Trilha Algoritmo

Encontro 04 - Procedimentos e Funções.

vnt/school
powered by venturus

Recapitulação

1. Estrutura de repetição: **repita...ate**
2. Estrutura de repetição: **enquanto...faca**
3. Estrutura de repetição: **para...faca**
4. Exemplos.
5. Atividades.

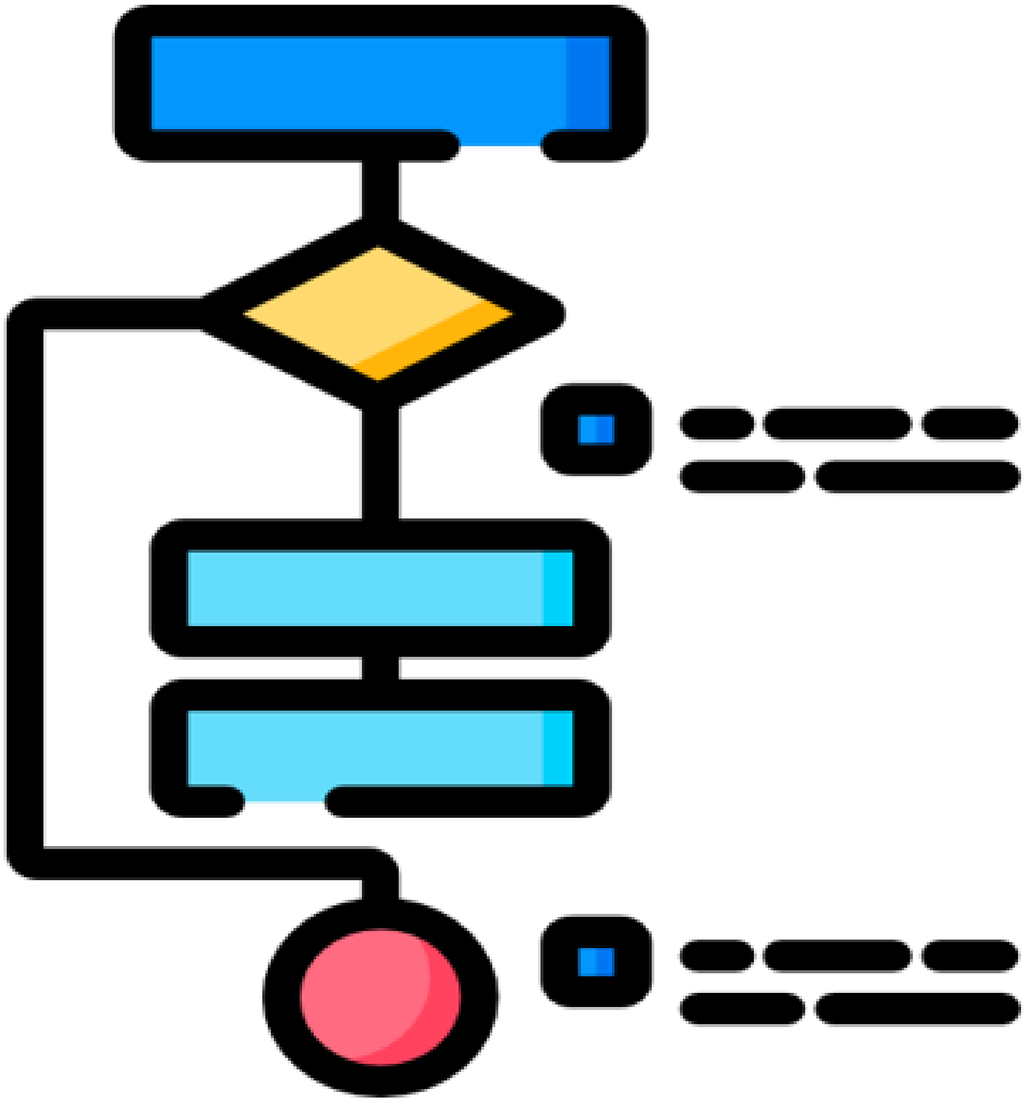
vnt/school
powered by venturus

Agenda

1. O que é subprograma?
2. Procedimentos
3. Funções
4. Exemplos
5. Exercícios

vnt/school
powered by venturus

O que é Subprograma?



vnt/school
powered by venturus

Subprograma

É um programa que auxilia o programa principal através da realização de uma determinada subtarefa.

