnera ditzake (edo mota berdinekoak)

- Erregistro baten definizioan:

- Izen bat esleitzen zaie erregistroko eremu bakoitzari
- Eremu bakoitzari dagokion mota adieraziko da

- Erregistro motatako aldagaien deklarazioa:

  ikasle1, ikasle2: Ikasle;



```
type ikasle is record
        exp_zenb: Integer;
        izen, abizena: String(1..30);
        kurtsoa: Integer;
        taldea: Character;
end record;
```

Erregistroko **eremuen** atzipena:

```
get(ikasle1.exp_zenb);
get(ikasle1.izena);
ikasle1.abizena := “Cousteau          “  -- debe haber 30 caracteres!!!
Ikasle1.taldea := 'A';
Ikasle2.kurtso := 1;
Ikasle1.kurtso:= ikasle2.kurtso;
```

- Erregistro motatako aldagaien deklarazioa:

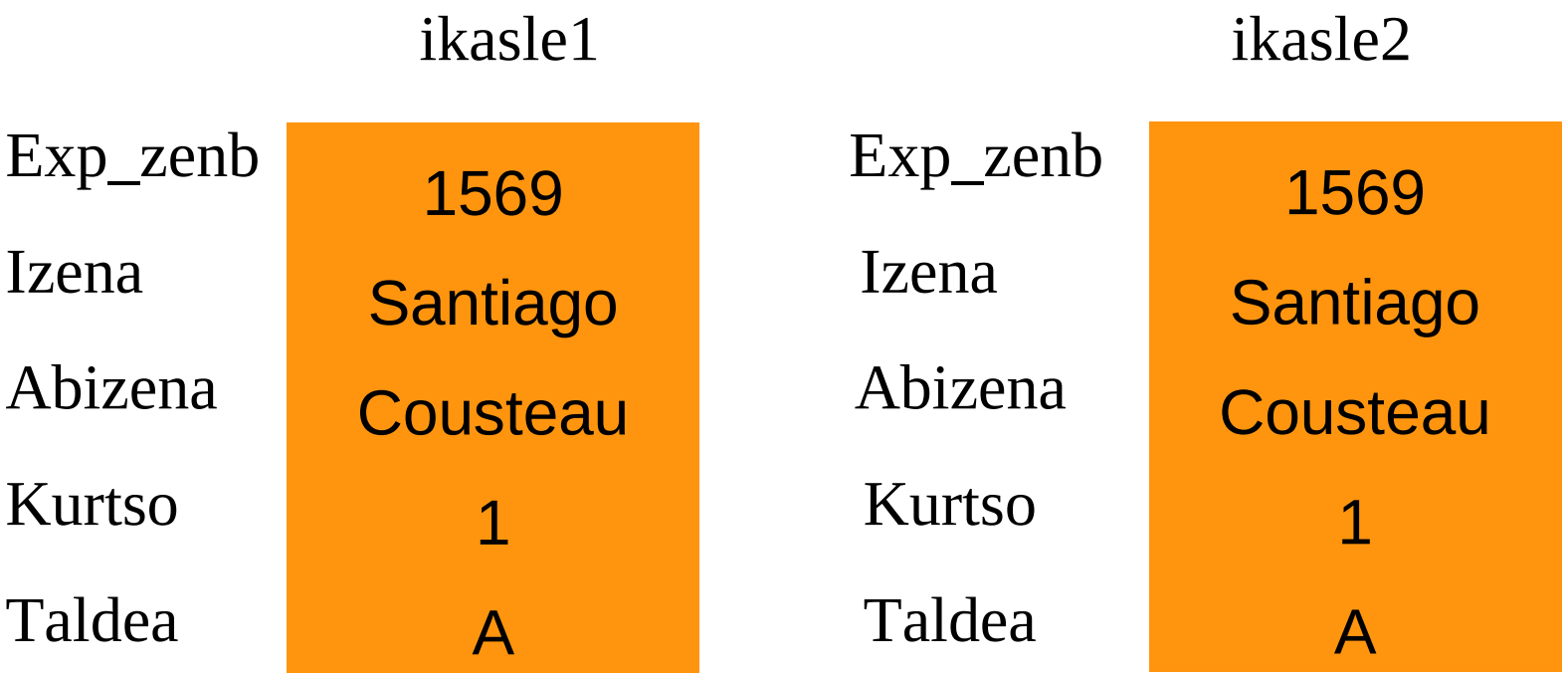
  ikasle1, ikasle2: Ikasle;

	ikasle1
Exp_zenb	1569
Izena	Santiago
Abizena	Cousteau
Kurtso	1
Taldea	A

	ikasle2
Exp_zenb	????
Izena	????
Abizena	????
Kurtso	1
Taldea	????

