

Algoritmos Computacionais: Exemplos de instruções em Scilab

Professor: Rodrigo Rocha
Curso: Engenharia Civil
Aula: 8

Pseudo-Linguagem

▶ Operadores Aritméticos

- São utilizados para realização de cálculos matemáticos

Operador	Operação	Exemplo de expressões
*	Multiplicação	$6 * 3 = 18$
+	Soma	$6 + 3 = 9$
-	Subtração	$6 - 3 = 3$
\ ou div	Divisão Inteira	$6.3 \setminus 3 = 2$ ou $6.3 \text{ div } 3 = 2$
/	Divisão real	$6.3 / 3 = 2.1$
mod ou %	Resto da divisão inteira	$5 \text{ mod } 2 = 1$ ou $5 \% 2 = 1$
^ ou exp	Potenciação	$6^2 = 36$ ou $\text{exp}(6,2) = 36$

Scilab

▶ Operadores Aritméticos

- São utilizados para realização de cálculos matemáticos básicos

Operador	Operação	Exemplo de expressões
*	Multiplicação	$6 * 3 = 18$
+	Soma	$6 + 3 = 9$
-	Subtração	$6 - 3 = 3$
/	Divisão Inteira	$6.3 / 3 = 2$
modulo	Resto da divisão inteira	<code>modulo(5,2) == 1</code>
^ ou **	Potenciação	$6^2 = 36$ ou $6^{**}2 = 36$

Expressões

- ▶ Expressões são combinações de variáveis, constantes e operadores
- ▶ Quando montamos expressões temos que levar em consideração a ordem com que os operadores são executados, conforme a tabela de precedência da linguagem.

Pseudo-Linguagem

▶ Operadores Relacionais

- São expressões que o resultado é um valor lógico.

Operador	Operador relacional	Exemplo de Expressão
>	Maior	$6 > 5 = \text{verdadeiro}$
<	Menor	$6 < 5 = \text{falso}$
<=	Menor ou igual	$6 >= 6 = \text{verdadeiro}$
>=	Maior ou igual	$6 <= 6 = \text{verdadeiro}$
=	Igual	$6 = 5 = \text{falso}$
<>	Diferente	$6 <> 5 = \text{verdadeiro}$



Scilab

▶ Operadores Relacionais

- São expressões que o resultado é um valor lógico.

Operador	Operador relacional	Exemplo de Expressão
>	Maior	$6 > 5 == \%t$
<	Menor	$6 < 5 == \%f$
<=	Menor ou igual	$6 >= 6 == \%t$
>=	Maior ou igual	$6 <= 6 == \%t$
==	Igual	$6 == 5 == \%f$
<> ou ~=	Diferente	$6 <> 5 == \%T$ ou $6 ~= 5 == \%T$



Pseudo-Linguagem

▶ Operadores Lógicos

- Os operadores lógicos atuam sobre expressões e também resultam em valores lógicos **VERDADEIRO** ou **FALSO**.

Operador	Operação	Descrição
E	Conjunção	Resulta VERDADEIRO se ambas as partes forem verdadeiras. EX: $6=3$ e $9<10$ = Falso
Ou	Disjunção	Resulta VERDADEIRO se uma das partes é verdadeira. EX: $6=3$ ou $9<10$ = Verdadeiro
Nao	Negação	Nega uma afirmação, invertendo o seu valor lógico: se for VERDADEIRO torna-se FALSO , se for FALSO torna-se VERDADEIRO . EX: $nao(6=3)$ = verdadeiro



Scilab

▶ Operadores Lógicos

- Os operadores lógicos atuam sobre expressões e também resultam em valores lógicos **%T** e **%t** ou **%F** e **%f**

Operador	Operação	Descrição
&	Conjunção	Resulta %T se ambas as partes forem verdadeiras. EX: $6==3 \ \& \ 9<10 == \%F$
	Disjunção	Resulta %T se uma das partes é verdadeira. EX: $6==3 \ \ 9<10 = \%T$
~	Negação	Nega uma afirmação, invertendo o seu valor lógico: se for %T torna-se %F , se for %F torna-se %T . EX: $\sim(6==3) = \%T$