Também costuma receber os nomes de sub-rotina, subalgoritmo, método ou módulo.

Os subprogramas são chamados dentro do corpo do programa principal como se fossem comandos.

vnt/school
powered by venturus

Subprograma

Após seu término, a execução continua a partir do ponto onde foi chamado.

É importante compreender que a chamada de um subprograma simplesmente gera um desvio provisório no fluxo de execução.

vnt/school
powered by venturus

Subprograma

Quando utilizar tal recurso?

É conveniente utilizá-los quando uma determinada tarefa é efetuada em diversos lugares no mesmo algoritmo.

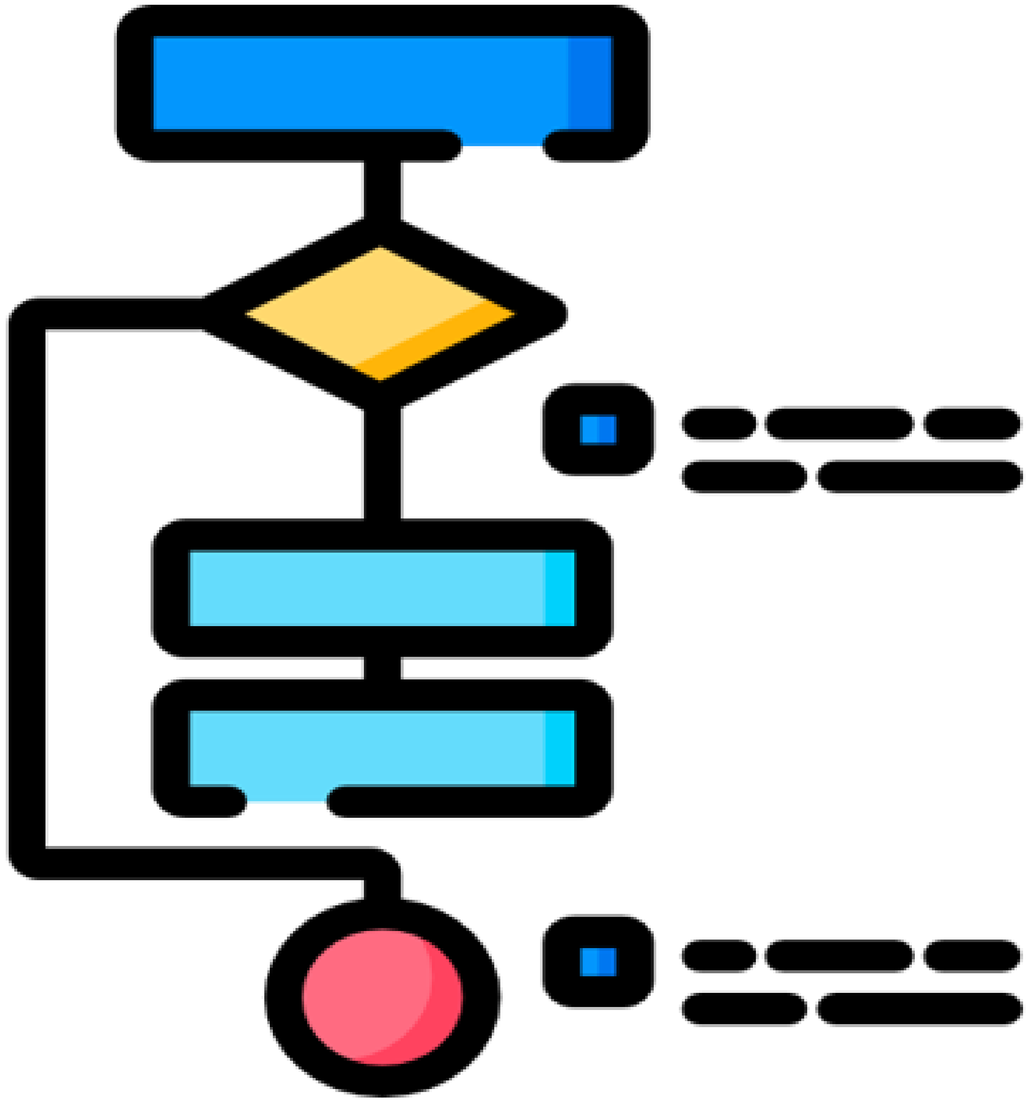
Ao invés de escrever-se um trecho diversas vezes, escreve-se um sub-algoritmo e chama-o diversas vezes.

