

Conceitos

Algoritmos I 2014/1

Profa: Daniela Scherer dos Santos

daniela.santos37@ulbra.edu.br

www.danielascherer.com.br

- ◆ Tipos primitivos de dados;
- ◆ Identificadores;
- ◆ Operadores:
 - ◆ aritméticos;
 - ◆ relacionais;
 - ◆ lógicos;
- ◆ Comandos:
 - ◆ atribuição;
 - ◆ entrada e saída de dados.

Tipos Primitivos



Por que os mágicos
tiram de suas cartolas
COELHOS e não
GIRAFAS?

Tipos Primitivos



Por que os mágicos
tiram de suas cartolas
COELHOS e não
GIRAFAS?

O coelho, além de mais elegante,
possui um tamanho e estrutura
corpórea que, além de caber dentro da
cartola, não torna o trabalho do mágico
tão árduo ao retirá-lo de lá

Tipos Primitivos



cartola => memória do computador
coelho => valor/dado que será
armazenado na memória para que o
algoritmo execute sua tarefa

Dependendo do valor/dado que se deseja
armazenar, uma quantidade maior ou
menor de memória deverá ser alocada
para guardá-lo.

Tipos Primitivos

Quando pensamos em DADOS que devem ser processados pelo algoritmo, devemos pensar na FORMA e no TAMANHO que eles poderão ter.

São NUMÉRICOS?
Serão apenas LETRAS?

Tipos Primitivos

Quando pensamos em DADOS que devem ser processados pelo algoritmo, devemos pensar na FORMA e no TAMANHO que eles poderão ter.

São NUMÉRICOS?
Serão apenas LETRAS?

657	“General Osório, 345”	Falso
123.78	“a”	Verdadeiro
“Márcia da Silva”	123	23.56

Tipos Primitivos

- Tipo inteiro:
 - toda informação numérica que pertença ao conjunto de números inteiros (negativos, nulo, positivos);
 - Exemplos: 5, 786, -458, 0, 370000
- Tipo real:
 - toda informação numérica que pertença ao conjunto dos números reais (negativos, nulos, positivos);
 - Exemplos: 768.85, 1.2, -3.67, 345643.987, 0.65

Tipos Primitivos

- Tipo caracter:
 - toda informação composta por um conjunto de caracteres alfanuméricos (0..9 / A..Z / a..z) e/ou caracteres especiais (#,/, \$, %, *, ?, ~, >, @, ...);
- Tipo Lógico:
 - toda informação que pode assumir duas situações – verdadeiro ou falso;

Identificadores

- Um IDENTIFICADOR é um nome/rótulo atribuído a uma porção de memória reservada para guardar o valor/dado que o algoritmo precisa armazenar;
- CONSTANTES: armazenam informações que não sofrem alterações com o decorrer do tempo;
- VARIÁVEIS: armazenam informações que têm a possibilidade de variar em algum instante no decorrer do tempo.

Identificadores

- Regras para a criação de identificadores:
 - devem ser formados por um ou mais caracteres, sendo que o primeiro deverá ser uma letra;
 - o caractere de sublinha (__) é considerado válido;
 - não é permitido o uso de outros caracteres especiais (por exemplo: %, *, +, ç);
 - não podem conter espaço;

VÁLIDOS

x
nome
a13
SAL_1
cidade
salLiquido

INVÁLIDOS

1xyz
nome aluno
sal*y
@email
cidade natal
dt nasc

Um identificador deve expressar em seu nome o que realmente irá manipular, não deixando dúvida ou ambiguidade.

Pelos padrões de projeto, as variáveis devem ser escritas em letras minúsculas. Caso exista a necessidade de expressá-la em 2 ou mais palavras, estas, a partir da 2^o, devem ter sua primeira letra escrita em maiúscula.

