

# Estrutura de Linguagens 2016.2



Ada

Apresentada por  
Deborah Vancelotte

## Origem

Criada na década de 70 pelo Departamento de Defesa dos Estados Unidos.

Tem origem de Pascal, Simula, Fortran 77 e outras linguagens.

Utilizada em sistemas de controle de tráfego aéreo, aviônicos, dispositivos médicos e etc



Foi primeira linguagem de programação orientada ao objeto padronizada internacionalmente.

Nome Ada é uma homenagem a condessa Ada Lovelace.

## Classificação



Programação Estruturada

Tipagem Estática

Linguagem Imperativa

Orientada a Objeto

Alto Nível

# Ada 95

```
--Declaracoes de bibliotecas  
Procedure nome_do_programa is  
  --Declaracoes de variaveis  
begin  
  --Corpo do programa com as instrucoes e calculos  
end nome_do_programa;
```

## Python vs Ada

- Obrigatoriedade do ponto e vírgula
- Declaração do início/fim do programa principal
- Indicação de onde os comandos termina

- **Writability**

## Ada

```
with Ada.Text_IO; use Ada.Text_IO;

procedure Array_Length is
  NomeComposto : constant array (Positive range <>) of access constant String
    := (new String'("Deborah"),
        new String'("Fernanda"));

begin
  Ada.Text_IO.Put_Line (" Array Size : " & Integer'Image (NomeComposto'Length));
end Array_Length;
```

## Python

```
print('Array size:' + str(len(['Deborah', 'Fernada'])))
```

## Python vs Ada

- Garantia que em todo o programa a variável A será somente números inteiros.
- Facilidade de compreender outras operações envolvendo a variável A.

+ Legibilidade

## Ada

```
With Ada.Text_IO; Use Ada.Text_IO;  
With Ada.Integer_Text_IO; Use Ada.Integer_Text_IO;  
Procedure Duplicar is  
    A,B: Integer;  
  
begin  
  
    A := 5;  
    B := A * 2;  
    Put_Line("Duplicado é");  
    Put(B);  
  
end Duplicar;
```

## Python

```
a = 5  
b = a * 2  
print("Duplicado é \n" + str(b))
```

## Python vs Ada

- O Ada fornece um extenso conjunto de recursos para criar programas com módulos de código simultâneos.
- Um módulo de código concorrente Ada é chamado uma task.
- Em Python não é possível fazer essa operação com thread de forma paralela pois existe a limitação do Global Interpreter Lock ou GIL.

**+ Expressividade**

```
task seefriend is
    entry handshake;
end seefriend;

task body seefriend is
begin
    delay 5.0;

    accept handshake do
        Put("Vamos marcar alguma coisa?");
    End handshake;

    delay 1.0;

end seefriend;

begin
    Reset (G);
    A := Random(G);

    for i in 1 .. A loop
        delay(1.0);
    end loop;

    seefriend.handshake;

end Duplicar;
```



Fim

<https://github.com/dvancelotte>