
Algoritmos I

Variáveis, Entrada e Saída de Dados

Carlos de Salles Soares Neto

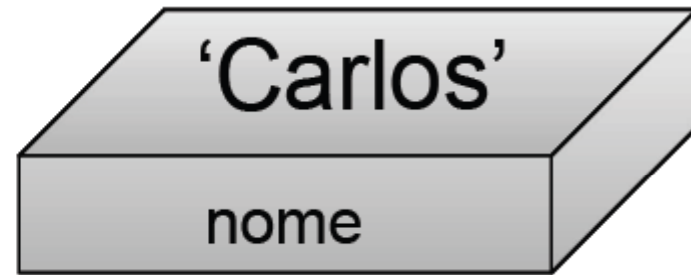
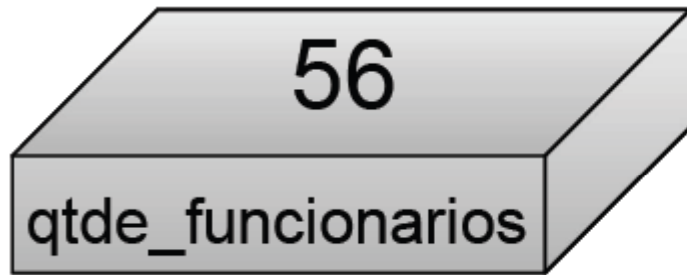
csalles@deinf.ufma.br

Segundas e Quartas, 17h40 às 19h10



Variáveis

- Área de memória para o armazenamento de dados



- Variáveis em algoritmos são sutilmente diferentes do mesmo conceito em matemática. O nome é dado porque aqui seu valor pode variar com o tempo

Nomenclatura de Variáveis

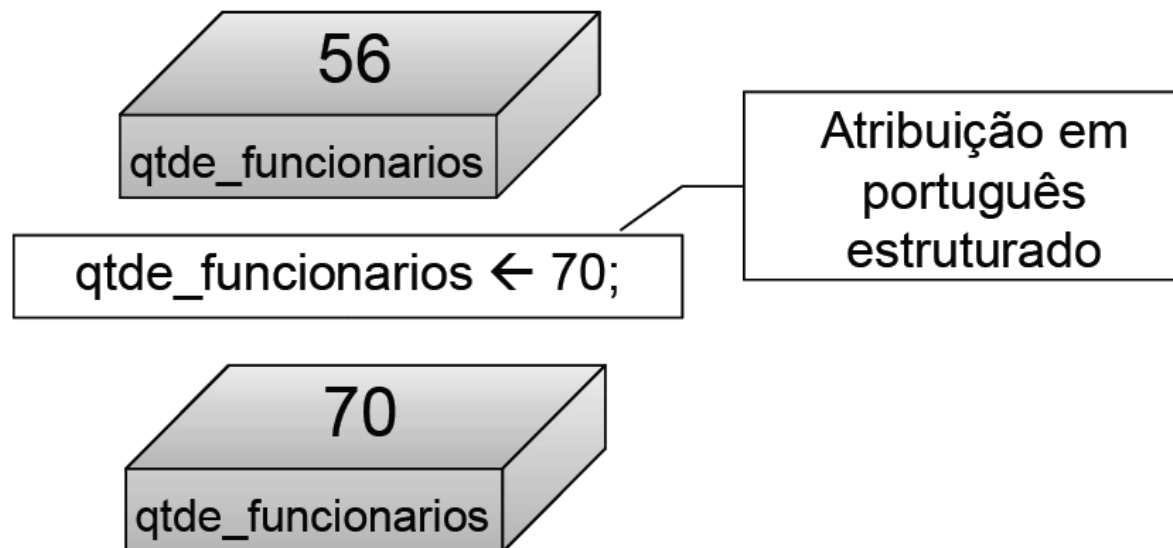
- Os nomes de variáveis devem começar com uma letra, seguida de outras letras, dígitos, ou sublinhado ("_");
- Toda variável possui um tipo associado;
- Os nomes de variáveis devem ser autoexplicativos;
- **Exemplos:**
 - qtde_funcionarios
 - idade
 - h, gr, qtde (nomes ruins)
- Historicamente, as letras i, j, k são usadas para contadores