Exemplos: `dataDeNascimento`, `salarioBruto`

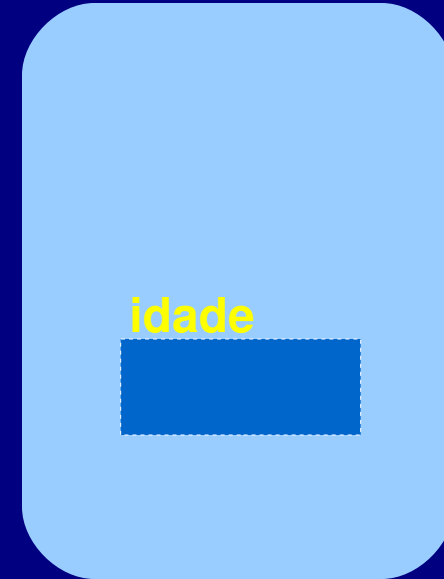
Declarando Variáveis

- Quando fazemos a definição de identificadores, chamamos esse processo de DECLARAÇÃO DE VARIÁVEIS;
- Declaramos variáveis para reservar uma quantidade de memória apropriada para armazenar um dado;

Declarando Variáveis

- Exemplos:
inteiro idade;

MEMÓRIA



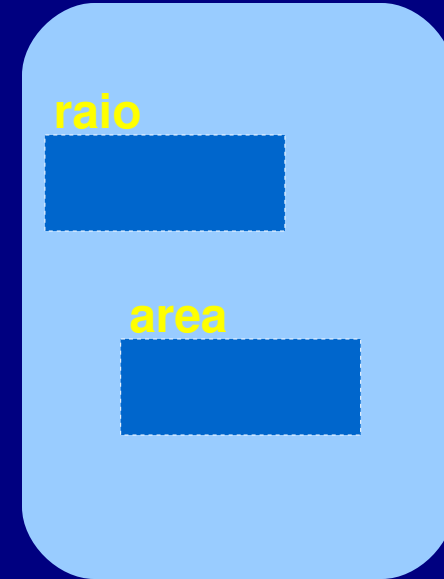
Obs.: nesse exemplo estamos reservando um espaço de memória, chamado *idade*, com tamanho suficiente para guardar um número do tipo inteiro e toda a vez que usarmos a palavra *idade* nas instruções do algoritmo, estaremos nos referindo ao valor numérico que está armazenado neste espaço de memória no computador.

Declarando Variáveis

- Exemplos:

real area, raio;

MEMÓRIA



Obs.: havendo mais de uma variável do mesmo tipo, elas podem ser declaradas na mesma linha, separadas por vírgula.

Declarando Variáveis

- Exemplos:

algoritmo somaValores

var

inteiro valor1, valor2, soma;

inicio

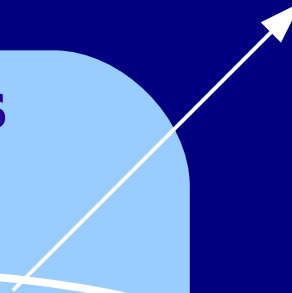
leia (valor1, valor2);

soma = valor1 + valor2;

escreva (soma);

fim

declaração das variáveis



Exercícios

1) Assinale os nomes de identificadores válidos:

- a) (X)
- b) KM/L
- c) P{0}
- d) U2
- e) UYT
- f) B52
- g) AH!
- h) Rua
- i) aluno
- j) cEp
- k) #cidade
- l) salarioLiquido
- 1) data de nascimento

Exercícios

2) Supondo que as variáveis **notaDoAluno**, **nomeDoAluno**, **numeroDaMatricula**, **sexo** sejam utilizadas para armazenar respectivamente a nota do aluno, o nome do aluno, o número da matrícula e o sexo, declare-as corretamente, ou seja, associe o tipo primitivo adequado ao dado que será armazenado.

3) Encontre os erros nas seguintes declarações de variáveis:

inteiro endereço, numeroFilhos;
caracter idade, X;
real peso, R\$;
lógico aprovado;

Expressões Aritméticas

- São expressões cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiro ou real);

- Exemplos:

$$a = x + y;$$

$$x = 4 * y;$$

$$y = 2 * (a + b);$$

Operadores Aritméticos

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão entre dois valores inteiros

—▶ $5 + 2$

—▶ $x - 7$

—▶ $85 * y$

—▶ $100 / x$

—▶ $5 \% 2 \rightarrow$ resultado 1 pois:

•Prioridades nas expressões aritméticas:

1) parênteses mais internos

2) – (menos unário)

3) funções matemáticas

4) * / %

5) + -

$$\begin{array}{r} 5 \overline{) 2} \\ \underline{4} \\ 1 \end{array}$$

resto da divisão de 5 por 2

Expressões Lógicas

- São expressões onde os operandos são relações e/ou variáveis do tipo lógico e os operadores são relacionais ou lógicos;
- Exemplos:
 $x > y$
 $(a == b) \text{ OU } (x == z)$

Operadores Relacionais

São utilizados para efetuar comparação entre valores do mesmo tipo, que podem ser constantes, variáveis ou expressões. O resultado da comparação será sempre lógico (verdadeiro ou falso).

==	Igualdade
>	Maior
<	Menor
!=	Diferente
<=	Menor ou Igual
>=	Maior ou Igual

Operadores Relacionais

Exemplos:

$a \leq b$

$x == y$

$10 \leq 5$ (esta expressão resultaria o valor lógico Falso, pois 10 não é menor ou igual a 5)

$8 > 4$ (esta expressão resultaria o valor lógico Verdadeiro, pois 8 é maior do que 4)

Operadores Lógicos

São utilizados para realizar operações sobre valores lógicos com a função de conectar condições a serem testadas durante a execução do algoritmo.

Operadores Lógicos

Operadores	Função	Exemplos
E	o operador E é utilizado quando necessitamos estabelecer que todas as condições de uma sentença devem ser verdadeiras	$(x > 5)$ E $(x < 10)$
OU	o operador OU é utilizado quando precisamos estabelecer que basta que uma das condições de uma sentença seja verdadeira	$(x == 10)$ OU $(x == 15)$
XOU	o operador XOU é utilizado quando necessitamos estabelecer que apenas uma das condições de uma sentença pode ser verdadeira	a XOU b
NÃO	o operador NÃO é utilizado quando é necessário estabelecer o valor contrário de uma condição	NÃO (a)

Operadores Lógicos

TABELA VERDADE DO OPERADOR E:

Valor 1	Operador	Valor 2	Resultado
Falso	E	Falso	Falso
Falso	E	Verdadeiro	Falso
Verdadeiro	E	Falso	Falso
Verdadeiro	E	Verdadeiro	Verdadeiro

Exemplos:

Se $(mediaAluno \geq 7)$ **E** $(presencaAluno \geq 20)$ Então
considere o aluno aprovado

Operadores Lógicos

TABELA VERDADE DO OPERADOR OU:

Valor 1	Operador	Valor 2	Resultado
Falso	OU	Falso	Falso
Falso	OU	Verdadeiro	Verdadeiro
Verdadeiro	OU	Falso	Verdadeiro
Verdadeiro	OU	Verdadeiro	Verdadeiro

Exemplos:

Se $(mediaAluno < 7)$ **OU** $(presencaAluno < 20)$ Então
considere o aluno reprovado

Operadores Lógicos

TABELA VERDADE DO OPERADOR XOU:

Valor 1	Operador	Valor 2	Resultado
Falso	xou	Falso	Falso
Falso	xou	Verdadeiro	Verdadeiro
Verdadeiro	xou	Falso	Verdadeiro
Verdadeiro	xou	Verdadeiro	Falso

Operadores Lógicos

TABELA VERDADE DO OPERADOR NÃO:

Operação <i>nao</i>	
A	nao A
Verdadeiro (V)	Falso (F)
Falso (F)	Verdadeiro (V)

Exemplos:

Se NÃO(mediaAluno \geq 7) Então
considere aluno reprovado

Operadores Lógicos

- Prioridade entre os operadores lógicos:
 - 1) NÃO
 - 2) E
 - 3) OU
 - 4) XOU

Prioridade entre Operadores

- Prioridade entre TODOS os operadores:
 - 1) Parênteses mais internos
 - 2) - (menos unário)
 - 3) Funções matemáticas
 - 4) Operadores aritméticos:
 - 1) * / %
 - 2) + -
 - 5) Operadores relacionais
 - 6) Operadores Lógicos:
 - 1) NÃO
 - 2) E
 - 3) OU
 - 4) XOU

Prioridade entre Operadores




Nos casos de operadores de mesma prioridade, será observada a ordem de execução da esquerda para a direita.