Subprograma

Eles reduzem o tamanho do algoritmo.
Facilitam a compreensão e visualização do algoritmo.
São declarados no início do algoritmo e podem ser chamados em qualquer ponto após sua declaração.

Funções --> retorna algum valor
Procedimento --> não retorna nada.

Procedimento



vnt /school
powered by venturus

Procedimento

Sintaxe:

Procedimento <identificador> ([var] <parâmetros>)
var
    <declaração de variáveis>
inicio
    <lista de comandos>
fimprocedimento

Procedimento

Identificador: Nome do procedimento.

Passagem de parâmetros por referência: utiliza-se a construção VAR antes dos identificadores para indicar a passagem por referência. Os identificadores são separados por vírgula.

Parâmetros: Entre um mesmo tipo de dados são separados por vírgula. Entre tipos diferentes de dados a separação é feita com ponto-e-vírgulas ';'.

Procedimento

Exemplo:

Procedimento **sem parâmetro**.

Utiliza uma variável local **aux** para armazenar provisoriamente o resultado deste cálculo

```
1  Algoritmo "SomaProcedimento"
2
3  Var
4     n, m, soma, res : inteiro
5
6  procedimento soma
7  var aux: inteiro
8  inicio
9     // n, m e res são variáveis globais
10    aux <- n + m
11    res <- aux
12 fimprocedimento
13
14 Inicio
15    n <- 3
16    m <- -11
17    soma
18    escreva(res)
19 Fimalgoritmo
```

vnt/school
powered by venturus

Procedimento

Exemplo:

A mesma tarefa pode ser executada através de um procedimento **com parâmetros.**

```
1  Algoritmo "SomaProcedimento2"
2
3  Var
4     n,m,soma,res: inteiro
5
6  procedimento soma (x,y: inteiro)
7  inicio
8     // res é variável global
9     res <- x + y
10 fimprocedimento
11
12 Inicio
13    n <- 4
14    m <- -8
15    soma(n,m)
16    escreva(res)
17 Fimalgoritmo
```