Erregistroak: esleipenak

- Ikasle2 := Ikasle1;

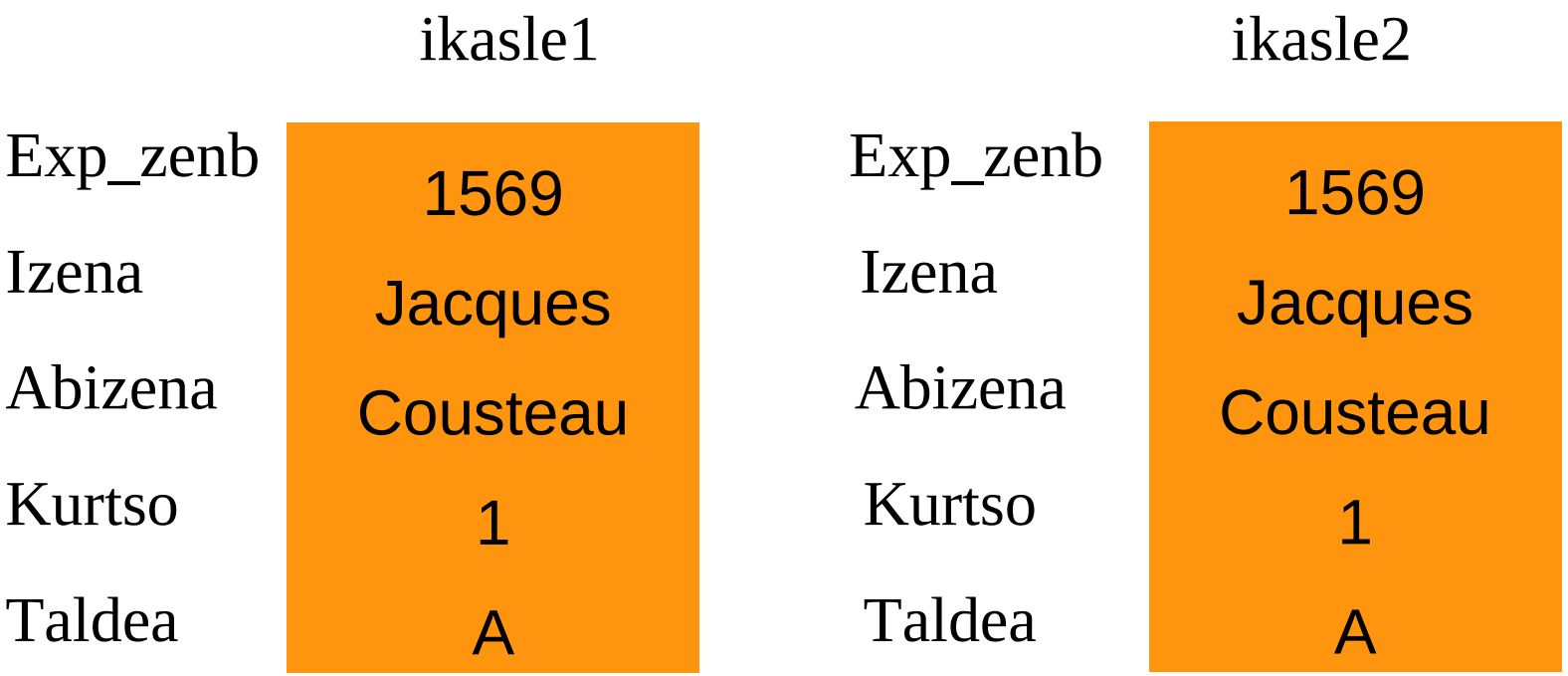


ikasle2.izena := “Jacques “;
Ikasle2.kurtso:= 5;

ikasle2

Exp_zenb	1569
Izena	Jacques
Abizena	Cousteau
Kurtso	5
Taldea	A

- Ikasle1 := Ikasle2;



ERREGISTROEKIN EGIN DAITEZKEN ERAGIKETAK

- Erregistro baten eremu baten atzipena("."):

  Ikasle1.izena := “Aitor ";

- Esleitu erregistro batek daukan informazioa beste bati:

  Ikasle1 := Ikasle2;

ERREGISTROEKIN EGIN DAITEZKEN ERAGIKETAK

- Konparatu bi erregistro

  Ikasle1 = Ikasle2

  Ikasle1 /= Ikasle2

- (<, >, <= y >= ez dute zentzurik)

MOTEN KONKORDANTZIA

- **type Ikasle is record**
  Exp_zenb: Integer;
  Izen,abizena: String(1..30);
  kurtsoa: Integer;
  taldea: Character;
  end record;

- **type Produktua is record**
  izen: Character;
  salneurria: Integer;
  kopurua: Integer;
  end record;

- Aldagaien deklarazioa:

  Ikasle1, Ikasle2: Ikasle;

  prodA, prodB, prodC: Produktu;

- Esleipen posible:

  Ikasle1 := Ikasle2;

  prodA := ProdB;

  Ikasle1.kurtso := prodA.kopurua;

- Esleipen ezinak:

  Ikasle1 := prodA;

  prodA.kopurua := ikasle1.izena;

  ikasle1.izena := prodB.izena;

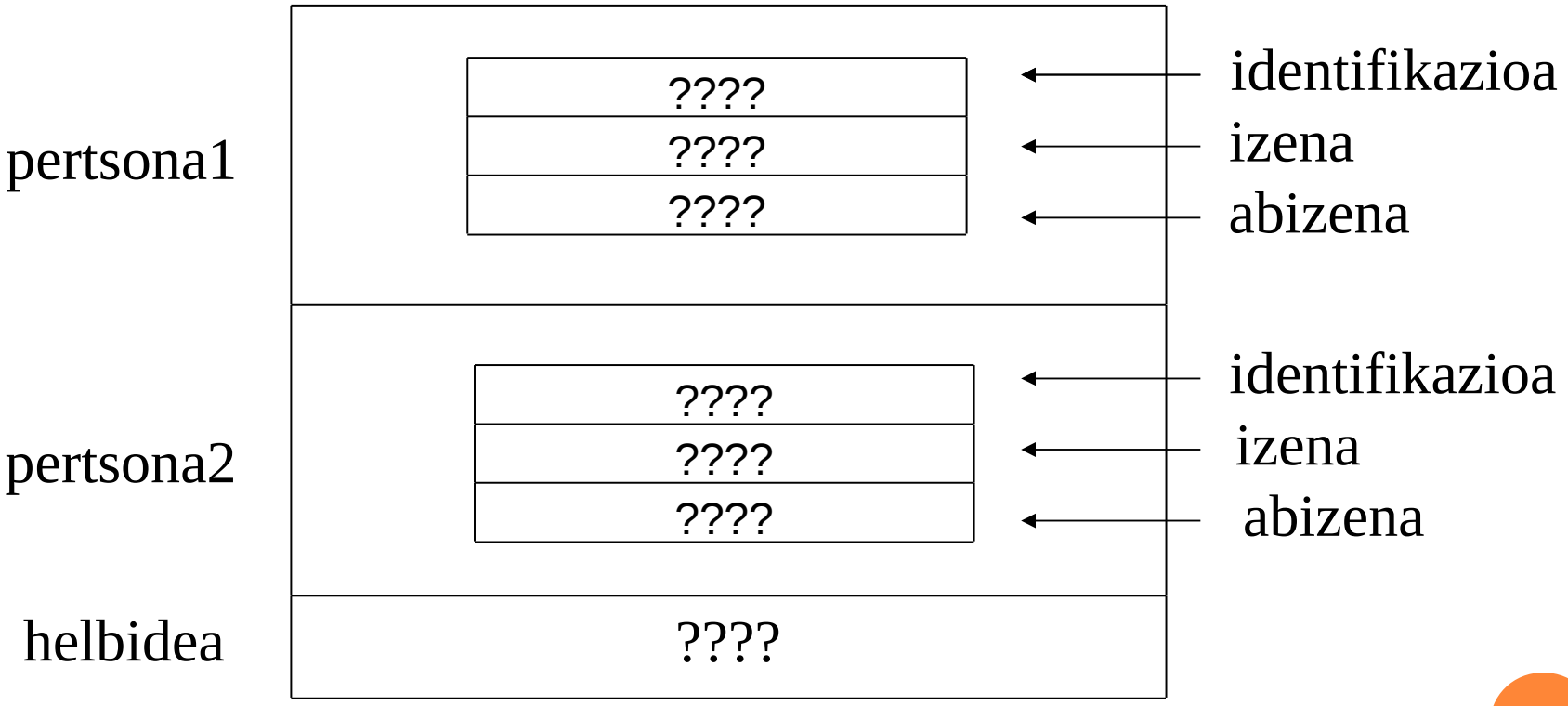
Erregistroak beste erregistro batzuekin

Moten deklarazioa:

```
type t_pertsona is record
    identifikazioa: Integer;
    izena, abizena: string(1..20);
end record;

type t_pareja is record
    pertsona1, pertsona2: t_persona;
    helbidea: string(1..30);
end record;
```

