



ADA-ko sarrera



ADA Sarrera

DoD-ek (USAko defentsa Dptua.) sustatua
Ezaugarriak:

- motekiko oso zorrotza
- Programazio inperatiboa (tradizionala)
- Objektuei zuzendutako programaziorako



ADA Sarrera

Programa baten oinarritzko egitura:

```
with Package_Name; use Package_Name;  
procedure Program_Name is  
  ---Specification  
    Variable : Some_Type;  
  begin  
    Statement_1;  
    Statement_2;  
  end Program_Name;
```



```
with Ada.Text_IO; use Ada.Text_IO;  
procedure Hello_World is
```

```
    ---espezifikazioa
```

```
    ---Sarrera:-
```

```
    ---Aurre: -
```

```
    ---Irteera: mezu bat
```

```
    ---Post: idatzi "Hello world!!!"
```

```
-- ez da aldagarik behar
```

```
begin
```

```
    Put ("Hello world!");
```

```
end Hello_World;
```



Espezifikazio eta beste oharrak

- Oharrak: konpiladoreak ez ditu interpretatuko
--- gidoietaz baliatuko gara oharrak idazteko
Adibidez

```
-- Totala kalkulatzeko formula  
Total := Partzial1 + Partzial2 +  
Partzial3;  
  
-- Ohar luze bat hainbat lerrotan  
-- banatu daiteke
```



Aldagaien deklarazioa

○ Adibidez

Deklarazioa soilik:

```
zenb, kontadore, i: Integer;  
erro_karratua: Float;  
kar, Letra: Character;  
Lerroa: String(1..80);
```

Deklarazioa eta hasieraketa:

```
Ordenatua: Boolean := false;  
Maximoa: Integer := 50;  
azkeneko_karakterea: Character :=  
'.';
```



Konstanteen deklarazioa

Konstanteen deklarazioa :

```
Entitatea: constant String(1..4) :=  
"ACME";
```

```
Pi: constant Float := 3.1416;
```

```
Bi_Pi: constant Float := 2.0 * Pi;
```



Oinarrizko datu motak Adaz

- **Integer** (zenbaki osokoak)
Domeinua [Integer'First, Integer'Last]
Eragiketak =,/=,>,>=,<,<=,+,-,*,/,rem,**
- **Float** (zenbaki errealak)
Domeinua [Float'First, Float'Last]
Eragiketak =,/=,>,>=,<,<=,+,-,*,/,** (float**integer)
- **Character** (karakterearak)
Eragiketak =,/=,>,>=,<,<=
- **Boolean** (boolearrak)
Eragiketa logikoak and, or, xor, not
- **String**
Eragiketak =,/=,>,>=,<,<=



Esleipena ADAz

- `:=`
- Moten arteko konpatibilitatea egotea garrantzitsua da
 - $\text{Mota (aldagaia)} = \text{Mota (espresioa)}$

bataz-bestekoa: float;

notak, ikasle_kop: integer;

bataz-bestekoa :=notak/ikasle_kop; **ERRORE!!!**

bataz-bestekoa :=float(notak)/float(ikasle_kop);



Oinarrizko S/I

- get eta put
get datu mota guztiekin erabil daiteke.
- **Nahiz eta Get eta Put desberdin bat egon datu mota bakoitzeko.**

Ada gai da jakiteko ze get edo put erabili nahi dugun irakurri edo idatzi nahi dugun datuen moten arabera.

Honi deitzen diogu eragileen sobrekarga .

Erabiltzeko erreza!!! Mota bost axola!!!



Ingurune klausulak

Beste fitxategi batean (liburutegietan) dauden eragiketak eta datu motak erabiltzeko aukera ahalbideratzen dute

Beharrezkoa da hurrengo liburutegiak gehitzea sarrera/irteera eragiketak erabili ahal izateko:

```
with Ada.Text_IO; use Ada.Text_IO;  
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;  
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
```



Baldintza eragiketak

```
If baldintza then  
    eragiketa(k);  
end if;
```

```
If baldintza then  
    eragiketa(k);  
elsif baldintza  
    eragiketa(k);  
elsif baldintza  
    eragiketa(k);  
end if;
```

```
If baldintza then  
    eragiketa(k);  
else  
    eragiketa(k);  
end if;
```



Iteraziozko eragiketak (Errepikatu-atera)

Loop

eragiketa(k);

exit when *baldintza*;
End loop;

Loop exit when baldintza;

eragiketa(k);

End loop;



Iteraziozko eragiketak (Aldagaiaren balio bakoitzeko)

```
n1 eta n2 arteko bakoitzeko  
errepikatu  
    eragiketa(k);  
amaiera
```

```
for I in n1 .. n2  
loop  
    eragiketa(k);  
end loop
```

```
for i in 10..1 loop  
    Put(i);  
    New_Line;  
end loop;
```



Iteraziozko eragiketak (Bitartean)

```
while baldintza loop  
    eragiketa(k);  
end loop;
```



Algoritmoa

```
znb1,znb2,akum,kont:integer;
irakurri(znb1);
irakurri(znb2);
((znb1=0) eta (znb2=0))egia
balitz
    idatzi ("balio zehaztu
            gabea");
bestela
    akum←1;
    kont ←0;
    Errepikatu-atera (kont = znb2)
                        egia balitz;
    akum ← akum*znb1;
    kont ← kont+1;
    errepikatu-amaitu;
    idatzi(akum);
amaiera
```

Programa

```
With Ada.Text_IO,Ada.Integer_Text_IO;
Use Ada.Text_IO,Ada.Integer_Text_IO;
procedure Berreketa_kalkulatu is
  znb1,znb2,akum,kont:integer;--deklarazioa
begin
    get(znb1);--datuen irakurketa
    get(znb2);
    if znb1=0 and znb2=0 then --filtro KASU KRITIKOA
      put("balio zehaztu gabea");
    else --kasu OROKORRAK
      akum:=1;
      kont:=0;
      loop exit when kont=znb2;
        akum:=akum*znb1;
        kont:=kont+1;
      end loop;
      Put("emaitza = ");
      Put(akum);
    end if;
end Berreketa_kalkulatu ;
```