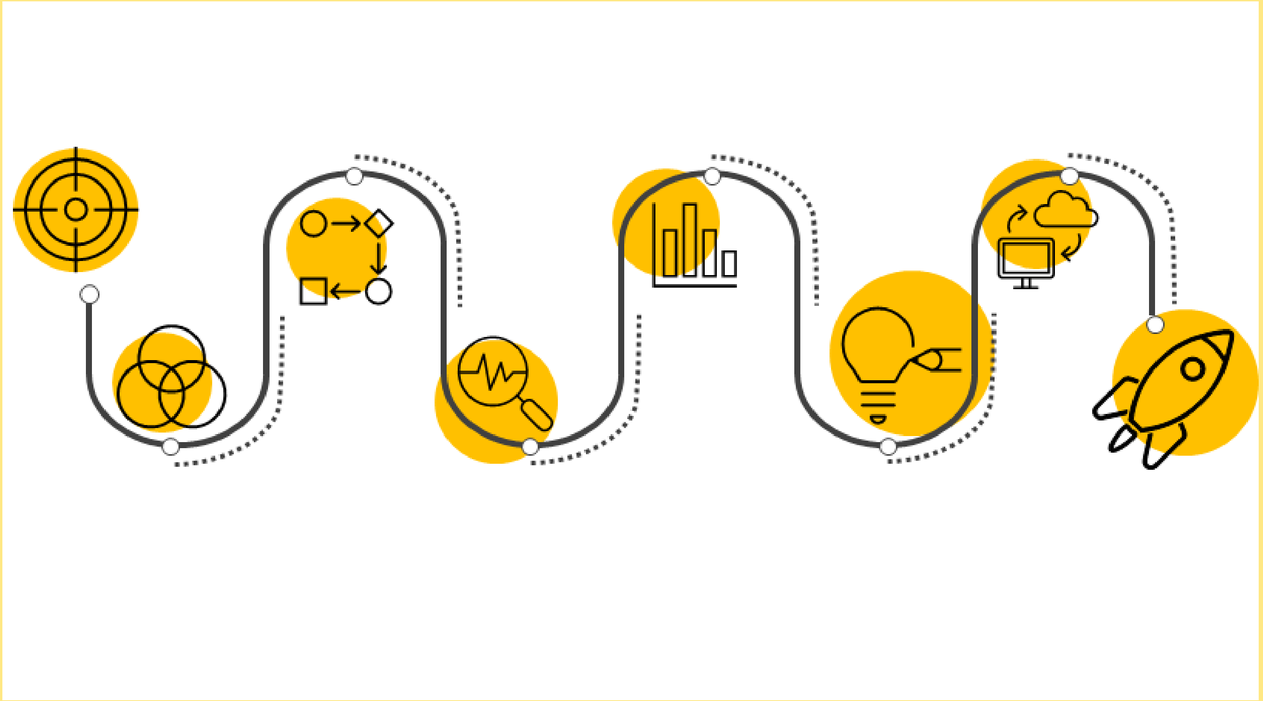
vnt/school
powered by venturus

Procedimento

Imagine um procedimento que envia e-mail. **Precisa retornar resultado?**

Para ler o valor digitado por um usuário é utilizado o procedimento **LEIA.**

Para mostrar um texto na tela é utilizado o procedimento **ESCREVA**.