Pseudo-Linguagem

▶ Tabela-Verdade

A	B	A E B	A OU B	NÃO A	NÃO B
VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	FALSO
VERDADEIRO	FALSO	FALSO	VERDADEIRO	FALSO	VERDADEIRO
FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO	FALSO
FALSO	FALSO	FALSO	FALSO	VERDADEIRO	VERDADEIRO

Scilab

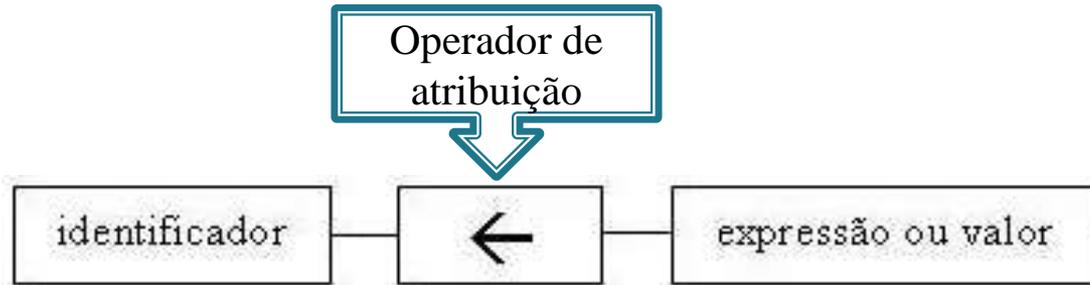
► Tabela-Verdade

A	B	A & B	A B	~A	~B
%T ou %t	%T ou %t	%T ou %t	%T ou %t	%F ou %f	%F ou %f
%T ou %t	%F ou %f	%F ou %f	%T ou %t	%F ou %f	%T ou %t
%F ou %f	%T ou %t	%F ou %f	%T ou %t	%T ou %t	%F ou %f
%F ou %f	%F ou %f	%F ou %f	%F ou %f	%T ou %t	%T ou %t

Comandos Básicos

▶ Atribuição

- Operação para armazenar uma expressão ou um determinado valor em uma variável para manipular posteriormente;



Exemplos:

Peso <- 78.7

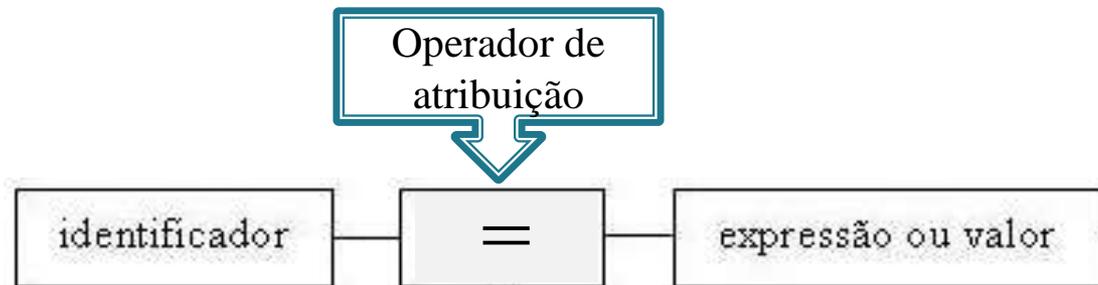
Nome <- "João da Silva"

Achei <- FALSO

Scilab – Comandos Básicos

▶ Atribuição

- Operação para armazenar uma expressão ou um determinado valor em uma variável para manipular posteriormente;



Exemplos:

Peso = 78.7

Nome = "João da Silva"

Achei = %F

Scilab – Comandos Básicos

▶ Comentários

- São declarações não compiladas que podem ser adicionadas ao código-fonte para referência e documentação de do programa.

▶ Operador: //

- Qualquer linha que se inicie por duas barras (//) não serão executadas.