Comando de Atribuição

- Um comando de atribuição permite-nos fornecer um valor para uma variável;
- Para simbolizar a atribuição utiliza-se o sinal de **=** e lê-se *recebe*;

Sintaxe do comando de atribuição:

identificador = valor;



guardar um
objeto em
uma caixa

Comando de Atribuição

- Exemplos:

```
real nota1,nota2,media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;
```

Comando de Atribuição

- Exemplos:

```
real nota1,nota2,media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;
```

declara três variáveis (nota1, nota2 e media)



MEMÓRIA

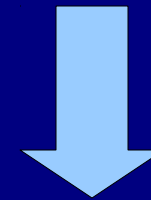


Comando de Atribuição

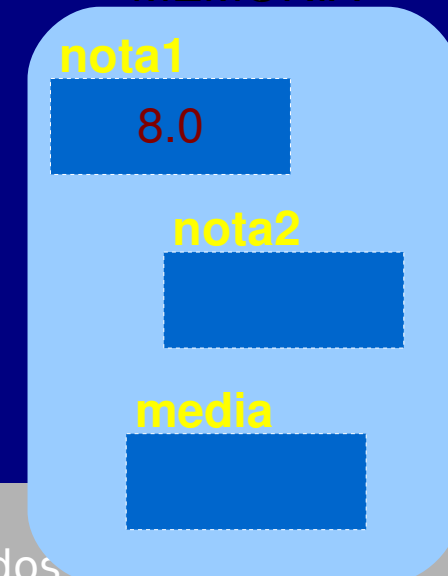
- Exemplos:

atribui o valor 8.0 para a variável *nota1*. Lê-se “*nota1* recebe 8.0”

```
real nota1, nota2, media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;
```



MEMÓRIA

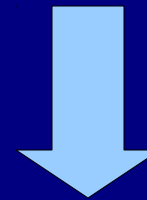


Comando de Atribuição

- Exemplos:

```
real nota1,nota2,media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;
```

atribui o valor 7.0 para a variável
nota2



MEMÓRIA

nota1

8.0

nota2

7.0

media

Comando de Atribuição

- Exemplos:

```
real nota1,nota2,media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;
```

atribui o resultado da expressão aritmética $(nota1 + nota2)/2$ para a variável *media*



MEMÓRIA

nota1

8.0

nota2

7.0

media

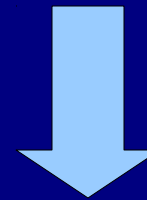
7.5

Comando de Atribuição

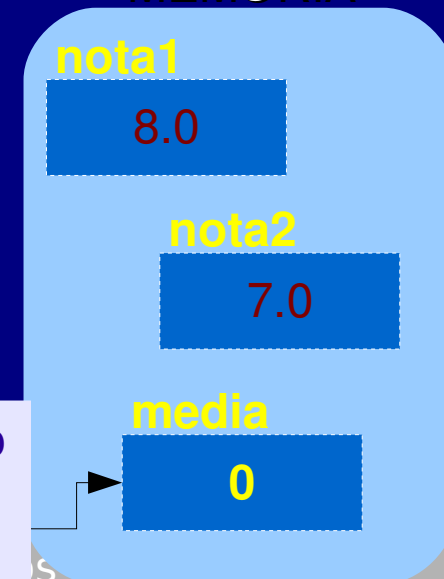
- Exemplos:

```
real nota1,nota2,media;  
nota1 = 8.0;  
nota2 = 7.0;  
media = (nota1 + nota2)/2;  
...  
media = 0;
```

atribui o valor 0 (zero) para a variável *media*



MEMÓRIA



o valor anterior (7.5) é substituído pelo novo valor 0. O valor 7.5 que havia sido anteriormente armazenado é “perdido”.