Declaração de Variáveis

- A declaração de uma variável serve para identificá-la (dar um nome) e apresentar seu tipo;
- Sintaxe de uma declaração:
 <variável 1>, <variável 2>, ... :<tipo>
- **Atribuindo valores a variáveis**
- – Uma variável sempre armazena apenas um único
- valor;

Atribuindo valores a variáveis

- Uma variável sempre armazena apenas **um único** valor



- Numa atribuição, o novo valor da variável substitui o antigo

Atribuições

- Atribuir valores de tipos diferentes é um erro:
quantidade : **inteiro**;
quantidade ← 'Carlos';

Comando de Entrada de Dados

- Sintaxe do comando:
 - **ler** <variável1>, <variável2>, ...
- O comando **ler** é a forma mais rudimentar de entrada de dados por parte do usuário;
- A entrada de dados em português estruturado é feita exclusivamente por meio deste comando.

Entrada de Dados – Exemplos

```
clientes: inteiro;  
ler clientes;
```

```
nome_cliente: cadeia;  
idade_cliente: inteiro;  
ler nome_cliente, idade_cliente;
```

Saída de Dados

- Sintaxe do comando
 - **escrever** <variável1 ou expressão
ou mensagem>, ...
- O comando **escrever** é a forma mais rudimentar de saída de dados;
- Em português estruturado, a saída de dados é feita exclusivamente por meio deste comando.

Saída de Dados – Exemplos

```
escrever 'Olá. Boa tarde!'
```

Exibe na tela:

```
Olá. Boa tarde!
```

```
nome_cliente: cadeia;
```

```
idade_cliente: inteiro;
```

```
nome_cliente ← 'João';
```

```
idade_cliente ← 20;
```

```
escrever 'O cliente ', nome_cliente, ' tem ',  
idade_cliente, ' anos de idade.';
```

Exibe na tela:

```
O cliente João tem 20 anos de idade.
```

Sintaxe Geral de Português Estruturado

Tipos

```
<tiponovo1> = <tipo1>;
```

```
<tiponovo2> = <tipo2>;
```

Variáveis

```
<variável1>, <variável2>, ... : <tipo>;
```

```
<variável3>, <variável4>, ... : <tipo>;
```

```
//Isso é um comentário que será ignorado
```

Início

```
<comando1>; ...
```

Fim.

```
Módulo <nomemódulo1>;
```

```
[[ ...
```

```
]];
```

```
Módulo ...
```

Problema 1 – Equações de Segundo Grau

- Uma equação de segundo grau tem o seguinte formato:

$$ax^2 + bx + c = 0, a \neq 0 \wedge a, b, c \in R$$

- Com base nisso, escreva um programa que lê os valores a , b , c e exibe o resultado da equação de segundo grau.

Problema 1 - Solução

Variáveis

```
a, b, c, x1, x2 : real;
```

Início

```
escrever 'Informe o valor de a:';
```

```
ler a;
```

```
escrever 'Informe o valor de b:';
```

```
ler b;
```

```
escrever 'Informe o valor de c:';
```

```
ler c;
```

```
x1 ←  $(-b + \text{RAIZ}(b*b - 4*a*c)) / (2*a);$ 
```

```
x2 ←  $(-b - \text{RAIZ}(b*b - 4*a*c)) / (2*a);$ 
```

```
escrever 'X1 = ', x1;
```

```
escrever 'X2 = ', x2;
```

Fim.

Problema 2 – Área de um retângulo

- A área de um retângulo é definida por:

$$base * altura$$

- Faça um programa que receba do usuário a base e a altura de um retângulo e exiba a área desse retângulo.

Problema 2 - Solução

Variáveis

```
base, altura, area : real;
```

Início

```
escrever 'Informe a base do retângulo:';
```

```
ler base;
```

```
escrever 'Informe a altura do retângulo:';
```

```
ler altura;
```

```
area ← base*altura;
```

```
escrever 'Area = ', area;
```

Fim.

Problema 3 - Rendimentos

- No Brasil, uma aplicação financeira paga 20% de imposto de renda sobre os rendimentos brutos.
- Faça um programa que leia do usuário seu saldo em uma aplicação financeira e os rendimentos em um mês da mesma.
- Exiba como resposta o saldo atualizado da aplicação excluindo o imposto de renda.