▶ Exemplos:

- // Este método calcula o fatorial de $n \dots x <- y$;
- // Inicializa a variável x com o valor de y

Comandos Básicos

▶ Comandos de Entrada

- São usados para representar a entrada de dados com o uso dos dispositivos de entrada do computador
- Expressão: **Leia** (<variável>)

▶ Exemplo:

- **Leia** (nome)
- **Leia** (NumeroAgencia)
- **Leia** (NumeroConta)

Scilab – Comandos Básicos

▶ Comandos de Entrada

- São usados para representar a entrada de dados com o uso dos dispositivos de entrada do computador

Expressão: `<variável> = input(<texto de saída>)`

▶ Exemplo:

- `nome = input("Entre com o nome", "string")`
- `NumeroAgencia = input("Entre com o número da agencia")`
- `NumeroConta = input("Entre com o número da conta")`

Scilab – Comandos Básicos

▶ Comandos de Saída

- É utilizado para mostrar mensagens na tela para o usuário.
- ▶ `escreva(<frase de saída> , <lista de expressão>)`
- ▶ OBS: Para pular uma linha adiciona o “\n” dentro da “frase de saída”

▶ Exemplo:

- `escreval("Digite seu nome: ")`
- `escreva(" Digite sua agencia: ")`
- `escreval("Digite sua conta: ")`
- `escreval(" O seu nome é ", "Rodrigo")`



Monitor - Unidade de Saída

Digite seu nome:
Digite sua Agencia: Digite sua Conta:
O seu nome é Rodrigo

Scilab – Comandos Básicos

▶ Comandos de Saída

- É utilizado para mostrar mensagens na tela para o usuário.
- ▶ **printf**(<frase de saída> <formato de saída> , <lista de expressão>)
- ▶ OBS: Para pular uma linha adiciona o “\n” dentro da “frase de saída”

▶ Exemplo:

- **printf**("Digite seu nome: ")
- **printf**("\\n Digite sua agencia: ")
- **printf**("Digite sua conta: ")
- **printf**("\\n O seu nome é %s",'Rodrigo')



Monitor - Unidade de Saída

Digite seu nome:
Digite sua Agencia: Digite sua Conta:
O seu nome é Rodrigo

Scilab – Formato de saída dos valores

- ▶ <frase> é a sentença que se quer imprimir na tela, e que pode ser formulada junta por códigos de formato como %g.
- ▶ %g é um código de formato geral para expressões com valores numéricos.
- ▶ Existem vários outros códigos de formato como %d, %f ou %s,
- ▶ **EX:**
 - %f – Formato para valores fracionados
 - %i – Formato para valores inteiros
 - %s – Formato para textos

Comandos Básicos

- ▶ Saída com identificador / Variável

Algoritmo	Unidade de Saída
... A: Real A ← 19.4 Escreva(A) ...	19.4



Scilab – Comandos Básicos

- ▶ Saída com identificador / Variável

Algoritmo	Unidade de Saída
... A = 19.4 printf(“%f”,A) ...	19.4

Comandos Básicos

- ▶ Saída com expressão

Algoritmo	Unidade de Saída
... A,B: Real A ← 19.4 B ← 10.6 Escreva(A+B) ...	30



Scilab – Comandos Básicos

- ▶ Saída com expressão

Algoritmo	Unidade de Saída
... A = 19.4 B = 10.6 printf(“%g” , A+B) ...	30



Comandos Básicos

▶ Saída com texto

Algoritmo	Unidade de Saída
... Escreva (“Olá, ”) Escreva (“Boa Noite!”) ...	Olá, Boa Noite!

Scilab – Comandos Básicos

- ▶ Saída com texto

Algoritmo	Unidade de Saída
... printf(“%s” , ‘Olá’) printf(“%s” , ‘Boa Noite!’) ...	Olá, Boa Noite!