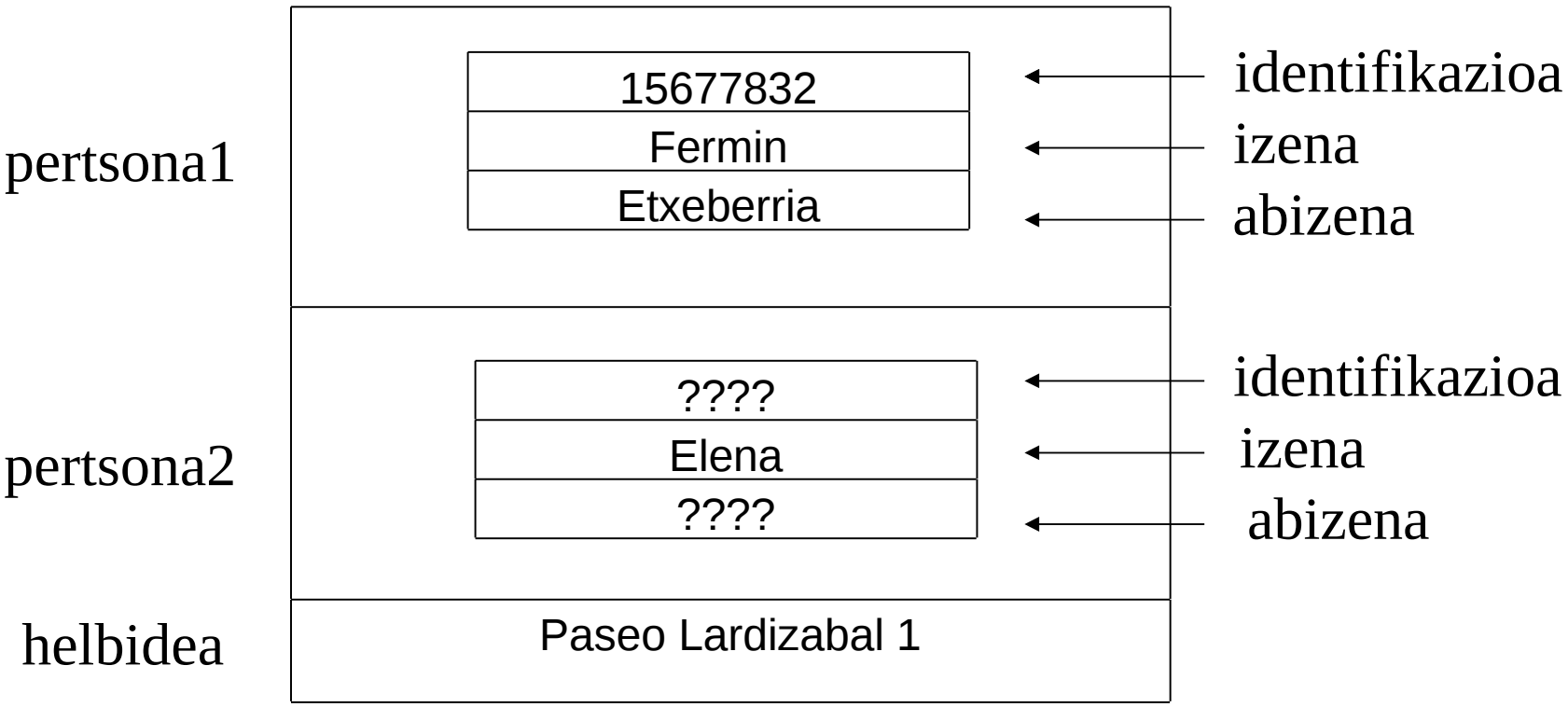
Moten deklarazioa:
pareja: t_pareja



Erregistroak beste erregistro batzuekin

Esleipenak:

```
pareja.helbidea := "Paseo Lardizabal 1";
pareja.pertsona1.identifikazioa := 15677832;
pareja.pertsona1.izena := "Fermin ";
pareja.pertsona1.abizena := "Etxeberria";
pareja.pertsona2.izena := "Elena ";
```


pertsona1
15677832
Fermin
Etxeberria
identifikazioa
izena
abizena
pertsona2
????
Elena
????
identifikazioa
izena
abizena
helbidea
Paseo Lardizabal 1

```
procedure irakurri (P: out t_pertsona) is
--pre: datu mota, zenbaki batez eta bi stringek osatuta dago eta pertsona identifikatzen du
--pos: se han leído los datos en P
begin
        Get(P.identifikazioa);
        Get(P.izena);
        Get(P.abizena);
end
```

```
procedure Idatzi (P: in t_pertsona) is
--pos: P pertsonaren datuak idatzi dira
begin
        Put(P.identifikazioa);
        Put(P.izena);
        Put(P.abizena);
end
```

```
procedure asignatu_pareja(pareja: in out t_pareja; Pos: in numero;
Per: in t_persona) is

--pre: Pos-en balioa 1 o 2 da
--pos: pertsona1 eremuak Per-en balioa hartuko du baldin eta Pos 1
bada, bestela pertsona2-ak jasoko du

begin
        if pos = 1 then
            pareja.pertsona1:=per;
        else
            pareja.pertsona2:=per;
        end if
end asignatu_pareja;
```

