Algoritmos e Programação

Tipos de dados

Prof. Dr. Joaquim V. C. Assunção

CENTRO DE TECNOLOGIA
UFSM
2025

Definições de termos

- Dado ≠ Informação
- Exemplo: ao falarmos 1960, estamos apenas colocando um número; ao dizermos que esta é a data de criação da UFSM estamos agregando valor ao dado.

Computador manipula dados

- O computador manipula dados não, necessariamente, informações.
- Os dados possuem diferentes tipos, os tipos mais básicos chamamos de primitivos.
- Podemos definir os seguintes tipos primitivos → →

Tipos primitivos

`int`

- Inteiro
  - Qualquer número sem precisão decimal.

Exemplos:

- Tirei **8** na prova.
- Meu prédio tem **4** andares.
- Meu personagem é nível **9**.
- A temperatura chegou a **-3**.
- Colar é trapaça e trapacear não é permitido, trapaça ← 0.
  Como bom jogador vou jogar nas regras, logo nunca vou tirar **0**.

`short`
`long`

Tipos primitivos

`float`

- Real (comumente conhecido como *float ou double*)
  - Qualquer número com precisão decimal

`double`
`long double`

Exemplos:

- Achei um bom X frango, e custa só R$ **10,50**.
- Tenho **1,73** de altura.
- Final do mês, fiquei com **-10,50** no banco.
- Vou tirar, no mínimo, um **8,2** em algoritmos.

Tipos primitivos

`char`

- Caractere
  - Um único dado que não é tratado como inteiro, por extensão temos o **literal**, que é um conjunto de caracteres.

Exemplos:

- Tirei um '**A**'.
- Sou o último da chamada meu nome começa com '**Z**'.
- Todo e-mail tem um caractere '**@**'.

`string`

- Um literal é um "**conjunto de caracteres**".
- Usamos "aspas" para denominar algo que é tratado literalmente, portanto, literal.

Tipos primitivos

`bool`

- Lógico
  - Qualquer dado que assuma a face de verdadeiro ou falso.

Exemplos:

- Se a==1, então a != 2 // Verdadeiro
- Se a saída é 1 então é verdade // Verdadeiro
- A==B e B==C, logo C != A // Falso

Usamos: V, F, 1, 0 ... *T, F, 1, 0*

Constantes

- Um dado é constante quando não sofre alterações ao longo do programa.

Ex:

- 5
- “Não muda”
- 2018
- 1,1415
- “J”

Variáveis

- Um dado que se pode ser alterado no tempo.

Ex:

- `dia ← 3;`
- `saldo ← 300.25;`
- `estaChovendo ← FALSE;`
- `top_score_from ← "Bill";`

- `Dia ← 4;`

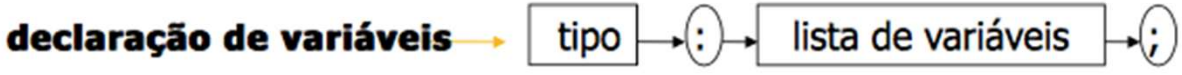
Formação de identificadores

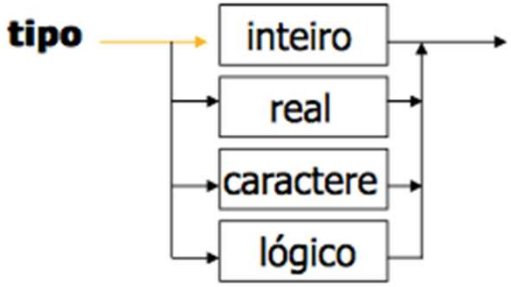
- Os dados (constantes ou variáveis) são representados em um algoritmo através de identificadores
  - Podem ser compreendidos como os nomes das informações
  - Devem começar por um caracter alfabético
  - Podem ser seguidos por mais caracteres alfabéticos ou numéricos
  - Não devem ser usados caracteres especiais

Declaração de variáveis

- Analogia do armário:
  - Se temos um objeto por gaveta. Por analogia...
  - Os objetos (que podem ser substituídos) → dados
  - Gavetas → variáveis

  - Portanto, precisamos diferenciar as gavetas através dos identificadores (etiquetas, rótulos) e também especificar o material (tipos primitivos) dos objetos que podem ser armazenados.





Declaração de variáveis

inteiro: i, j, w, x, y;

caractere: letra, vogal;

real: peso, dolar, pi;

lógico: chove

Declaração de variáveis

- Não podemos permitir que mais de uma variável possua o mesmo identificador
- Só podemos guardar dados (objetos) em variáveis (gavetas) do mesmo material (tipo primitivo)
  - Exemplo: uma variável do tipo primitivo inteiro só pode armazenar números inteiros

Exercícios

1. Dadas as variáveis: *peso, saldoBancario, idade, nome, cursandoInformatica, nota, tomada* (ligada ou não?), *gastoDeEnergia.* Instancie corretamente cada variável.
2. Encontre o erro nas seguintes declarações:

```
inteiro: sobrenome, nfilhos;

caracter: idade, nota;

real: peso, R$;

lógico: lâmpada, aprovado?;
```

Operadores aritméticos

Operador	Função	Exemplos
+	Adição	2 + 3, X + Y
-	Subtração	4 – 2, N – M
*	Multiplicação	5 * 5, A * B
/	Divisão	3 / 3, X / A

Outros operadores

Operador	Função	Exemplos
`pot(x,y)`	Potenciação x^y	`pot(3,2)`
`rad(x)` ou `raiz(x)`	Raiz quadrada	`raiz(9), raiz(a)`
`mod`	Resto da divisão	`9 mod 2` //1
`div`	Quociente (inteiro) da divisão	

Precedência de operadores

Prioridade	Operadores	
1	parênteses mais internos	
2	pot raiz	
3	* /	
4	+ -	

*Dica: Algumas linguagens de programação colocam *div* e *mod* na prioridade 2, outras na 3. Na dúvida use parênteses.

Exemplos

Exercícios

- Supondo que A, B, C são variáveis de tipo inteiro, com valores iguais a 5, 10, -8, respectivamente. Uma variável real D com valor de 1.5. Qual o valor das seguintes expressões?

1. (2 * A) mod 3 – C
2. raiz(-2 * C) div 4
3. pot(B,2)*3+3(1+2)*2
4. A mod 2 * 2 + 10 mod 3

Exercícios

- O que imprime o algoritmo abaixo?

```
literal a, b;
char c, d;
a <- "un";
c <- 'd'
//c <- 'i'
d <- 'o'
b <- "Olá M";
//O mesmo que escreva(b, a, c, d);
escreva(b + a + c + d);
```

- Os três algoritmos ao lado possuem erros de sintaxe.
- Escreva o algoritmo corretamente para que ele imprima o produto de **a** e **b** informados pelo usuário.

```
//Alg. A
int a, b;
leia(a); leia(b); leia(res);
res <- a*b;
escreva("res");
```

```
//Alg. B
int a, b;
leia(a); leia(b); leia(res);
res <- a.b
escreva("res");
```

```
//Alg. C
int a, b;
leia(a); leia(b);
leia(res <- a*b);
escreva("res");
```