Comandos Básicos

- ▶ Saída com “texto” seguido de uma expressão

Algoritmo	Unidade de Saída
<pre>... A,B,C : inteiro A ← 7 B ← 9 Escreval(“Entre com um valor:”) Leia(C) //Valor fornecido pelo usuário como entrada Escreval(“A soma é: ”, (A + B) * C) ...</pre>	<pre>Entre com um valor: 10 A soma é: 160</pre>



Scilab – Comandos Básicos

- ▶ Saída com “texto” seguido de uma expressão

Algoritmo	Unidade de Saída
<pre>... A = 7 B = 9 C = input("Entre com um valor:") //O valor de C será fornecido pelo usuário como entrada printf("\n A soma é: %g", (A + B) * C) ...</pre>	<pre>Entre com um valor: 10 A soma é: 160</pre>



Comandos Básicos

- ▶ Saída com “texto” seguido de um identificador/variável

Algoritmo	Unidade de Saída
... A, B, RESPOSTA : inteiro A ← 7 B ← 9 RESPOSTA ← A + B Escreval(“A soma entre ”,A, “ e ”, B, “ é: ”, RESPOSTA) ...	A soma entre 7 e 9 é: 16

Scilab – Comandos Básicos

- ▶ Saída com “texto” seguido de um identificador/variável

Algoritmo	Unidade de Saída
<pre>... A = 7 B = 9 RESPOSTA = A + B printf(“A soma entre %i e %i é %i: ”, A, B, RESPOSTA) ...</pre>	A soma entre 7 e 9 é: 16



Resolvendo Problema com algoritmo

- ▶ **Problema:**
 - Calcular a média aritmética recebendo dois valores como entrada
- ▶ **Solução**

Algoritmo "Cálculo de Média Aritmética"

VAR

A,B,Media : REAL

Início

Escreval ("Programa que calcula a média aritmética de dois valores.")

Escreval ("Digite o primeiro valor: ")

Leia (A)

Escreval ("Digite o segundo valor: ")

Leia (B)

Media ← $(A+B) / 2$

Escreval ("A média dos dois valores é : ", Media)

FimAlgoritmo

Scilab – Resolvendo Problema com algoritmo

▶ **Problema:**

- Calcular a média aritmética recebendo dois valores como entrada

▶ **Solução**

```
printf ("Programa que calcula a média aritmética de dois valores. \n")
```

```
A = input("Digite o primeiro valor:")
```

```
B = input("Digite o segundo valor: ")
```

```
Media = (A+B) / 2
```

```
printf("A média dos dois valores é %g: ", Media)
```

Resolvendo Problema com algoritmo

▶ **Problema:**

- Lê o valor do raio e calcular a área do círculo correspondente.
- A área do raio é dada pela fórmula: $Pi * Raio^2$

▶ **Solução**

algoritmo "Calcula Área Circulo"

Var

Area, Raio: REAL

inicio

Escreval ("Entre com o raio: ")

Leia (Raio)

Area \leftarrow $Pi * Raio^2$

Escreva ("A área do circulo com o raio ", Raio, " é: ", Area)

fimalgoritmo

Resolvendo Problema com algoritmo

▶ **Problema:**

- Lê o valor do raio e calcular a área do círculo correspondente.
- A área do raio é dada pela fórmula: $Pi * Raio^2$

▶ **Solução**

```
// Algoritmo"Calcula Área Circulo"
```

```
Raio = input("Entre com o raio: ")
```

```
Area = %Pi * Raio^2
```

```
printf("A área do circulo com o raio %f é %f", Raio, Area)
```

Struct

- ▶ **Descrição** : Esta função retorna uma estrutura com campos de nomes `field1`, `field2`, ..., e valores correspondentes `value1`, `value2`, ...
- ▶ Criando um struct (estrutura)
 - Expressão: Variável = `struct`(`field1`,`value1`,`field2`,`value2`...)
- ▶ **Parâmetros**
 - `field1`, `field2`, .. strings, representam os nomes dos campos
 - `value1`, `value2`, .. representam os valores dos campos correspondentes

Utilizando struct

Exemplos

```
// criando um dado inicializando com valores  
struct date_st=struct('dia',25,'mes' ,'DEZ')
```

```
// criando uma struct sem inicializar  
struct date_st=struct()
```

```
//mudando o mes  
date_st.mes='AGO';
```

```
//mudando o dia  
date_st.dia=19;
```

```
// adicionando um novo campo  
date_st.semana=32
```