Erregistroak erabiltzearen abantailak

Beharrezko aldagai kopurua murrizten du:

pertsona1, pertsona2, pertsona3: t_pertsona

Bestela:

identifikazio1, identifikazio2, identifikazio3: integer
izena1, izena2, izena3: string(1..20)
abizena1, abizena2, abizena3: string(1..20)

Erregistroak erabiltzearen abantailak

Deklaratu behar diren aldagai kopurua murrizten du:

Procedure sartu (pertsona: **in** t_pertsona; zerrenda: **in out** t_lista_pertsona) is

Bestela:

Procedure sartu (identifikazioa: in integer; izena, abizena: **in** string(1..10); zerrenda: **in out** t_lista_pertsona) is

Non deklaratzen dira motak

Aukera 1

```
procedure adibidea ..... is

-- moten deklarazioa:

   type ikasle is record
   .....
   end record;

      type tabla15 is array
      .....;
      ....
      .... (beste deklarazioak)

-- aldagaien deklarazioa:
     A: ikasle;
     T1, T2: tabla15;

begin
     ....
end adibidea;
```

OHARRA:

- Modu honetan deklaratutako aldagaiak (ikasle, tabla15), soilik Azpiprogrametan edo programa berean erabil daitezke

- Deklaratutako azpiprogramekin ere berdina gertatuko zen!!!!

Non deklaratzen dira motak

Beste modu bat, moten deklarazioak modulu batean deklaratzea Izango da (edo paketeetan)

Abantailak: deklarazio berdinak beste programek ere erabili ahal izango dute

definizioak.ads

Aukera 2:

```
package definizioak is
-- moten deklarazioa
     type ikasle is record
          ......
     end record;
     Type tabla15 is array .....;
end definizioak;
```

Modu honetara, azpiprogarma nahiz programa desberdinek erabili ahalko dute:

X.adb

```
With definizioak;
Use definizioak;
Procedure X .....
        A: ikasle;
        T1, T2: tabla15;
begin
    .....
end X;
```

Y.adb

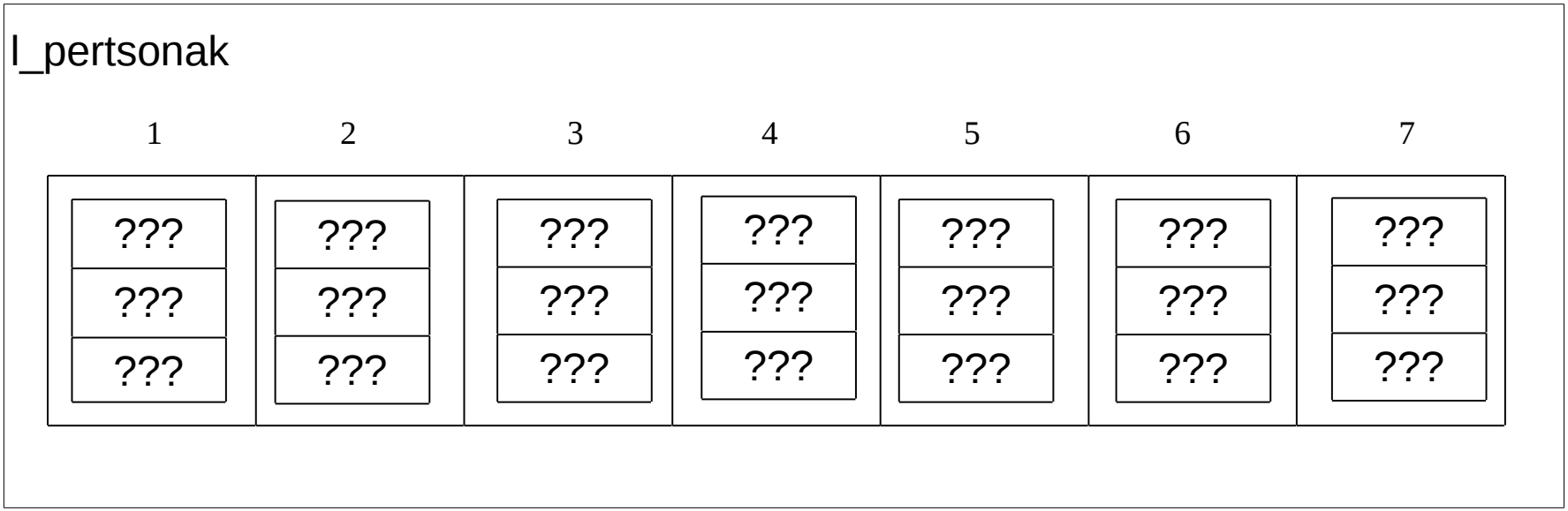
```
With definizioak;
Use definizioak;
Procedure Y .....
        B: ikasle;
        T6, T8: tabla15;
begin
    .....
end Y;
```

