

Design de Software Aplicado a Engenharias

Tipos de dados e expressões

Prof. Fabio Takeshi Matsunaga

Objetivo

Missão da aula

- Dados e informações.
- Apresentar os tipos de
- dados. Constantes e
- variáveis.

Expressões aritméticas e lógicas.

Variáveis

- Os programas de computadores manipulam dados e informações.
- Informações são os itens necessários para a existência de computadores.
- Existe uma diferença tênue entre dados e informações:
 - **Dados:** 21 de Setembro.
 - **Informação:** 21 de Setembro é o dia da árvore.

Tipos de dados

- Os tipos de dados primitivos são tipos de dados padrões a serem trabalhados pelos programas de computadores.
- Existem quatro tipos de dados primitivos que o computador é capaz de manipular:
 - **REAL.**
 - **INTEIRO.**
 - **CARACTER.**
 - **LÓGICO.**

Tipos de dados

INTEIRO

Tipo de dado numérico que pertence ao conjunto de números inteiros (positivo, negativo ou nulo).

- Fulano tem 5 irmãos.
- A escada possui 12 degraus.
- Meu vizinho comprou 2 carros novos. Ele tem 18 anos.
- Cornélio Procópio possui 48.615 habitantes.
- O curso de Engenharia de Software possui uma duração de 8 semestres letivos.

Tipos de dados

REAL

Tipo de dado numérico que pertence ao conjunto de números reais, formado pelos números inteiros e fracionários (positivos e negativos).

- Eu tenho 1,71 metros de altura.
- Meu saldo bancário é de R\$ 8540,55.
- Minha média final na matéria de Algoritmos foi de 9,5.
- O peso máximo para a categoria peso-pena é de 65,8 Kg.

Tipos de dados

CARACTER

Tipo de dado composto pelos caracteres alfanuméricos (alfabetos de A a Z e números de 0 a 9) e símbolos especiais (#, ?, !, @).

- A alternativa correta da questão 1 é a letra A.
- Pressione a tecla 1 para abrir o menu de ajuda.
- O símbolo é utilizado como identificador do domínio nos endereços de e-mail.

Tipos de dados

LÓGICO

**Tipo de dado que pode assumir apenas duas situações -
biestável, binário ou booleano**

- A porta pode estar aberta ou fechada.
- A lâmpada pode estar acesa ou
- apagada. O dia pode estar ensolarado ou
- chuvoso. O aluno pode ser aprovado ou
- reprovado.

O computador pode estar ligado ou desligado.

Tipos de dados

TEXTO

Não é um tipo de dado primitivo mas é uma informação que é composta por um conjunto de caracteres.

- Na prova consta a informação: "A avaliação é individual".
- No parque municipal estava uma placa dizendo: "Não pise na grama".
- O nome dele é "Fulano de Tal".
- O meu CPF é "1123.456.789-0".

Atividade em

Determine quais tipos de informações estão presentes nas sentenças a seguir :

- 1 A placa "Pare" está a 200 metros de distância.
- 2 Josefina subiu 5 degraus para pegar uma maçã
- 3 boa.
Alberta levou 3,5 horas para chegar ao hospital onde
- 4 foi atendida pelo Dr. José.
Astrogilda estampou a frase "Preserve o meio
- 5 ambiente" na sua camiseta e teve que pagar R\$100,59 ao pintor.
Felisberto recebeu sua medalha número 18 por ter alcançado a marca de 57,3 segundos nos 100 metros rasos.

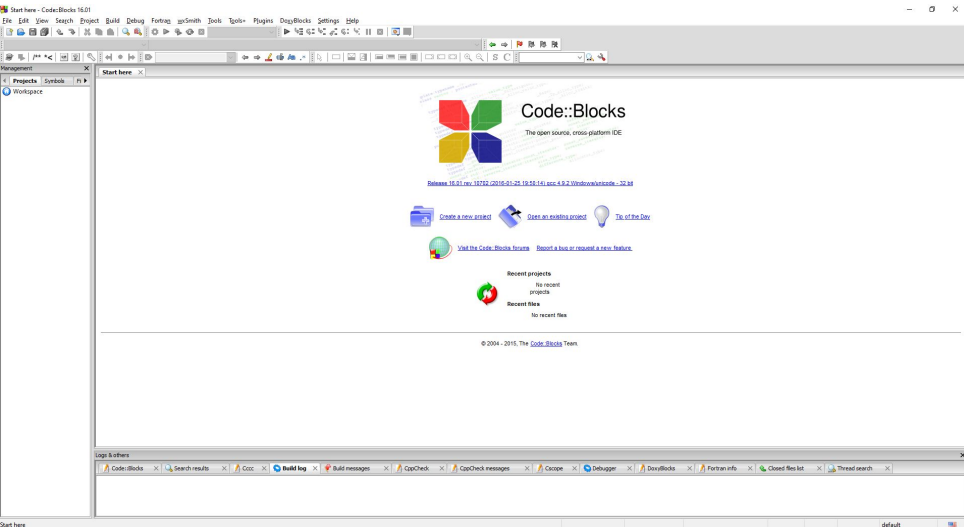
Dados

- Um tipo de dado é considerado constante se não sofre nenhuma variação no decorrer do tempo.
- Seu valor é o mesmo do início a fim da execução de um algoritmo.
- **Exemplos:** $\pi = 3,1416$, $e = 2,718$

Dados

- Um dado variável tem a possibilidade de ser alterado em algum instante no decorrer do tempo.
- Durante a execução do algoritmo, o valor do dado sofre alteração que dependa do momento ou circunstância.
- **Exemplos:** cotação do dólar, peso de uma pessoa, índice da inflação, idade.