vnt/school
powered by venturus

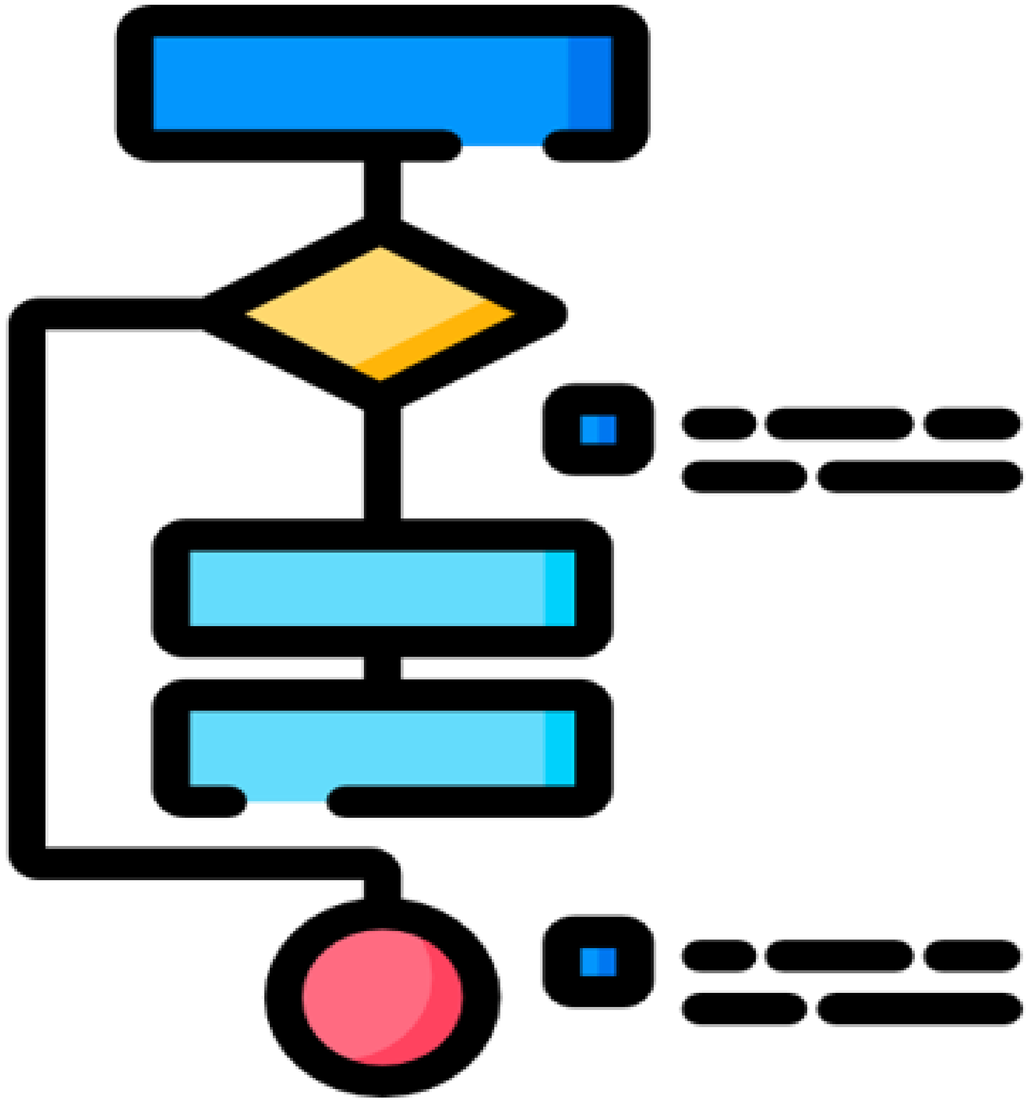
Vamos praticar?

Procedimento

Exercício: Algoritmo para verificar se um valor e par ou ímpar usando rotinas internas ou procedimentos.

vnt/school
powered by venturus

Funções



vnt /school
powered by venturus

Funções

Uma **função** é um instrumento que tem como objetivo retornar um valor ou uma informação.

A **chamada de uma função** é feita através da citação do seu nome seguido opcionalmente de seus argumentos iniciais entre parênteses.

Funções

Sintaxe:

funcao<identificador> ([var]<parâmetro>) <tipo de retorno>
var
    <declaração de variáveis locais>
inicio
    <lista de comandos>
retorne<variável de retorno>
fimfuncao

Funções

Identificador: Nome da função.

Passagem de parâmetros por referência: utiliza-se a construção VAR antes dos identificadores para indicar a passagem por referência. Os identificadores são separados por vírgula.

Parâmetros: Entre um mesmo tipo de dados são separados por vírgula. Entre tipos de dados a separação é feita com ponto-e-vírgulas ';'.

Tipo de retorno da função: Real, Inteiro, Lógico ou Caractere.

Declaração de variáveis locais: idêntica a declaração de variáveis globais. As variáveis declaradas localmente tem validade dentro do escopo da função.

Retorne: local onde é colocado a variável de retorno.

Funções

Exemplo:

Voltando ao exemplo anterior, no qual calculamos a soma.

