

# Programação Estruturada

## Conceitos Elementares

**Prof. Edson Alves**

Faculdade UnB Gama

2020

# Sumário

1. Programação Estruturada
2. Fortran
3. Variáveis

# Principais características



# Fortran

- ▶ Fortran (*IBM Mathematical FORMula TRANslation System*) é uma linguagem de programação desenvolvida na década de 1950
- ▶ Até os dias atuais é uma das principais (ou a principal) linguagem utilizada em programação científica
- ▶ O primeiro compilador foi desenvolvido na IBM, por uma equipe liderada por John W. Backus, nos anos de 1954 a 1957
- ▶ O ISO/IEC 1539-1:1997 contém o padrão Fortran 95, um dos mais populares da linguagem
- ▶ Fortran apresenta notável performance em computação numérica, o que levou a sua adoção em pesquisas científicas e aplicações computacionalmente intensivas, como meteorologia, física, engenharia, etc

# GFortran

- ▶ O projeto GNU Fortran (GFortran) consiste em um *front-end* de compilador e bibliotecas de *run-time* para o GCC que dêem suporte à linguagem Fortran
- ▶ Ele é totalmente compatível com o padrão Fortran 95 e inclui suporte legado ao formato Fortran 77
- ▶ Em distribuições Linux com suporte ao apt, ele pode ser instalado com o comando

```
$ sudo apt-get install gfortran
```

- ▶ Para testar a instalação, insira o seguinte comando no terminal:

```
$ f95 -v
```

# Hello World!

```
1 ! Implementação do Hello World em Fortran
2 program hello
3
4     write(*,*) 'Hello, World!'
5
6 end program hello
```

## Compilação, linkedição e execução

- ▶ Para compilar um código Fortran (extensões .f90) é preciso invocar o GFortran, utilizando a *flag* `-c`:

```
$ f95 -c hello.f90
```

- ▶ No processo de linkedição é preciso indicar, os código-objetos que comporão o executável e, opcionalmente, o nome deste executável (opção `-o`):

```
$ f95 hello.o -o hello
```

- ▶ É possível executar ambas etapas em um só comando:

```
$ f95 hello.f90 -o hello
```

- ▶ Para rodar o executável criado, basta usar os mesmos mecanismos disponíveis em Linux para invocar um programa como, por exemplo, indicar seu caminho:

```
$ ./hello
```

# Variáveis em Fortran

- ▶ Em Fortran, as variáveis simbolizam regiões de memória, as quais podem ser lidas ou escritas
- ▶ Cada variável é identificada por um **nome**, que deve iniciar com um caractere alfabético e conter apenas caracteres alfanuméricos ou o símbolo '\_'
- ▶ Em Fortran não há distinção entre caracteres maiúsculos e minúsculos
- ▶ Assim como nas linguagens imperativas, uma variável identifica tanto o endereço da região de memória quanto o valor armazenado
- ▶ Qual dos dois valores será utilizado depende do contexto (se é um *l-value* ou um *r-value*)



# Declaração de variáveis e tipos de dados

- ▶ Uma variável pode ser declarada em Fortran usando a seguinte sintaxe

```
tipo_do_dado :: nome_da_variavel [= valor_inicial]
```

- ▶ Os principais tipos de dados em Fortran são: **real**, **integer**, **complex** e **character**
- ▶ O valor inicial é opcional
- ▶ Strings podem ser declaradas indicando-se o número de caracteres que a compõe

```
character (len = N) :: s      ! string de N caracteres
```

- ▶ Para declarar **constantes**, isto é, variáveis com permissão para leitura apenas), é utilizada a palavra-chave **parameter**:

```
complex, parameter :: pi = 3.1415
```

- ▶ No caso de constantes, o valor inicial é mandatório
- ▶ A expressão **implicit none** determina que todas as variáveis devem ser declaradas antes de seu uso, e é boa prática sempre utilizá-la no início dos programas

## Exemplo de declaração e uso de variáveis em Fortran

```
1 ! Computa a área de um círculo de raio r
2 program area
3
4     implicit none
5
6     real, parameter :: pi = 3.141592      ! Declaração de constante
7     real :: A                             ! Declaração de variável real
8     integer :: r                          ! Declaração de variável inteira
9
10    r = 8                                  ! Define um valor para r por meio de atribuição
11
12    A = pi * r ** 2                        ! Área do círculo
13
14    write(*,*) 'Area = ', a
15
16 end program area
```

# Operadores aritméticos e funções intrínsecas

- ▶ Sendo uma linguagem voltada para computação científica, Fortran tem suporte para uma série de operadores aritméticos
- ▶ No caso das expressões com mais de um operador, o operador de menor precedência é computado antes do de menor precedência
- ▶ Além disso, há um bom número de funções *intrínsecas* da linguagem, disponíveis sem a necessidade de importar arquivos ou bibliotecas externas
- ▶ Boa parte destas funções são relacionadas às funções matemáticas e manipulação numérica

Operador	Precedência	Operação
**	1	Expoenciação
*	2	Multiplicação
/	2	Divisão
+	3	Adição
-	3	Subtração

## Funções intrínsecas úteis

Função	Retorno
<code>abs(a)</code>	Valor absoluto de a
<code>sin(w)</code>	Seno de w
<code>cos(w)</code>	Cosseno de w
<code>conjg(z)</code>	Conjugado complexo de z
<code>log10(x)</code>	Logaritmo em base 10 de x
<code>mod(r1, r2)</code>	Resto da divisão de r1 por r2

**Tabela:** r: real ou inteiro, z: complexo, w: real ou complexo, x: real, a: qualquer tipo.

# Referências

1. **PADMAN**, Rachael. [Computer Physics: Self-study guide 2 – Programming in Fortran 95](#), University of Cambridge, Department of Physics, 2007.
2. GNU Fortran. [Welcome to the home of GNU Fortran](#), acesso em 29/01/2020.
3. **SHALOM**, Elad. *A Review of Programming Paradigms Throughout the History – With a Suggestion Toward a Future Approach*, Amazon, 2019.
4. Wikipédia. [Fortran](#), acesso em 29/01/2020.