

02

Variáveis

Fundamentos de Programação

Revisão

- Algoritmo → Uma série de ações em uma ordem específica
- Normalmente, os comandos em um programa são executados na ordem nos quais eles foram escritos. Isto é chamado de execução sequencial
- Ex: um algoritmo para calcular a área de um retângulo

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```


Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Memória

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Memória

```
altura =
largura =
area =
```

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Valor da altura:

Memória

altura =
largura =
area =

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Valor da altura: 5

Memória

altura = 5
largura =
area =

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Valor da altura: 5
Valor da largura:

Memória

altura = 5
largura =
area =

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

```
Valor da altura: 5
Valor da largura: 4
```

Memória

```
altura = 5
largura = 4
area =
```


Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

```
Valor da altura: 5
Valor da largura: 4
```

Memória

```
altura = 5
largura = 4
area = 20
```

Conceito de Variável

- Variáveis são posições nomeadas da memória usadas para guardar um valor que pode ser modificado pelo programa
- No algoritmo anterior, altura, largura e area são variáveis

```
ALGORITMO
  DECLARE altura, largura, area: REAL
  ESCREVA "Valor da altura: "
  LEIA altura
  ESCREVA "Valor da largura: "
  LEIA largura
  area ← largura x altura
  ESCREVA "A area do retângulo é ",area
FIM_ALGORITMO
```

Monitor

Valor da altura: 5
Valor da largura: 4
A área do retângulo é 20

Memória

altura = 5
largura = 4
area = 20

Conceito de Variável

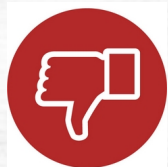
- Toda variável possui
 - Nome
 - Tipo
 - Valor

Nome ou Identificador

- Diversas linguagens de programação possuem regras diferentes quanto aos nomes das variáveis
- Na linguagem C
 - Os caracteres permitidos são: números, letras e o caractere sublinhado (_)
 - O primeiro caractere deve ser uma letra ou sublinhado
 - Não são permitidos espaços em branco e caracteres especiais (@, \$, +, %, ...)
 - Há diferença entre letras minúsculas e maiúsculas (Case Sensitive)



A
a
nota
NOTA
X5
A32
NOTA1
MATRICULA
nota_1



5b — *por começar com número;*
e 12 — *por conter espaço em branco;*
x-y — *por conter o caractere especial -;*
prova 2n — *por conter espaço em branco;*
nota(2) — *por conter os caracteres especiais ();*
case — *por ser palavra reservada;*

Nome ou Identificador

- Não é permitido usar as palavras reservadas, ou seja, palavras que pertençam à linguagem de programação

Tabela 1.2 Uma lista das palavras-chave de C ANSI.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Tipo

- Conjunto de valores e operações sobre esses valores
 - **Inteiros:** -2, -1, 0, 1, 2
 - **Reais:** 2.5, 3.0, -201.001
 - **Lógicos:** verdadeiro, falso
 - **Caractere:** 'a', 'G', '*', '#'
 - **Cadeia de caracteres (String):** "Ana Maria", "Bom dia", "Ola Mundo"

Tipo

- Na Linguagem C, tem-se 5 tipos básicos
 - **int** ↔ inteiros
 - **float** ↔ reais com 6 casas decimais de precisão
 - **double** ↔ reais com 10 casas decimais de precisão
 - **char** ↔ caracteres
 - **void**

Declaração de Variável

- Para usar uma variável, é necessário criá-la antes declarando o nome e o tipo da variável

```
int    x;  
float  n1, n2, soma;  
char   nome;
```

- Ao declarar uma variável, estamos reservando um espaço de memória para armazenar valores de um determinado tipo

Atribuição

- Comando que altera o valor de uma variável com o resultado de uma expressão

```
x ← 0
```

“Está sendo armazenado na memória reservada pelo nome ‘x’ o número inteiro zero”

- Na Linguagem C, o operador de atribuição é o símbolo igual (=)

```
x = 0;  
y = 2.5;  
z = 'm';
```

Entrada / Saída

- Praticamente todo programa precisa de interação com o usuário
- Entrada padrão será o teclado
 - Na linguagem C, o comando padrão de leitura é o comando `scanf()`

```
scanf("%d",&x );  
scanf("%f",&y );  
scanf("%c",&letra);  
scanf("%s",nome );
```

- **%d** ↔ inteiros
- **%f** ↔ reais
- **%c** ↔ caracteres
- **%s** ↔ strings

Entrada / Saída

- Praticamente todo programa precisa de interação com o usuário
- Saída padrão é o monitor, em formato de texto
 - Escreve-se na tela uma string e os valores das variáveis
 - Na linguagem C, o comando padrão de escrita é o comando `printf()`

```
printf("%d",x);  
printf("sexo: %c",sexo);  
printf("A soma de %f e %f eh igual a %f", n1, n2, soma);  
printf("Meu nome eh %s",nome);
```

- **%d** ↔ inteiros
- **%f** ↔ reais
- **%c** ↔ caracteres
- **%s** ↔ strings