Problema 3 – Solução

Variáveis

```
saldo_atual, saldo_novo, rendimentos : real;
```

Início

```
escrever 'Informe o saldo do fundo:';
```

```
ler saldo_atual;
```

```
escrever 'Informe os rendimentos:';
```

```
ler rendimentos;
```

```
saldo_novo ← saldo_atual *  
    (rendimentos*0.8);
```

```
escrever 'Novo saldo = ', saldo_novo;
```

Fim.

Problema 4 – Qual a senha?

- Para entrar em um sistema de segurança o espião precisa informar sua senha de acesso. A senha é 'abre-te sesamo'.
- Faça um programa que pede para o usuário informar a senha e exibe VERDADEIRO caso a senha seja correta ou FALSO em caso contrário.

Problema 4 – Solução

Variáveis

```
senha : cadeia;  
senha_correta : lógico;
```

Início

```
escrever 'Qual a senha?';  
ler senha;  
senha_correta  $\leftarrow$  senha='abre-te sesamo';  
escrever senha_correta;
```

Fim.

Problema 5 – Média em uma disciplina

- A média final de uma disciplina é calculada da seguinte forma:

$$\frac{nota1 + nota2 + nota3}{3}$$

- Faça um programa que leia as três notas de um aluno em uma disciplina e exiba sua média. Além disso, o programa exibe VERDADEIRO se o aluno tiver média sete ou mais e FALSO em caso contrário.

Problema 5 – Solução

Variáveis

```
notal, nota2, nota3, media : real;  
aprovado : lógico;
```

Início

```
escrever 'Informe a nota 1:';  
ler notal;  
escrever 'Informe a nota 2:';  
ler nota2;  
escrever 'Informe a nota 3:';  
ler nota3;  
media ← (notal+nota2+nota3)/3;  
aprovado ← media>=7;  
escrever 'Media = ', media;  
escrever 'Aprovado = ', aprovado;
```

Fim.

Problema 6 – *Pay per view*

- Em um sistema de TV por assinatura, alguns canais funcionam no sistema de *pay per view* e outros não.
- Os canais pay per view são múltiplos de 11 e os canais 56 a 62. Todos os demais canais são livres, ou seja, estão incluídos na assinatura mensal.
- Faça um programa que leia do usuário um número de canal e exiba na tela a mensagem '**Canal livre FALSO**' se o canal for pay per view ou '**Canal livre VERDADEIRO**' em caso contrário.

Problema 6 – Solução

Variáveis

```
canal : inteiro;  
pay_per_view, livre : lógico;
```

Início

```
escrever 'Informe o canal:';  
ler canal;  
pay_per_view ← (canal%11=0) OU  
               (canal>=56 E canal<=62);  
livre = NÃO pay_per_view;  
escrever 'Canal livre ', livre;
```

Fim.

Etapas da Construção de um Algoritmo

- Entender o problema
- Identificar as saídas
- Identificar as entradas do problema
- Analisar o processamento
- Manter separadas as etapas de entrada, processamento e saída
- Definir a sequencia de comandos
- Se necessário, realizar a verificação manual de algoritmos (chinês)

Modularização de algoritmos

- Equivale a dividir um problema maior em vários subproblemas, mais simples de serem resolvidos
- Método “dividir para conquistar”

Exemplo de modularização (Problema 1)

Variáveis

a, b, c, x1, x2 : real;

Início

ler_abc;

calcular_raizes;

escrever_raizes;

Fim.

Módulo ler_abc;

[[

escrever 'Informe o valor de a:';

ler a;

escrever 'Informe o valor de b:';

ler b;

escrever 'Informe o valor de c:';

ler c;

]];

Módulo calcular_raizes;

[[

x1 $\leftarrow$ $(-b + \text{RAIZ}(b*b - 4*a*c)) / (2*a)$;

x2 $\leftarrow$ $(-b - \text{RAIZ}(b*b - 4*a*c)) / (2*a)$;

]];

Módulo escrever_raizes;

[[

escrever 'X1 = ', x1;

escrever 'X2 = ', x2;

]];