Comando de Entrada de Dados

- Os algoritmos precisam ser “alimentados” com dados para que possam efetuar as operações e cálculos que são necessários a fim de alcançar o resultado desejado;
- Para que o algoritmo possa receber os dados de que necessita adotaremos o comando de entrada de dados denominado *leia*;
- Sintaxe do comando *leia*:

leia (identificador);

onde, identificador é o nome da variável que armazenará o dado que está sendo lido

Comando de Entrada de Dados

- Exemplo:

```
real nota1, nota2, media;
```

```
leia (nota1);
```

```
leia (nota2);
```

```
media = (nota1 + nota2)/2;
```

Comando de Entrada de Dados

- Exemplo:
real nota1, nota2, media;
leia (nota1);
leia (nota2);
 $media = (nota1 + nota2)/2;$

a instrução `leia(nota1)` permite que seja fornecido (pelo usuário do algoritmo) um valor para a variável `nota1`

armazenará o
valor que foi
lido

MEMÓRIA

nota1

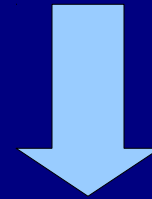
nota2

media

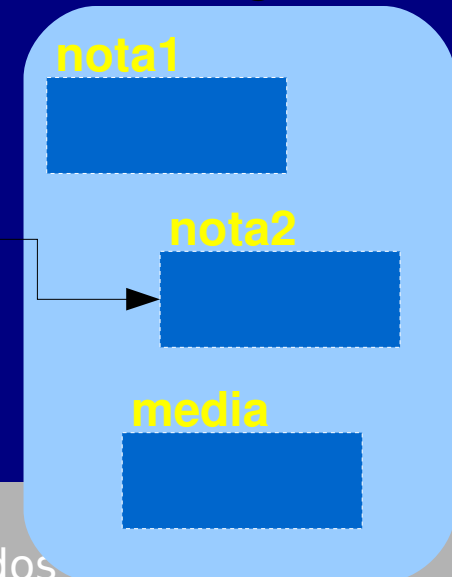
Comando de Entrada de Dados

- Exemplo:
real nota1, nota2, media;
leia (nota1);
leia (nota2);
media = (nota1 + nota2)/2;

a instrução leia(nota2) permite que seja fornecido (pelo usuário do algoritmo) um valor para a variável nota2



MEMÓRIA



armazenará o
valor que foi
lido para nota2

Comando de Entrada de Dados

- Exemplo:

```
real nota1, nota2, media;  
leia (nota1);  
leia (nota2);  
media = (nota1 + nota2)/2;
```

atribui o resultado da expressão aritmética $(nota1 + nota2)/2$ para a variável media

MEMÓRIA

nota1

nota2

media

receberá o resultado da operação
de acordo com os valores de nota1
e nota2

Comando de Saída de Dados

- Para que o algoritmo possa mostrar os dados que calculou, como resposta ao problema que resolveu, adotaremos o comando de saída *escreva*
- Sintaxe do comando *escreva*:

escreva (identificador/expressão);

onde *identificador* é a variável cujo conteúdo deseja-se escrever na tela e *expressão* poderá ser um texto informativo para o usuário.

Comando de Saída de Dados

- Exemplo:

```
real nota1, nota2, media;  
leia(nota1);  
leia(nota2);  
media = (nota1 + nota2)/2;  
escreva ("A média do aluno é: ", media);
```

Comando de Saída de Dados

- Exemplo:

```
real nota1, nota2, media;  
leia(nota1);  
leia(nota2);  
media = (nota1 + nota2)/2;  
escreva ("A média do aluno é: ", media);
```

supondo que o valor da variável nota1 seja 8.00 e o valor da variável nota2 seja 7.00. A instrução escreva do exemplo iria fornecer a seguinte saída:

A média do aluno é 7.5

Algoritmo completo

```
algoritmo calculaMedia;  
var  
    real nota1, nota2, media;  
inicio  
    leia(nota1);  
    leia(nota2);  
    media = (nota1 + nota2)/2;  
    escreva ("A média do aluno é: ", media);  
fim
```

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F.. Lógica de Programação: a construção de algoritmos e estruturas de dados. São Paulo: Makron Books, 2000.

JUNIOR, D. P. et.al. Algoritmos e Programação de Computadores. Rio de Janeiro: Elsevier Editora Ltda, 2012.