

Pseudocódigo

Com base nos conceitos vistos podemos aprofundar nosso estudo sobre os métodos de representação de algoritmos, iniciaremos pelo pseudocódigo.

Pseudocódigo é uma técnica textual de representação de um algoritmo. Também é conhecida como Português Estruturado ou Portugol.

Nele os verbos (ações) disponíveis para utilização são limitados e empregados no imperativo, deve-se evitar as expressões excessivamente longas, estas restrições visam eliminar a possibilidade de ambigüidade.

A técnica é baseada em uma PDL (Program Design Language), que é uma linguagem genérica na qual é possível representar um algoritmo de forma semelhante à das linguagens de programação.

Pseudocódigo – Estrutura

A estrutura de um algoritmo em pseudocódigo pode variar um pouco de acordo com o autor ou com base na linguagem de programação que será utilizada posteriormente, mas essas variações ocorrem apenas na sintaxe, pois a semântica deve ser exatamente a mesma.

A estrutura que empregaremos para a construção de nossos pseudocódigos será a seguinte:

algoritmo “nome” //Tem como objetivo identificar o
//algoritmo, deve-se utilizar um nome o mais significativo
//possível, para facilitar a identificação

var

// Seção de Declarações - Neste ponto são informadas
//quais variáveis, e seus respectivos tipos, serão
// utilizadas no algoritmo

inicio

// Seção de Comandos - Aqui será escrita a seqüência de
// comandos que deve ser executada para solucionar
// o problema em questão

fimalgoritmo //marca o final do algoritmo

Pseudocódigo

As palavras reservadas presentes na estrutura do pseudocódigo apresentado foram:

algoritmo

var

inicio

fimalgoritmo

Pseudocódigo – Exemplo

Observaremos agora um pseudocódigo que recebe um valor inteiro, fornecido pelo usuário, e o retorna no monitor.

algoritmo “exemplo 1”

var x: inteiro

inicio

 leia (x)

 escreva (x)

fimalgoritmo

Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exibe o resultado desta manipulação.

algoritmo “exemplo 2”

var numero, resposta: inteiro

inicio

 escreva (“Digite um número inteiro: ”)

 leia (numero)

 resposta <- numero+2

 escreval (“Resultado (número + 2): ”, resposta)

fimalgoritmo

Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exibe o resultado desta manipulação.

algoritmo “exemplo 2”

var n: inteiro

inicio

 escreva (“Digite um número inteiro: ”)

 leia (n)

$n \leftarrow n + 2$

 escreval (“Resultado (número + 2): ”, n)

fimalgoritmo

Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exibe o resultado desta manipulação.

algoritmo “exemplo 2”

var n: inteiro

inicio

 escreva (“Digite um número inteiro: ”)

 leia (n)

 escreval (“Resultado (número + 2): ”, n+2)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1”

var n1, n2, res: inteiro

inicio

 escreva (“Digite o multiplicando inteiro: ”)

 leia (n1)

 escreva (“Digite o multiplicador inteiro: ”)

 leia (n2)

 res <- n1 * n2

 escreva (“Resultado da multiplicação: ”, res)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1b”

var n1, n2: inteiro

inicio

 escreva (“Digite o multiplicando inteiro: ”)

 leia (n1)

 escreva (“Digite o multiplicador inteiro: ”)

 leia (n2)

 escreva (“Resultado da multiplicação: ”, $n1 * n2$)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1c”

var n1, n2: inteiro

inicio

 escreva ("Digite o multiplicando e em seguida ")

 escreva (" o multiplicador (ambos inteiros): ")

 leia (n1, n2)

 escreva ("Resultado da multiplicação: ", n1*n2)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo "exercício 1d"

var n1, n2: inteiro

inicio

 escreva ("Digite o multiplicando e em seguida ")

 escreva (" o multiplicador (ambos inteiros): ")

 leia (n1, n2)

 escreva (n1, "*", n2, "=", n1*n2)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2a”

var valor, resultado: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 resultado <- valor * 0.7

 escreva (“Valor com desconto de 30%: ”)

 escreva (resultado)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2b”

var valor: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 valor <- valor * 0.7

 escreva (“Valor com desconto de 30%: ”)

 escreva (valor:6:2)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2c”

var valor: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 escreva (“Valor com desconto de 30%: ”, valor *
0.7:6:2)

fimalgoritmo

Pseudocódigo – Exercício 3

Faça um pseudocódigo de um algoritmo que receba 4 notas, calcule e mostre a média aritmética entre elas.

algoritmo "exercício 3a"

var

nota1, nota2, nota3, nota4, media: real

inicio

escreva ("Entre com a primeira nota: ")

leia (nota1)

escreva ("Entre com a segunda nota: ")

leia (nota2)

escreva ("Entre com a terceira nota: ")

leia (nota3)

escreva ("Entre com a quarta nota: ")

leia (nota4)

media <- (nota1+nota2+nota3+nota4)/4

**escreva ("A média aritmética entre as notas fornecidas é ",
media:5:2)**

fimalgoritmo

algoritmo "exercício 3b"

var

nota, media: real

inicio

media <- 0

escreva ("Entre com a primeira nota: ")

leia (nota)

media <- media + nota

escreva ("Entre com a segunda nota: ")

leia (nota)

media <- media + nota

escreva ("Entre com a terceira nota: ")

leia (nota)

media <- media + nota

escreva ("Entre com a quarta nota: ")

leia (nota)

media <- media + nota

escreva ("A média aritmética entre as notas fornecidas é ", media/4:5:2)

fimalgoritmo

algoritmo "exercício 3b"

var

nota, media: real

inicio

escreva ("Entre com a primeira nota: ")

leia (media)

escreva ("Entre com a segunda nota: ")

leia (nota)

media <- media + nota

escreva ("Entre com a terceira nota: ")

leia (nota)

media <- media + nota

escreva ("Entre com a quarta nota: ")

leia (nota)

media <- media + nota

escreva ("A média aritmética entre as notas fornecidas é ", media/4:5:2)

fimalgoritmo

Pseudocódigo – Exercício 4

Elabore um algoritmo que calcule a área de um círculo.

$$\text{Área} = \pi * r^2$$

algoritmo "exercício 4"

var

raio, area, ~~pi~~: real

Inicio

pi <- 3.141592

escreva ("Entre com o valor do raio: ")

leia (raio)

area <- pi*raio*raio

escreva ("Área do círculo: ", area)

fimalgoritmo

algoritmo "exercício 4"

var

raio, area, constante: real

Inicio

~~**constante <- 3.141592**~~

escreva ("Entre com o valor do raio: ")

leia (raio)

area <- constante*raio^2

escreva ("Área do círculo: ", area)

fimalgoritmo

algoritmo "exercício 4"

var

raio, area: real

inicio

escreva ("Entre com o valor do raio: ")

leia (raio)

area <- 3.141592*raio^2

escreva ("Área do círculo: ", area)

fimalgoritmo

Pseudocódigo – Exercício 5

Construa o pseudocódigo de um algoritmo capaz de ler dois valores inteiros armazenando-os em variáveis distintas, após o armazenamento o conteúdo das variáveis deve ser trocado e, ao termino do processo, o algoritmo deve retornar o novo conteúdo de cada uma das variáveis .