Arrayak eta erregistroak tartekatzen

```
type t_pertsona is record
        identifikazioa: Integer;
        izena, abizena: string(1..20);
end record;
type t_lista_pertsonak is array (1 ..7) of t_pertsona;
```

Arrayak eta erregistroak tartekatzen

l_pertsonak: t_lista_pertsonak;



Arrayak eta erregistroak tartekatzen

l_pertsonak(3).izena:= “Ana   ”;

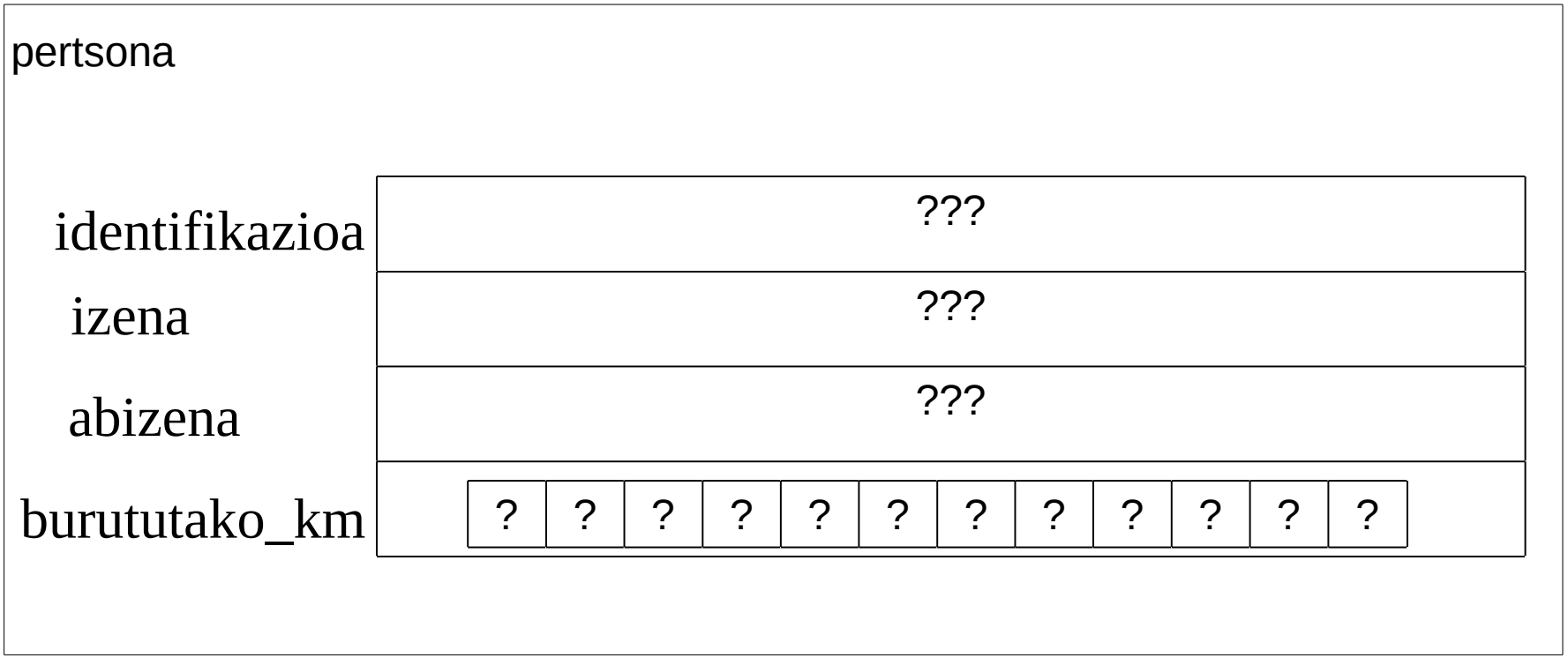
l_pertsonak

1	2	3	4	5	6	7
???	???	???	???	???	???	???
???	???	Ana	???	???	???	???
???	???	???	???	???	???	???

