

Algoritmos e lógica de programação

Definição de finitos passos para resolver um problema.



Algoritmos

Um algoritmo pode ser definido como uma sequência finita de passos (instruções) para resolver um determinado problema. Sempre que desenvolvemos um algoritmo estamos estabelecendo um padrão de comportamento que deverá ser seguido (uma norma de execução de ações) para alcançar o resultado de um problema.

Para o desenvolvimento de um algoritmo eficiente é necessário obedecermos algumas premissas básicas no momento de sua construção:

- ◁ Definir ações simples e sem ambiguidade;
- ◁ Organizar as ações de forma ordenada
- ◁ Estabelecer as ações dentro de uma sequência finita de passos.



Algoritmos

Os algoritmos são capazes de realizar tarefas como:

- Ler e escrever dados
- Avaliar expressões **algébricas**, relacionais, lógicas
- Tomar decisões com base nos resultados das expressões avaliadas
- Repetir um conjunto de ações de acordo com uma condição





Formas de Representação de Algoritmos

Dentre as formas de representação de algoritmos mais conhecidas podemos citar:

Descrição Narrativa;

Fluxograma Convencional;

Pseudocódigo, também conhecido como Linguagem Estruturada.





Descrição Narrativa

1

3

2

4

Fluxograma



Início e final do fluxograma



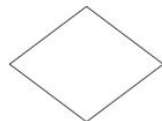
Operação de entrada de dados



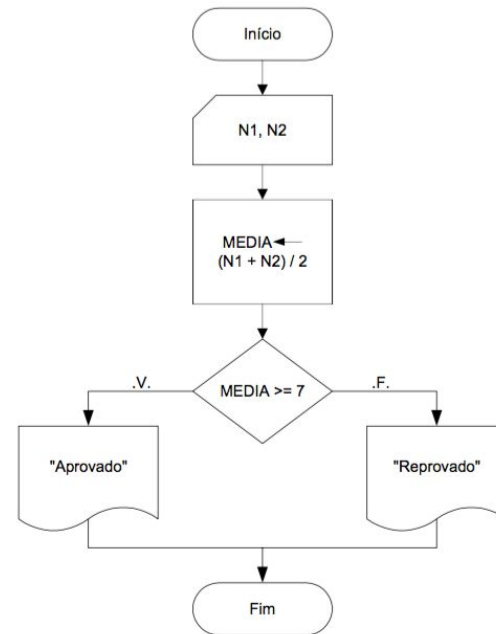
Operação de saída de dados



Operação de atribuição



Decisão





Pseudocódigo

Pseudocódigo

Algoritmo Calculo_Media

Var N1, N2, MEDIA: real

Início

Leia N1, N2

$MEDIA \leftarrow (N1 + N2) / 2$

Se $MEDIA \geq 7$ então

Escreva "Aprovado" Senão

Escreva "Reprovado" Fim_se

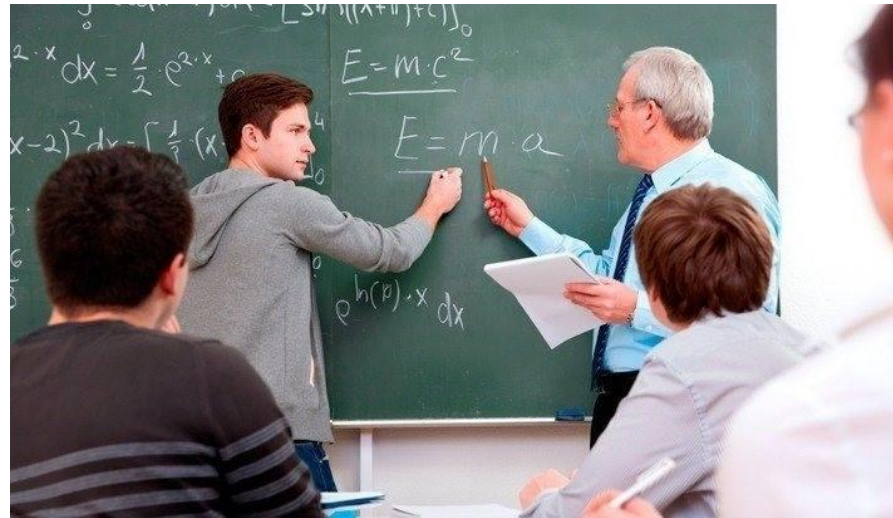
Fim



Exercício

Desenvolver um algoritmo para determinar se um aluno está aprovado, em recuperação ou reprovado.

