

Lógica de Programação

André Luyde da Silva Souza

Aula 01: Introdução a Lógica de Programação

Objetivos

- Entender a necessidade de se utilizar uma linguagem formal para construir algoritmos.
- Compreender os conceitos de variáveis, constantes e tipos de dados.
- Conhecer os conceitos de comandos de atribuição, entrada e saída de dados.
- Conhecer os operadores aritméticos e a ordem de precedência.
- Construir nossos primeiros algoritmos em Portugol.

Linguagens de Programação - Introdução

- Para que os computadores sejam capazes de interpretar os algoritmos que desenvolvemos, precisamos transformar a sequência de passos que escrevemos em linguagem natural para uma linguagem que possa ser “entendida” pelo computador.
- Essas linguagens são chamadas de linguagens de programação

Linguagens de Programação - Portugal

- O Portugal é uma linguagem de programação que une o formalismo das linguagens de programação à facilidade de compreensão da linguagem natural.

Linguagens de Programação - Variável

- O que é?
 - É uma posição nomeada de memória
 - Usada para guardar um valor que pode ser modificado pelo programa.
- Declaração de Variáveis
 - Área de memória reservada pelo computador para o nosso uso.
 - Sintaxe:
var
nome_da_variavel : tipo_da_variável

Linguagens de Programação - Variável

var

nome_da_variavel : tipo_da_variável

■ var

- A palavra var é utilizada para indicar o início do bloco de declaração de variáveis de um algoritmo
- É escrita apenas uma vez, independente do número de variáveis a serem declaradas.

■ nome_da_variavel

- Nome que vamos dar para a área de memória reservada
- Toda variável tem um nome através do qual é referenciada

■ tipo_da_variável

- É o tipo de dados que armazenamos na variável para que o computador saiba o tamanho do espaço de memória que reservará.

Linguagens de Programação - Variável

■ Nomes das variáveis

Quadro 2.1: Regras para nomear variáveis

REGRA	EXEMPLO
Inicie sempre por um caractere alfabético, nunca por um número.	Nome (correto) - 1nome (errado)
Não utilize caracteres especiais como " , () / * ; + .	Nome(M); N*B
Não coloque espaços em branco ou hífen entre nomes.	salario-bruto
Utilize, se necessário, <i>underline</i> .	salario_bruto
Crie suas variáveis com nomes sugestivos.	Se vai guardar salário de funcionários, dê à variável o nome salario .

Linguagens de Programação - Tipos de Variáveis

- Tipo inteiro: Declaramos variáveis do tipo numérico inteiro quando precisarmos armazenar valores inteiros, positivos ou negativos (0, 1, 5, 7, -10, -5...).
- Exemplo
 - Idade
 - Número de filhos
- Código:
var
idade: inteiro

Linguagens de Programação - Tipos de Variáveis

- Tipo real: Declaramos variáveis do tipo numérico real para armazenar valores reais; em outras palavras, valores com ponto decimal (5.7, 3.2, -8.5).
- Exemplo
 - Salário
 - peso
- Código:
var
salario: real

Linguagens de Programação - Tipos de Variáveis

- Tipo caractere: Declaramos variáveis do tipo literal caractere para armazenar caractere, que pode ser uma letra, um símbolo ou até mesmo uma palavra.
- Exemplo
 - nome
 - Orientação Sexual (M ou F)
- Código:
var
nome: caractere

Linguagens de Programação - Tipos de Variáveis

- Tipo lógico: Declaramos variáveis do tipo lógico para armazenar valores lógicos. O valor de variáveis desse tipo será sempre VERDADEIRO ou FALSO.
- Código:
var
resultado: logico

Linguagens de Programação - Definição de Variáveis

- Declarar uma variável peso, altura, idade e nome de uma pessoa
- Como ficaria a definição destas variáveis?
- Código:
var
altura, peso: real
idade : inteiro
nome: caractere

Linguagens de Programação - Constantes

- Definição: É uma variável com valor pré-definido que não pode ser modificado por nenhuma função de um programa.
- Obedecem as mesmas regras de nomenclaturas já vistas em variáveis.
- Código:
const
porcentagem < - 0.08

