

# Algoritmos de Programação 2

Professor: José Eurípedes Ferreira de Jesus Filho  
*jeferreirajf@gmail.com*

Universidade Federal de Jataí – UFJ

Vamos revisar alguns conceitos?

# Lógica de programação

- Linguagem natural:
  - ✓ Os pensamentos são intangíveis.
  - ✓ Utilizamos a fala para nos comunicarmos.
  - ✓ Utilizamos o raciocínio lógico para ordenar os pensamentos.
- Lógica:
  - ✓ Estuda os elementos das preposições, os tipos de preposições e de silogismos e as interações entre eles.
  - ✓ Busca técnicas de formalização, dedução e análises para validar argumentos.
- Lógica de programação:
  - ✓ Uso da lógica para desenvolver **algoritmos**.
  - ✓ Desejamos desenvolver **algoritmos bem definidos**.

# Variáveis e operadores

- Variáveis e constantes.
- Quatro tipos primitivos:
  - ✓ Caractere.
  - ✓ Inteiro.
  - ✓ Real.
  - ✓ Lógico.
- Identificadores.

# Variáveis e operadores

- Operadores aritméticos:
  - ✓ Soma: +
  - ✓ Subtração: -
  - ✓ Multiplicação: \*
  - ✓ Divisão: /
  - ✓ Resto da divisão: %
- Operador atribuição:
  - ✓ Atribuição: =
- Expressões aritméticas.

# Op. relacionais, lógicos e tabela verdade

- Operadores relacionais

- ✓ Utilizados para fazer comparações lógicas relativas entre expressões.
- ✓ As relações resultam em um valor lógico (verdadeira ou falsa).

- Operadores lógicos

- ✓ Utilizados para construir proposições compostas.
- ✓ Adição lógica, multiplicação lógica e negação.

- Tabela verdade

- ✓ Tabela matemática utilizada na álgebra booleana e no cálculo de proposições.
- ✓ Mapeia os resultados de uma expressão lógica a partir de cada entrada possível para as variáveis da expressão.
- ✓ Se  $n$  é o número de variáveis da expressão lógica a ser mapeada,  $2^n$  é a quantidade de linhas da tabela.

# SE/SENÃO

- Condicional SE

- ✓ Executa comandos baseado no resultado verdadeiro de uma condição.
- ✓ Os comandos só serão executados caso a condição seja verdadeira.
- ✓ Se a condição for falsa, o programa segue o curso normalmente.

- Condicional composta SE/SENÃO

- ✓ Analisa uma condição e executa determinados comandos caso ela seja verdadeira ou executa outros comandos caso ela seja falsa.
- ✓ Permite uma decisão também quando a condição é falsa.

# Estruturas de repetição

- Estrutura de repetição

- ✓ Permite que um bloco de instruções se repita enquanto uma determinada condição for verdadeira.
- ✓ Também chamado de **laço** ou **loop**.
- ✓ Se a condição **nunca se tornar falsa**, temos um **loop infinito**.

- Enquanto (While)

- ✓ A condição é verificada no início.

- Faça/Enquanto (Do/While)

- ✓ A condição é verificada no final.

- Para (For)

- ✓ A condição é verificada no início e está vinculada a uma contagem.



# Vetores e matrizes

- Vetores

- ✓ Conjunto de elementos homogêneos e consecutivos.
- ✓ Cada elemento é indexado por apenas um índice.
- ✓ Unidimensional.

- Matrizes

- ✓ Também é um conjunto de elementos homogêneos e consecutivos.
- ✓ Cada elemento é indexado por vários índices.
- ✓ Multidimensional.

# Procedimentos e funções

- **PROCEDIMENTOS**

- ✓ Subprograma sem retorno de valor.
- ✓ Usado para reduzir tamanhos de códigos, evitar repetições e facilitar reusabilidade.

- **FUNÇÕES**

- ✓ Subprograma com retorno de valor.
- ✓ Mesma utilização que os procedimentos.

- **PARÂMETROS**

- ✓ Podem ser passados por valor ou por referência.
- ✓ Passagem por valor faz uma cópia dos parâmetros.
- ✓ Passagem por referência passa o endereço dos parâmetros e permite alteração dos valores.

# Exercícios

- Construa a tabela verdade para a seguinte expressão  $\neg(A \cup (B \cap C))$ .

# Exercícios

- Resolva as expressões abaixo. Considere  $x=5$ ,  $y=2.5$  e  $A=\text{verdade}$ ,  $B=\text{falso}$  e  $C=\text{falso}$ :

1.  $((x+2.5)*x+y)/x-10 \leq y*x*10/3x-10.$

2.  $((x-y)*2 == x-y*2) \text{ OU } (A \text{ E } B).$

# Exercícios

- Desenvolva um procedimento em linguagem C que recebe um vetor  $v$  de tamanho  $n$  de inteiros e imprime na tela a soma de todos os elementos primos.

# Exercícios

- Desenvolva uma função em linguagem C que recebe duas **strings** e retorna a concatenação delas.

# Exercícios

- Desenvolva uma função em linguagem C que recebe um valor inteiro **n** e então calcula:

$$1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

# Exercícios

- Desenvolva o código de um programa em C onde o usuário informa a **quantidade de alunos** e a **quantidade de notas** de cada aluno. A partir daí, o usuário irá informar as **notas de cada aluno**. No final, o programa deve imprimir quantos alunos foram **aprovados** e quantos alunos foram **reprovados**. O critério de aprovação é a média das notas de um aluno ser maior ou igual a seis.