

Programação de Computadores

Primeiro contato

Sumário

Programa

- O que é um programa?
- Um conjunto de instruções encadeadas de modo lógico, convertido em uma linguagem que computadores podem interpretar
- O que é um conjunto de instruções encadeadas de modo lógico?

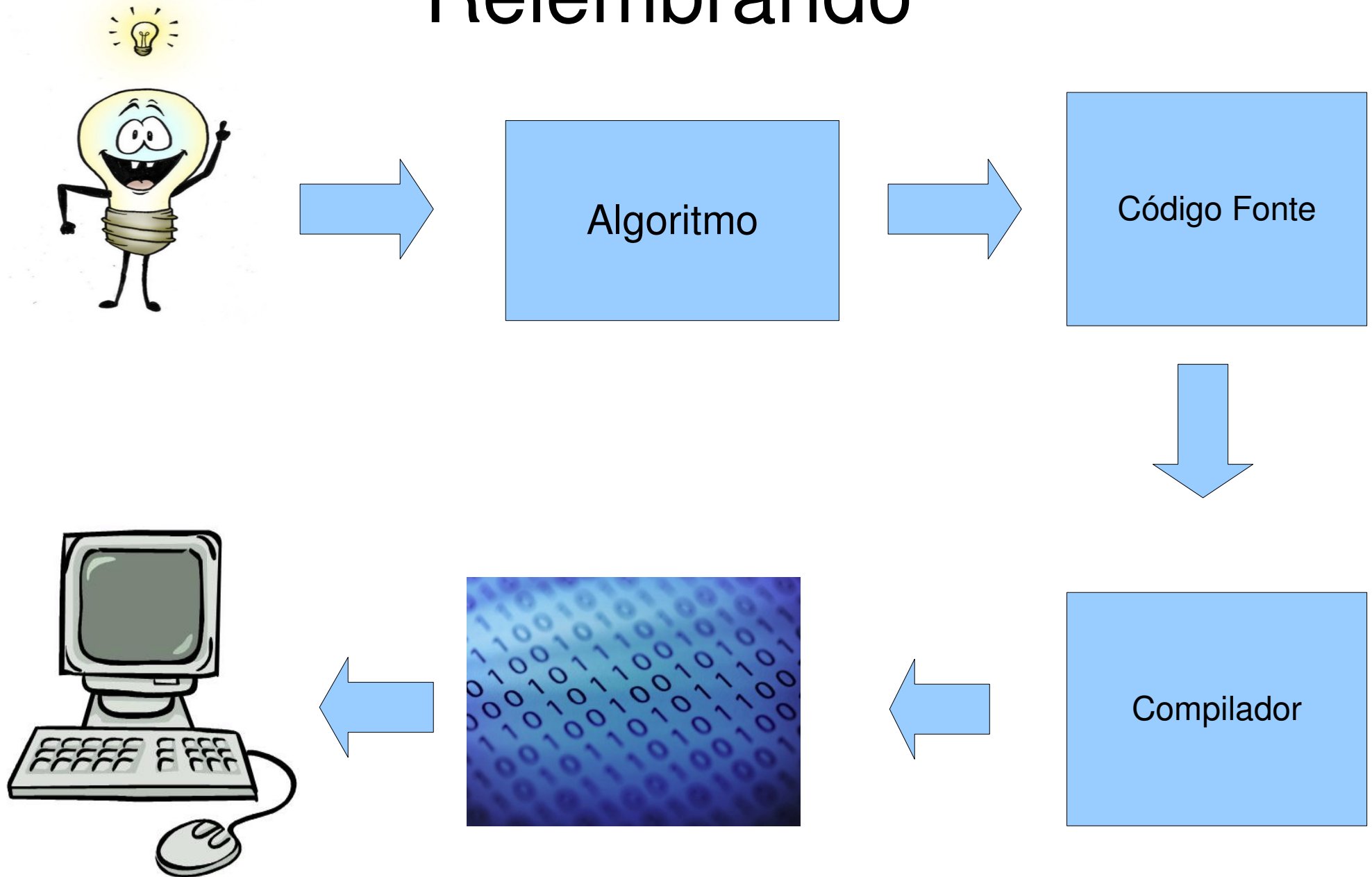
Algoritmo

- Um algoritmo é um conjunto de instruções (comandos/ordens) encadeados de modo lógico
- Como transformar algoritmos em instruções que o computador possa interpretar?

Compiladores

- Um compilador é responsável por transformar algoritmos, escritos em alguma linguagem, em instruções que o computador possa interpretar

Relembrando



Objetivo da Disciplina

- Estabelecer contato com uma linguagem de programação
- Construir programas nesta linguagem
- Ter contato com os problemas advindos de transformações de algoritmos para programas

Linguagem

- Linguagem Fortran 90
- Compilador gfortran (open-free)
- Ambiente - gedit

Observações

- Trabalhos e exercícios deverão ser entregues no padrão fortan 90
- Qualquer IDE pode ser usado para o desenvolvimento

Usando o compilador

Applications > Accessories > Terminal

Localizar a pasta correta

digite gfortran

Usando o Compilador

- `gfortran arquivo.f90`
- Foi gerado um arquivo chamado `a.out`
- Digite `./a.out`

Primeiro Programa

```
program Hello
```

```
    write(*,*) 'pedala robinho'
```

```
end program Hello
```

Refazendo

- Refaça o processo depois de salvar o arquivo
- Qual foi o resultado?

Análise do programa Hello

- Começa com “`program Hello`”
- Termina com “`end program Hello`”

Qualquer nome pode ser usado?