Linguagens de Programação - Comando de Atribuição

- É representado por uma seta ($< -$)
- Sintaxe:
 - identificador $< -$ expressão
- identificador
 - Nome da variável ou constante a ser utilizada.
- expressão
 - Valor ou expressão a ser armazenado
- Exemplos:

var

idade : inteiro

inicio

idade $< -$ 18

Linguagens de Programação - Comentário

- `/*` e `*/` - Indicam o início e o fim do comentário
- Pode ocupar mais de uma linha
- `//` - Comentários até o fim da linha
- Exemplos:

var

idade : inteiro `/* Criando uma variável inteira */`

início

idade < - 18 `// Atribuindo valor à variável`

Linguagens de Programação - Comando de Entrada de Dados

- Responsável pela leitura e armazenamento desses dados na variável que indicarmos.
- Sintaxe:
leia(variavel)
- leia()
 - Função responsável por ler o que o usuário digitou e armazenar o valor na variável indicada.
- Exemplo:
var
idade : inteiro
inicio
leia(idade)

Linguagens de Programação - Comando de Saída de Dados

- Usado para exibir mensagens ou valores para o usuário de nossos programas.
- Exibe no monitor valores de constantes, variáveis ou expressões.
- Sintaxe:
 escreva(expressão)
- escreva()
 - Função responsável por escrever no monitor uma mensagem para o usuário.
- expressão
 - Indica o que será escrito no monitor. É normalmente composta por um texto fixo seguido por uma vírgula e um nome de variável.

Linguagens de Programação - Comando de Saída de Dados

■ Exemplo:

var

idade : inteiro

inicio

leia(idade)

escreva(" A idade digitada é ", idade)

Linguagens de Programação - Estrutura Geral de um Algoritmo

- algoritmo "nome_do_algoritmo"
- const
 - /* Aqui devem ser declaradas todas as constantes do nosso algoritmo seguindo as regras de sintaxe já estudadas. Caso nosso algoritmo não necessite de constantes, esta seção não deve existir.*/
- var
 - /* Aqui devem ser declaradas todas as variáveis do nosso algoritmo seguindo as regras de sintaxe já estudadas. Caso nosso algoritmo não necessita de variáveis, esta seção não deve existir.*/
- inicio
 - /* Aqui ficará o corpo do algoritmo, ou seja, o conjunto de comandos que formará o algoritmo em si.*/
- finalgoritmo

Linguagens de Programação - Estrutura Geral de um Algoritmo

```
1 algoritmo "exemplo"
2 var
3   nome : caractere //criamos a variável nome do tipo cadeia
4 inicio
5   escreva("Digite seu nome: ") //Solicitamos que o usuário
6                               //digite seu nome
7   leia(nome) //lemos para a variável nome o valor digitado
8             //pelo usuário
9
10  escreva("Bom dia ", nome) //imprimimos na tela a mensagem
11                           //bom dia acompanhada pelo nome
12                           //digitado pelo usuário
13 fimalgoritmo //final do algoritmo
```

Linguagens de Programação - Operadores Aritméticos

- Utilizados para operações matemáticas

OPERADORES ARITMÉTICOS	PORTUGUÊS ESTRUTURADO
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Divisão Inteira	\
Exponenciação	^ ou Exp (<base>, <expoente>)
Módulo (resto da divisão)	%

Linguagens de Programação - Operadores Aritméticos

■ Prioridade

Prioridade	Operador	Operação
1 ^a	* / %	multiplicação, divisão, resto da divisão
2 ^a	+ -	adição, subtração

■ Também pode ser estabelecida por parênteses

■ $nota1 * (nota2 + nota3) / 3$

Linguagens de Programação - Expressões Lineares

- Algoritmos que realizam cálculo matemáticos
 - Expressões aritméticas devem ser linearizadas
 - Mapear os operadores da aritmética tradicional para os do Português Estruturado

- Exemplo

$$\left\{ \left[\frac{2}{3} - (5 - 3) \right] + 1 \right\} . 5$$

- Computacionalmente?

$$((2/3 - (5 - 3)) + 1) * 5$$

Linguagens de Programação - Expressões Lineares - Exemplo

- Fórmula para equações de segundo grau

$$X = \frac{-B \pm \sqrt{\Delta}}{2A}$$

onde

$$\Delta = B^2 - 4AC$$

- Computacionalmente?

$$x = (-B + (B \wedge 2 - 4 * A * C) \wedge (1/2)) / (2 * A)$$

$$x = (-B - (B \wedge 2 - 4 * A * C) \wedge (1/2)) / (2 * A)$$

Dúvidas



Em caso de Dúvidas utilizem o Fórum de Dúvidas na plataforma