Pode ser feito através de uma **função sem parâmetros**.

```
1  Algoritmo "SomaFuncao"
2
3  Var
4     n,m,soma,res: inteiro
5
6  funcao soma: inteiro
7  var aux: inteiro
8  inicio
9     // n, m e res são variáveis globais
10    aux <- n + m
11    retorne aux
12 fimfuncao
13
14 Inicio
15    n <- 4
16    m <- -8
17    res <- soma
18    escreva(res)
19 Fimalgoritmo
```

vnt/school
powered by venturus

Funções

Exemplo:

A mesma tarefa pode ser feita com uma **função com parâmetros** passados por valor.

```
1  Algoritmo "SomaFuncao2"
2
3  Var
4     n,m,soma,res: inteiro
5
6  funcao soma (x,y: inteiro): inteiro
7  inicio
8     retorne x + y
9  fimfuncao
10
11 Inicio
12    n <- 4
13    m <- -8
14    res <- soma(n,m)
15    escreva(res)
16 Fimalgoritmo
```

vnt/school
powered by venturus

Funções

Funções Predefinidas do Visualg

O visulag vem com bibliotecas de funções predefinidas que você pode utilizar em seus programas.

Função	Descrição
Cos (valor : real) : real	Cosseno
Exp (<base>,<expoente>)	Potenciação
Maiusc (c : caracter) : caracter	Converte em Maiúscula
Minusc (c : caracter) : caracter	Converte em Minúscula
Radpgrau (valor : real) : real	Converte Radiano para grau.
Raizq (valor : real) : real	Raiz quadrada

Funções

Exemplo:

Qual é a função?

Int = escreva apenas o valor inteiro. **3**

```
1  Algoritmo "Retorna um valor inteiro"
2
3  Var
4  valorReal: real
5  valorInteiro: inteiro
6
7  Inicio
8  valorReal <- 3,5577778999
9  valorInteiro <- int(valorReal)
10 Escreva("Valor Inteiro: ", valorInteiro)
11
12 Fimalgoritmo
```

vnt/school
powered by venturus

Funções

Exemplo:

Qual é a função?

RAIZQ = escreva a raiz quadrada do termo dentro dos parênteses.

```
1  algoritmo "Hipotenusa"
2  var
3     a, b, c : REAL
4  inicio
5
6     ESCREVA ("Digite o lado A do triângulo retângulo: ")
7     LEIA (a)
8     ESCREVA ("Digite o lado B do triângulo retângulo: ")
9     LEIA (b)
10
11    c := RAIZQ ( a*a + b*b )//Cálculo da hipotenusa!!
12
13    ESCREVA ("O valor da hipotenusa é: ", c)
14
15 fimalgoritmo
```

vnt/school
powered by venturus

Vamos Praticar!

Funções

Exemplo:

Encontrar a solução de uma raiz quadrada é encontrar os possíveis valores de x. O valor de **delta** influencia totalmente no resultado final.

$$ax^2 + bx + c = 0$$

$$\Delta = b^2 - 4.a.c$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

delta < 0 : não possui raízes reais.
delta = 0 : possui apenas uma raiz.
delta > 0 : possui duas raízes

Faça um programa que encontre as raízes.
Faça delta como uma função.