CodeBlocks – software para desenvolvimento



Programa em C

```
programa-basico.c x
1  #include <stdio.h>
2
3  main() {
4
5
6
7  }
8
```

Declaração de bibliotecas

Bibliotecas são recursos e ferramentas necessárias para o desenvolvimento de programas.

Bloco de programa main()

Região do código onde serão definidas todas as instruções do programa, que devem estar entre as chaves { }.

Variáveis

- Os programas de computadores manipulam dados e informações.
- As informações são armazenadas na memória principal, que posteriormente serão executadas e manipuladas temporariamente.
- A memória pode ser vista como uma divisão de múltiplos espaços ou células, cada qual armazena **variáveis**.

Identificadore

- Durante o algoritmo é importante nomear os parâmetros que representam valores em mutação e que podem ser processados pelo algoritmo.
- Esses nomes são definidos por **identificadores**.
- Um identificador é um nome associado a um valor que pode ser acessado ou alterado pelo algoritmo.

Nome de

Regras para definição de identificadores.

- Deve começar com um caracter alfabético (a..z e A...Z)
- Deve ser formado por combinações de letras, dígitos e sublinhado.
- Não deve começar com números, conter acentuações e símbolos especiais.

Nome de

Sintaxe para definir um identificador

<tipo> identificador1, identificador2, identificador3;

- **tipo**: **int** (inteiro) ou **float** (real)
- **identificador**: nome da variável utilizando as devidas regras.

Variáveis

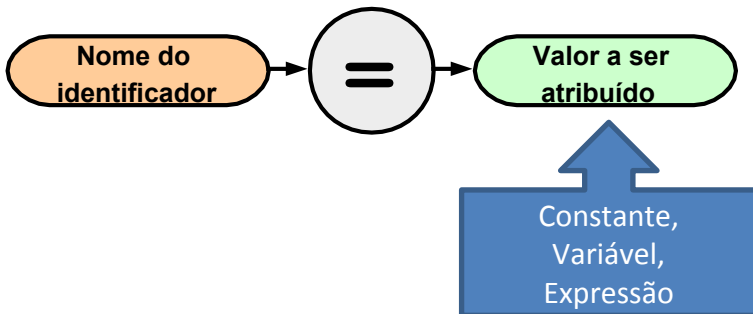
Exemplos de variáveis:

```
3  main() {  
4  
5      int x, y, z;  
6      float a, b, c;  
7  
8  }  
9
```

Comando de

- Uma atribuição permite fornecer um valor a uma variável, como se estivesse guardando um objeto na gaveta.
- O tipo do dado a ser fornecido deve ser compatível com o tipo declarado: **inteiro** recebe apenas números inteiros, **real** recebe apenas números reais.

Uma atribuição possui a seguinte sintaxe:



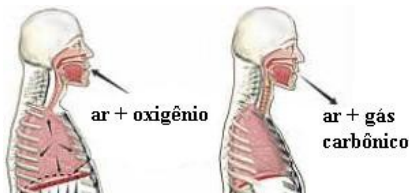
Variáveis

No exemplo abaixo, foram declaradas 2 variáveis x e y (inteiras).
Após a atribuição (linhas 6 e 7), x vale 10 e y vale 20.

```
1  #include <stdio.h>
2
3  main() {
4
5      int x, y;
6      x=10;
7      y=20;
8
9  }
10
```

Entrada e saída de dados

- Os algoritmos precisam ser alimentados com dados provenientes do meio externo, para que os cálculos sejam feitos a fim de alcançar o resultado desejado.
- Na nossa respiração, inspiramos gases da atmosfera (**entrada**) que serão processados pelo nosso organismo (**processamento**) que depois expiramos uma substância para o meio externo (**saída**).
- Em programas de computadores funciona da mesma forma com a diferença que no lugar de substâncias atmosféricas entram e saem **dados**.