O algoritmo vai receber 3 notas e a quantidade de faltas. Contando que a quantidade de aulas é 20, se a frequência do aluno for inferior a 75% ele está reprovado por falta, Se a frequência for superior a 75% e a média de suas notas for superior a 7 o programa deve informar que o aluno está aprovado



Lógica Proposicional

As proposições são determinadas por sentenças declarativas, pertencentes a uma certa linguagem, que formam um conjunto de palavras ou símbolos e expressam uma ideia. As sentenças declarativas são afirmações que podem receber apenas dois valores, Verdadeiro ou Falso. As proposições devem seguir os seguintes princípios:

1. **Princípio da identidade:** garante que uma proposição é igual a ela mesma.
2. **Princípio da não-contradição:** uma proposição não pode ser verdadeira e falsa.
3. **Princípio do terceiro excluído:** uma proposição é verdadeira ou falsa.



Lógica Proposicional

Qualquer sentença que não puder receber a atribuição de verdadeira ou falsa não é uma proposição. ***Sentenças interrogativas, exclamativas e imperativas não são proposições, pois não é possível dizer se são verdadeiras ou falsas.***

Exemplos de sentenças que não são proposições:

- Como foi a aula?
- O pior atentado nos EUA ocorreu em setembro de 2011?
- Limpe a cozinha.
- Que local de trabalho horroroso!
- Esta sentença não é verdadeira.



Proposições compostas

Proposição composta é a união de proposições simples por meio de um conector lógico. Este conector irá ser decisivo para o valor lógico da expressão.

Precedência de operadores

Em expressões que utilizam vários operadores não é possível saber qual proposição deve-se resolver primeiro.

Exemplo: $P \wedge Q \vee R$.

Com isso, usar parênteses é fundamental. A expressão do exemplo poderia ficar assim: $(P \wedge Q) \vee R$ ou $P \wedge (Q \vee R)$.

A ordem da precedência de operadores é:

1. $() , , \{ \}$
2. \neg
3. $\wedge, \vee, \underline{\vee}$
4. \rightarrow
5. \leftrightarrow



Conectivos lógicos

Proposições podem ser ligadas entre si por meio de conectivos lógicos. Conectores que criam novas sentenças mudando ou não seu valor lógico (Verdadeiro ou Falso). Exemplos dos principais conectores lógicos:

- “ \neg ” ou “ \sim ” (negação);
- “ \wedge ” (conectivo “e”);
- “ \vee ” (conectivo “ou”);
- “ \rightarrow ” (conectivo “se, então”);
- “ \leftrightarrow ” (conectivo “se, e somente se”);
- “ $\underline{\vee}$ ” (conectivo “ou exclusivo”);
- “ \downarrow ” (conectivo “negação conjunta”);
- “ \uparrow ” (conectivo “negação disjunta”).



Tabela Verdade

A tabela verdade é construída para determinar o valor lógico de uma proposição composta. Segue uma excelente estratégia para a construção desta.

Exemplo de construção da tabela verdade da proposição composta: $p \wedge q$

Primeiramente verifica-se quantas “variáveis”, ou proposições simples que temos na proposição composta do exercício. Neste caso existem duas: p e q .

Em seguida elevamos 2 ao número de variáveis, ou seja, 2^2 . Nossa base do expoente é 2 pelo fato de possuir-se apenas 2 valores lógicos possíveis nas proposições (Verdadeiro ou Falso). O resultado de 2^2 é 4. Então nossa tabela terá 4 linhas, nessas linhas estarão todos os valores lógicos possíveis da nossa proposição composta.



Tabela Verdade

Esta é a estrutura da tabela, agora para a preencher com os devidos valores lógicos utiliza-se a seguinte técnica: até a metade da primeira coluna coloca-se Verdadeiro, na outra metade Falso. Já na segunda coluna, intercala-se V e F. Desta forma adquire-se a seguinte tabela:

Esta é uma das melhores estratégias para a montagem de uma tabela verdade.

p	q	$p \wedge q$
V	V	Resultado
V	F	Resultado
F	V	Resultado
F	F	Resultado