- Experimento 1: coloque nomes diferentes em program e end program
- Experimento 2: mude o nome do programa de Hello para 2Hello
- Experimento 3: mude o nome do programa para Hello2
- Experimento 4: mude o nome do programa para “Hello Fulano”

Regras para identificadores

- Sempre começam com letras
- Não contêm espaços
- Podem conter números
- Podem conter alguns caracteres especiais (mas não todos)
- No máximo 31 caracteres

Análise do Programa Hello

```
write(*, *)'pedala robinho'
```

- O que este comando faz?
- Sintaxe **write** (saida, formato) **expressão**

Comando write

- A presença de um formato e uma unidade são obrigatória
- Formatos de texto serão alvo de estudos futuros
- * no formato significa sem formato
- * na saída significa saída padrão (monitor)

Comando write

- A expressão pode ser uma constante, um texto ou ainda uma expressão matemática/lógica

Declaração de variáveis

- A declaração de variáveis em Fortran é feita usando a seguinte sintaxe:
- **Tipo** :: identificador1, identificador2
- tipo pertence à {integer, real, complex, logical e character}
- Obs.: o tipo character é declarado de maneira especial: **character** nome(tamanho)

Exemplo 2

```
program HelloFulano  
  character(50) :: nome  
  write(*,*) 'Olá '//nome  
end program HelloFulano
```

Comando de atribuição

- Coloca valores específicos em variáveis
- Variavel = valor

Exemplo 2

```
program HelloFulano  
  character(50) :: nome  
  nome = 'Fulano'  
  write(*, *) 'Olá '//nome  
end program HelloFulano
```

Exemplo 2

- Qual foi o resultado da execução?
- Por este resultado qual a função das //?

Operadores de caracteres

- // - concatenação
- len(v) – retorna o tamanho de um texto v
- trim(v) – remove os espaços de v
- nome(p1:p2) – texto de nome entre p1 e p2

Exercício

- Faça um programa que imprima um quadrado composto por asteriscos

```
*****  
*           *  
*           *  
*****
```

Operadores Matemáticos

- $+$ → adição
- $-$ → subtração
- $/$ → divisão
- $*$ → multiplicação
- $**$ → potenciação

Operadores lógicos

- .and. → Operador lógico “e”
- .or. → Operador lógico “ou”
- .not. → Operador lógico “negação”
- .eqv. → Operador lógico “equivalente”
- .neqv. → Operador lógico “não equivalente”

Operadores relacionais

- $>$ → maior que ($>$)
- \geq → maior ou igual que (\geq)
- $<$ → menor que ($<$)
- \leq → menor ou igual que (\leq)
- $==$ → igual a ($=$)
- \neq → diferente de (\neq)

Funções matemáticas pré-definidas

<i>Função</i>	<i>Significado</i>	<i>Tipo do argumento</i>	<i>Tipo de Retorno</i>
ABS(x)	módulo de x	INTEGER	INTEGER
		REAL	REAL
SQRT(x)	raiz quadrada de x	REAL	REAL
SIN(x)	seno de x em radianos	REAL	REAL
COS(x)	cosseno de x em radianos	REAL	REAL
TAN(x)	tangente de x em radianos	REAL	REAL

Funções pré-definidas

<i>Função</i>	<i>Significado</i>	<i>Tipo do argumento</i>	<i>Tipo de Retorno</i>
ASIN(x)	arcoseno de x	REAL	REAL
ACOS(x)	arcocosseno de x	REAL	REAL
ATAN(x)	arcotangente de x	REAL	REAL
EXP(x)	expoencial de x	REAL	REAL
LOG(x)	logaritmo natural de x	REAL	REAL

+ funções

<i>Função</i>	<i>Significado</i>	<i>Tipo do argumento</i>	<i>Tipo de Retorno</i>
INT(x)	parte inteira de x	REAL	INTEGER
NINT(x)	inteiro mais próximo de x	REAL	INTEGER
FLOOR(x)	piso de x	REAL	INTEGER
FRACTION(x)	parte fracionária de x	REAL	REAL
REAL(x)	converte x em REAL	INTEGER	REAL

Funções e + funções

<i>Função</i>	<i>Significado</i>	<i>Tipo do argumento</i>	<i>Tipo de Retorno</i>
MAX(x1, x2, ...,xn)	máximo de x1, x2, ... xn	INTEGER /REAL	INTEGER /REAL
MIN(x1, x2, ..., xn)	mínimo de x1, x2, ... xn	INTEGER /REAL	INTEGER /REAL
MOD(x,y)	resto da divisão de x por y	INTEGER /REAL	INTEGER /REAL

O comando Read

- `READ(unidade,formato) var1`
- Associa um valor informado pela unidade no formato especificado à variável `var1`
- Um `*` no lugar de unidade significa entrada padrão (teclado)
- Um `*` no formato significa nenhum formato

Exemplo 3

```
program HelloCiclano  
  character(50) :: nome  
  nome = 'Ciclano'  
  read(*,*) nome  
  write(*, *) 'Olá '//nome  
  write(*, *) 'letra 1=', nome(1:1)  
end program HelloFulano
```

Exercícios

- Faça um programa que leia dois números, calcule e imprima a adição entre eles
- Faça o mesmo para a subtração, multiplicação e divisão
- Faça um programa que leia dois números e imprima o resultado das quatro operações aritméticas entre eles

Exercício

- Faça um programa que calcule a multiplicação de três números
- Faça um programa que calcule a média aritmética de três números
- Faça um programa que leia três valores lógicos e calcule o resultado do “e” destes valores

Exercício

- Faça um programa que leia um número de cinco dígitos e imprima cada um dos dígitos separadamente.
- Ex: número lido: 45678. O programa deve imprimir separadamente os dígitos 4, 5, ...