



MÓDULO III – PROGRAMAÇÃO

Aula 2

PIBIC-EM 2017

Alan Tavares – alan@fem.unicamp.br

Ementa do Módulo de Programação

- ▶ **Lógica de Programação – Aula 1**
 - ✓ Algoritmo / Definição de Problema / Análise Solução;
- ▶ **Estruturas de Controle (Laços de Repetição) – Aula 2/3**
 - ✓ Aplicação de programas: Sequencial / Condicional / Interação;
- ▶ **Vetores e Funções – Aula 4/5**
 - ✓ Criação / Aplicação / Exercícios Aplicados Robótica;
- ▶ **Introdução: Linguagem de Programação – Aula 6**
 - ✓ Conceitos



Lógica de Programação - **Exercícios**

1. Ler quatro valores inteiros, em seguida, efetue a média e mostre o resultado final.
2. Ler um preço de um produto em reais, e a taxa de conversão em dólar e informar o preço do produto convertido para dólar.
3. Realizar um simulador de poupança. Ler o valor do investimento inicial, ler o quanto você quer poupar mensalmente, ler taxa de rentabilidade anual e tempo em anos que você quer deixar poupando. Por fim, o programa deverá mostrar Total Valor poupado, Total Juros ganhos e o Quanto você terá de total no final.



Normas (convenções) Algoritmo

Algoritmo: Soma

Objetivo: Retornar a Soma de 2 (dois) valores

Entrada: Valor 1, Valor 2

Saída: Soma

Início

Ler : valor 1 = a;

Ler : valor 2 = b;

Efetuar Resultado: soma = a + b;

Escrever Resultado: ("Valor Soma: ", soma);

Fim



Normas (convenções) Algoritmo

Algoritmo: Converter para Dólar

Objetivo: Informar a conversão de um valor de Real para Dólar

Entrada: Preço em Real, Taxa de Conversão

Saída: Preço em Dólar

Início

Ler : preço em real / preal;

Ler : taxa de conversão / taxa;

Efetuar Conversão: preço de dólar / pdolar = preal*taxa;

Escrever Resultado: ("Preço em Dólar: ", pdolar);

Fim



Normas (convenções) Algoritmo

Algoritmo: Simulador de Poupança

Objetivo: Simular rendimento anual de taxa e valores pré determinados, por fim obter valor poupado e juros.

Entrada: Valor Inicial ; Depósito mensal ; Taxa a.a; Tempo a.a

Saída: Valor Poupado; Juros Ganhos ; Ganho Líquido

Início

Ler valor aplicado: **valorinicial;**

Ler depósito mensal: **depmensal;**

Ler taxa a.a: **taxa;**

Ler tempo a.a: **tempo;**

Continua...



Continua...

Calculo Valor Poupado: $\text{valorpop} = \text{valorinicial} + (\text{depmsal} * 12);$

Calculo Juros ganhos: $\text{juros} = \text{valorpop} * \text{taxa} * \text{tempo};$

Calculo Ganho Liquido: $\text{ganholiquido} = \text{valorpop} + \text{juros};$

Escreva: Valor Poupado: $\text{valorpop};$

Escreva Juros ganhos: $\text{juros};$

Escreva Ganho Liquido : $\text{ganholiquido};$

Fim



Lógica de Programação

1. Definição de Problema
2. Análise
3. Algoritmo
 1. Sequencial
 2. Condicional – conceitos básicos de lógica
 3. Interação



Lógica de Programação

- Proposições
 - Sentenças declarativas – Variável : **Casado**
 - Valor Verdade – **V** ou **F**

Exemplo

- I. Marcos se casou a dois anos com Joaquina.
Logo Status **Casado** = **V**



Operadores Lógicos / Preposições

1. Negação = “Não”
2. Conjunção = “E”
3. Disjunção = “Ou”
4. Condicionais “Se (...) Então”



Negação

Exemplo: Estado de negação é o inverso da afirmação.

estado = “estamos alegres de estudar programação”

Tabela da Verdade “Não”

ESTADO	ESTADO*
V	F
F	V



Conjunção

Exemplo: A procura do dia ideal

AF1 = “Hoje é quarta-feira”

AF2 = “Está nevando”

Tabela da Verdade “E”

AF1	AF2	AF1 \wedge AF2
V	V	V
V	F	F
F	V	F
F	F	F



Disjunção

Exemplo: Churrasco com os amigos

AF1 = “Trazer cerveja”

AF2 = “Trazer refrigerante”

Tabela da Verdade “Ou”

AF1	AF2	AF1 V AF2
V	V	V
V	F	V
F	V	V
F	F	F



Condicional

Exemplo: Será que vou passar na prova?

AF1 = “Estudou para prova” - **premissa**

AF2 = “Vai tirar 10 na prova” - **consequência**

Tabela da Verdade “Se (...) Então”

AF1	AF2	AF1 \Rightarrow AF2
V	V	V
V	F	F
F	V	V
F	F	F



Precedência dos Operadores

OPERADOR	PRECEDÊNCIA
*	1
\vee	2
\wedge	2
\Rightarrow	3

Em uma ambiguidade?
Associatividade a direita

Expressão:

$AF1 \wedge AF2 \Rightarrow AF1 \vee * AF2 \wedge * AF1$



Precedência dos Operadores

\neg : negação “Não”

\vee / \wedge : conjunção/disjunção “E” / “Ou”

\Rightarrow : condição “Se (...) Então”

Expressão:

$$[AF1 \vee (* AF2)] \Rightarrow [AF1 \wedge AF2]$$

$$\begin{array}{ccc} \text{V} & & \text{V} \quad \text{F} \\ \hline & & \hline \text{V} & \text{F} & = \text{F} \end{array}$$

Onde: $AF1 = \text{V}$; $AF2 = \text{F}$



Precedência dos Operadores

\neg : negação “Não”

\vee / \wedge : conjunção/disjunção “E” / “Ou”

\Rightarrow : condição “Se (...) Então”

Expressão:

$$AF1 \wedge * AF3 \Rightarrow AF2 \vee * AF3 \wedge AF1 \vee * AF1$$

Onde: $AF1 = V$; $AF2 = F$; $AF3 = F$