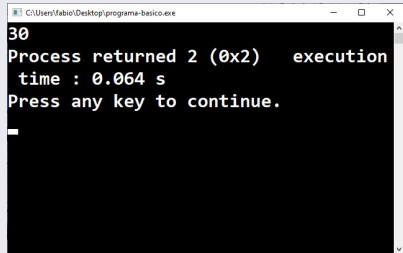


Comandos de entrada e

- **Entrada:** representa uma operação de leitura de dados do usuário. Uso do comando **scanf(x)**
- **Saída:** representa a saída de informações, através da exibição de dados na tela. Uso do comando **printf(x)**

Programa em C para somar 2 números

```
1  #include <stdio.h>
2
3  main() {
4
5      int x, y, s;
6      x=10;
7      y=20;
8      s=x+y;
9      printf("%i",s);
10
11  }
12
```



```
C:\Users\Fabio\Desktop\programa-basico.exe
30
Process returned 2 (0x2)   execution
time : 0.064 s
Press any key to continue.
```

Expressões

- Expressões aritméticas são cálculos e processamentos que são realizados entre dados numéricos.
- Para formar essas expressões, diversos operadores são utilizados:
 - **Aritiméticos** – unários e binários;
 - **Relacionais**;
 - **Lógicos**;

Operadores

- **Unários** – operam sobre um único operando
 - Menos unário ($-x$): utilizado para trocar o sinal algébrico de um valor (positivo para negativo ou vice-e-versa) ou para multiplicar um valor por -1 ;

Operadores

- **Binários** – operam sobre dois operandos
 - Soma ($x+y$): Soma dois valores;
 - Subtração ($x-y$): Subtrai dois valores (diferente do menos unário, apesar de usarem o mesmo símbolo); Multiplicação
 - ($x*y$): Produto de dois valores;
 - Divisão (x/y): Divisão de dois valores;
 - Módulo ($x\%y$): Resto da divisão de x com y ;

Operadores

- **Observação importante 1** : o operador de divisão / tem um efeito diferente quando aplicado em operandos inteiros e reais.
- **inteiro**: retorna a divisão inteira, sem o resto e as casas decimais. Exemplo: $5/2$ retorna 2.
- **real**: retorna a divisão com resultado exato, considerando as casas decimais. Exemplo: $5/2$ retorna 2.5.

Operadores

- **Observação importante 2** : o operador % somente se aplica a operandos inteiros.
- Exemplo: $5 \% 2 = 1$, pois o resto da divisão entre 5 e 2 é 1.

Precedência de

- Ao escrever fórmulas matemáticas e expressões aritméticas com múltiplos operadores nos algoritmos, é importante considerar a precedência de operadores, isto é, qual operação será executada primeiro.
- Esse conceito de precedência é a mesma utilizada que aprendemos na escola, em que a multiplicação e a divisão é realizada antes da soma e da subtração.
- Além destas mais simples e básicas, existe uma tabela de precedência completa que determina a ordem de execução dos operadores.

Tabela de

Precedência	Operadores
Maior precedência	()
	Funções
	*, /, %
	+, - (binários)
	<, >, <=, >= (relacionais)
	=, <> (igualdade)
Menor precedência	!, &&,

Exemplos de

Uma variável pode ser utilizada diversas vezes no algoritmo. Quando uma variável for usada novamente, será considerado seu último valor atribuído. Para testar o algoritmo, é importante fazer um **teste de mesa**.

Atribuição de variáveis

Algoritmo "exemplo1"

Var

A, B, C, D: **inteiro**

Inicio

A <- 10

B <- A / 2

C <- A - B

D <- A + B * C

Fimalgoritmo

Atividades sobre

Escreva os algoritmos para os seguintes problemas, considerando a sua entrada, processamento e saída.

- 1 Escreva um algoritmo para obter notas de três avaliações e em seguida calcular a média aritmética final.
- 2 Escreva um algoritmo para obter o preço de um produto e aplicar 15% de desconto sobre o preço do produto.
- 3 Escreva um algoritmo para obter o valor do raio de uma circunferência e calcular o valor da sua área.
- 4 Escreva um algoritmo para calcular a média ponderada obtida por um aluno para 2 notas bimestrais. A média ponderada é definido pela soma das notas multiplicada pelos seus pesos, que são 2 e 3 respectivamente, e dividido pela soma dos pesos.

Atividades sobre

Escreva os algoritmos para os seguintes problemas, considerando a sua entrada, processamento e saída.

- 1 Construa um algoritmo que, tendo como dados de entrada dois pontos quaisquer no plano, $P(x_1, y_1)$ e $P(x_2, y_2)$, escreva a distância entre eles. A fórmula que efetua tal cálculo é $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- 2 Desenvolva um algoritmo que leia um tempo em horas, minutos e segundos e em seguida converta o tempo em segundos.
- 3 Faça um algoritmo que leia o tempo de duração de um evento expressa em segundos e mostre-o expresso em horas, minutos e segundos. Por exemplo, se o valor da entrada for 7322 segundos, a saída deverá ser 2h 2m 2s.

Conclusão

- Nesta aula vimos que algoritmos manipulam **dados** de diversos
- **tipos** (inteiro, real, caracter e lógico).
- Os dados são identificados por **variáveis**.
Precisamos **declarar** as variáveis chamando-o por um **identificador**, que servirá de rótulo para referenciar a variável, junto com seu respectivo tipo.
- Com as variáveis declaradas é possível aplicar **operadores** aritméticos, relacionais e lógicos, que formam **expressões**.
As expressões podem ser atribuídas para outras variáveis, através de um **comando de atribuição**.
- Variáveis podem ser **lidas** pelo usuário e **escritas** na saída do programa.

OBRIGADO!

Contato

- **Fabio Takeshi Matsunaga**

- **E-mail:**

fabio.matsunaga@sistemafiep.org.br