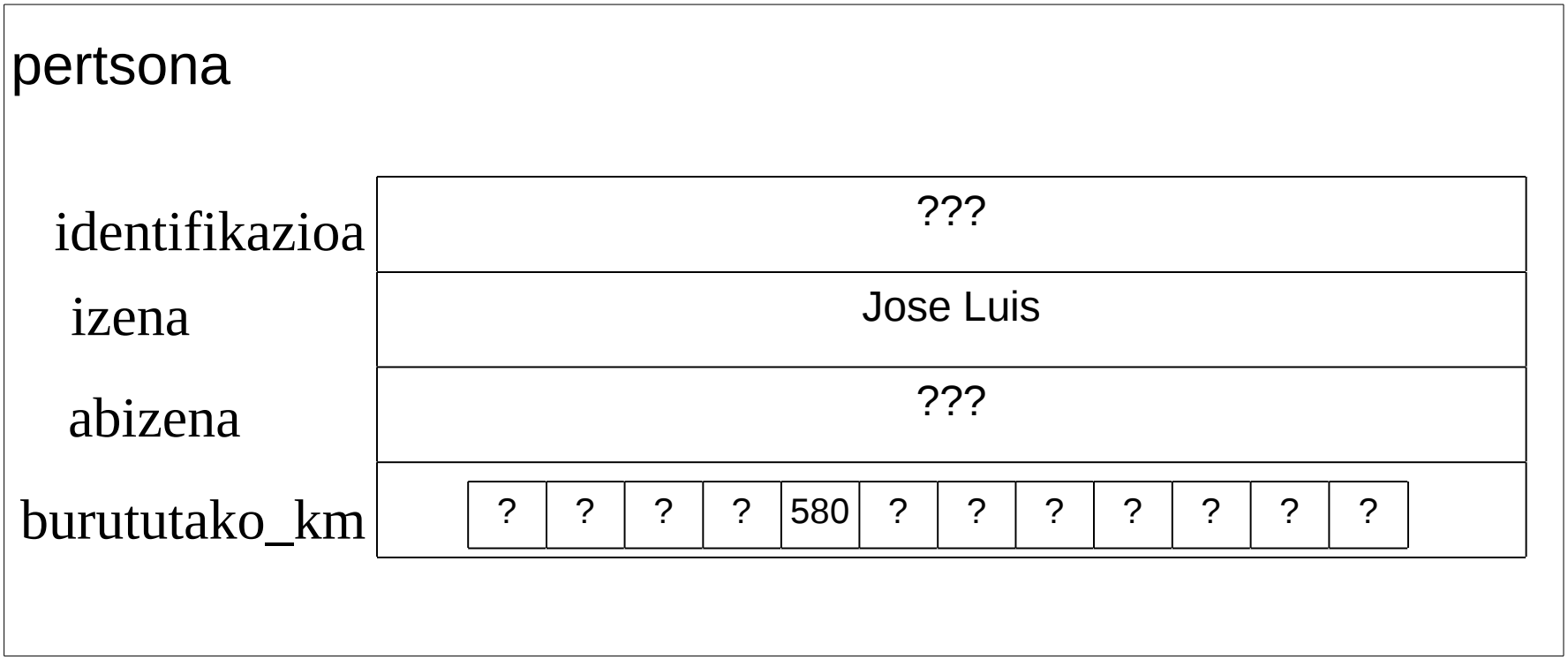
```
type t_lista_km is array (1 .. 12) of Integer;

type t_pertsona is record
      identifikazioa: Integer;
      izena, abizena: string(1..20);
      burututako_km: t_lista_km;
end record;
```

pertsona: t_pertsona



pertsona.izena:= “Jose Luis “
pertsona.burututako_km(5):= 580;



Adibideak:

Datu mota bat definitu ezazu non, hurrengo klaseko afarira joango diren ikasleen datuak gordeko dituen. Ikasle bakoitzeko, izena eta abizena gorde nahi dugu:

```
Type ikasle is record
    izena: string (1..30);
    abizena: string (1..30);
End record;
Type T_ikasleak is array (1..120) of ikasle;
```

```
procedure ikasleak_gestionatu is
      klaseko_ikasleak: T_ikasleak;
      izena: string (1..30);
begin
      egitura_bete(klaseko_ikasleak);
      get(izena);
      loop exit when izena= “                                “;
            ikaslea_borratu(klaseko_ikasleak, izena);
            get(izena);
      end loop;
end ikasleak_gestionatu;
```

```
procedure ikaslea_borratu(eguneratzeko_ikasleak: in out T_ikasleak;
                                        borratzeko_izena: in string(1..30)) is

ind:integer;
begin
        ind:=ikaslearen_pos(eguneratzeko_ikasleak,borratzeko_izena);
        borratu(eguneratzeko_ikasleak, ind);

end eliminar_alumno;
```

```
function ikaslearen_pos(eguneratzeko_klasea: in T_ikasleak;
                        izena_borratzeko: in string(1..30)) return integer is;

indizea:integer;
bilatua:boolean:=false;
begin
indizea:=1;
Loop exit when indizea > eguneratzeko_klasea'last or bilatua;

     if(eguneratzeko_klasea(indizea).izena = izena_borratzeko) then
bilatua:=true;
      else
            indizea:=indizea+1;
      end_if;

end_loop;

return(indizea);
end ikaslearen_pos;
```

```
procedure borratu (eguneratzeko_klasea: in out T_ikasleak;
                           ind: in integer) is
indizea:integer:=ind;
begin
       loop exit when indizea = eguneratzeko_klasea'last;
        if (indizea=ind) then
                    eguneratzeko_klasea(indizea):=eguneratzeko_klasea(indizea+1);
        end if;
              indizea:=indizea+1;
       end_loop;
end eliminar;
```

OHARRA: Baina ikasle bat borratu badugu, iada ez ditugu izango 120 ikasle, baizik eta 119!!! Gainera, programa exekutatzen den bakoitzean geroz eta ikasle gutxiago izango ditugu, beraz ..... zentzurik bai 'last bitarte errepikatzea????? ¿Nola konpondu hau?

ARRAYA ERREGISTRO BATEAN TXERTATUZ!!!!!!!!!!

- Datu mota hau izan ordez:

```
        type ikasle is record
        izena: string(1..30);
        abizena: string(1..30);
    end record
    type T_ikasleak is array (1..120) of ikasle;
```

- Honakoa izango dugu:

```
        type ikasle is record
        izena: string(1..30);
        abizena: string(1..30);
    end record
    type T_ikasleak is array (1..120) of ikasle;
    type ikasleen_zerrenda is record
        zenbat: integer;
        ikasleen_tabla: T_ikasleak
    end record;
```

```
procedure ikasleak_gestionatu is
      klaseko_ikasleak: ikasleen_zerrenda;
      izena: string (1..30);
begin
      klaseko_ikasleak.zenbat:=0;
      egitura_bete(klaseko_ikasleak);
      get(izena);
      loop exit when izena= “                                “;
            ikaslea_borratu(klaseko_ikasleak, izena);
            get(izena);
      end loop;
end ikasleak_gestionatu;
```

```
procedure egitura_bete(egunera_ikasleak: out ikasleen_zerrenda) is

izen, abizen:string(1..30);
begin
    get(izen);
    get(abizen);
        loop exit when izen="                    ";
            egunera_ikasleak.zenbat:=egunera_ikasleak.zenbat+1;
            egunera_ikasleak.ikasleen_tabla(egunera_ikasleak.zenbat).izena=izen;
            egunera_ikasleak.ikasleen_tabla(egunera_ikasleak.zenbat).abizena=abizen;
        end loop;

end egitura_bete;
```

```
procedure ikaslea_borratu(eguneratzeko_ikasleak: in out ikasleen_zerrenda;
                                        borratzeko_izena: in string(1..30)) is

ind:integer;
begin
        ind:=ikaslearen_pos(eguneratzeko_ikasleak,borratzeko_izena);
        borratu(eguneratzeko_ikasleak, ind);

end ikaslea_borratu;
```

```
function ikaslearen_pos(eguneratzeko_klasea: in ikasleen_zerrenda;
                        izena_borratzeko: in string(1..30)) return integer is;

indizea:integer;
bilatua:boolean:=false;
begin
indizea:=1;
loop exit when indizea > eguneratzeko_klasea.zenbat or bilatua;

     if(eguneratzeko_klasea.ikasleen_tabla(indizea).izena = izena_borratzeko)
then                 bilatua:=true;
      else
            indizea:=indizea+1;
      end_if;

end_loop;

return(indizea);
end ikaslearen_pos;
```

```
procedure borratu(eguneratzeko_ikasleak: in out ikasleen_zerrenda;
                                        ind: in integer) is

begin
        ezkerretara_mugitu(eguneratzeko_ikasleak, ind);
        eguneratzeko_ikasleak.zenbat:=eguneratzeko_ikasleak.zenbat-1;
end borratu;
```

```
procedure ezkerretara_mugitu(egun_ikasleak: in out ikasleen_zerrenda;
                                        ind: in integer) is

begin
    loop exit when indizea=egunera_ikasleak.zenbat;
         egun_ikasleak.ikasleen_tabla(indizea):=egun_ikasleak.ikasleen_tabla(indizea+1);
         indizea:=indizea+1;
   end loop;
end ezkerretara_mugitu;
```

Ariketa:

Sarrera estandarrean 10 saltzaileen datuak ditugu. Saltzaile bakoitzeko, berauen zenbaki identifikadorea, izena, abizena eta burutu dituzten kilometroak adierazteko 5 zenbaki ditugu:

123	Jorge	Pastor	0	48	100	500	230
600	Iñigo	Balda	800	1000	0	900	2500
...							
...							
...							

a) Datu hauek gordetzeko datu egitura definitu ezazu.

b) Garatu azpiprograma bat non, saltzaileen zerrenda emanda, kilometro baliorik handiena itzuliko duena eta saltzaileen izena.

Ariketa:

Egokitu ezazu aurreko datu mota, non, ikasleen izenak eta hauen notak gordetzeko erabiliko den (bektoreak ez du zertan guztiz beteta egon behar).

Datu egitura definitu ondoren, inplementa ezazu azpiprograma bat non ikasleen lista emanda, batazbesteko nota globala lortuko den.

a) Inplementa ezazu azpiprograma bat non, ikasle berri baten izena eta nota emanda, zerrendako azkeneko posizioan txertatuko duena