Funções

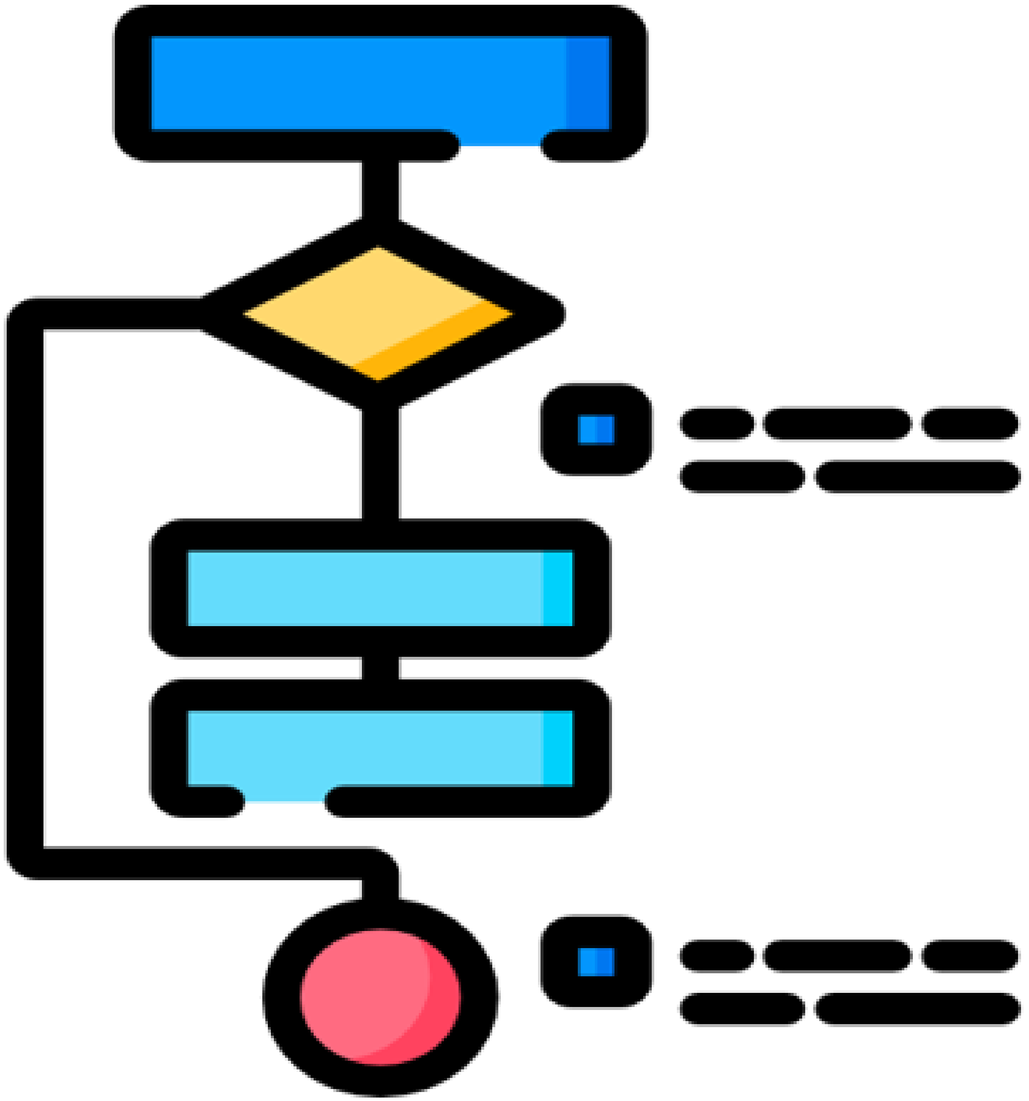
Apresentar o programa no Visualg.

Delta pode ser escrita como uma função.

```
1  algoritmo "EquaçãoDoSegundoGrau"
2  var
3     a, b, c, delta, x1, x2: REAL
4
5  funcao calcula_delta(): REAL
6  var
7     delta : REAL
8  inicio
9     delta := b*b - 4*a*c
10    RETORNE delta
11 fimfuncao
12
13
14
15                    ?
16
17
18
19
```

vnt/school
powered by venturus

Conclusão



vnt /school
powered by venturus

Conclusão

Funções e **procedimentos** são utilizados com muita frequência em desenvolvimento de softwares.

Evita duplicação de código quando é necessário executar a mesma operação várias vezes.

Deixa o entendimento do algoritmo mais intuitivo, pois é retirado a parte complexa do código do fluxo principal do algoritmo, etc.

vnt/school
powered by venturus

Conclusão

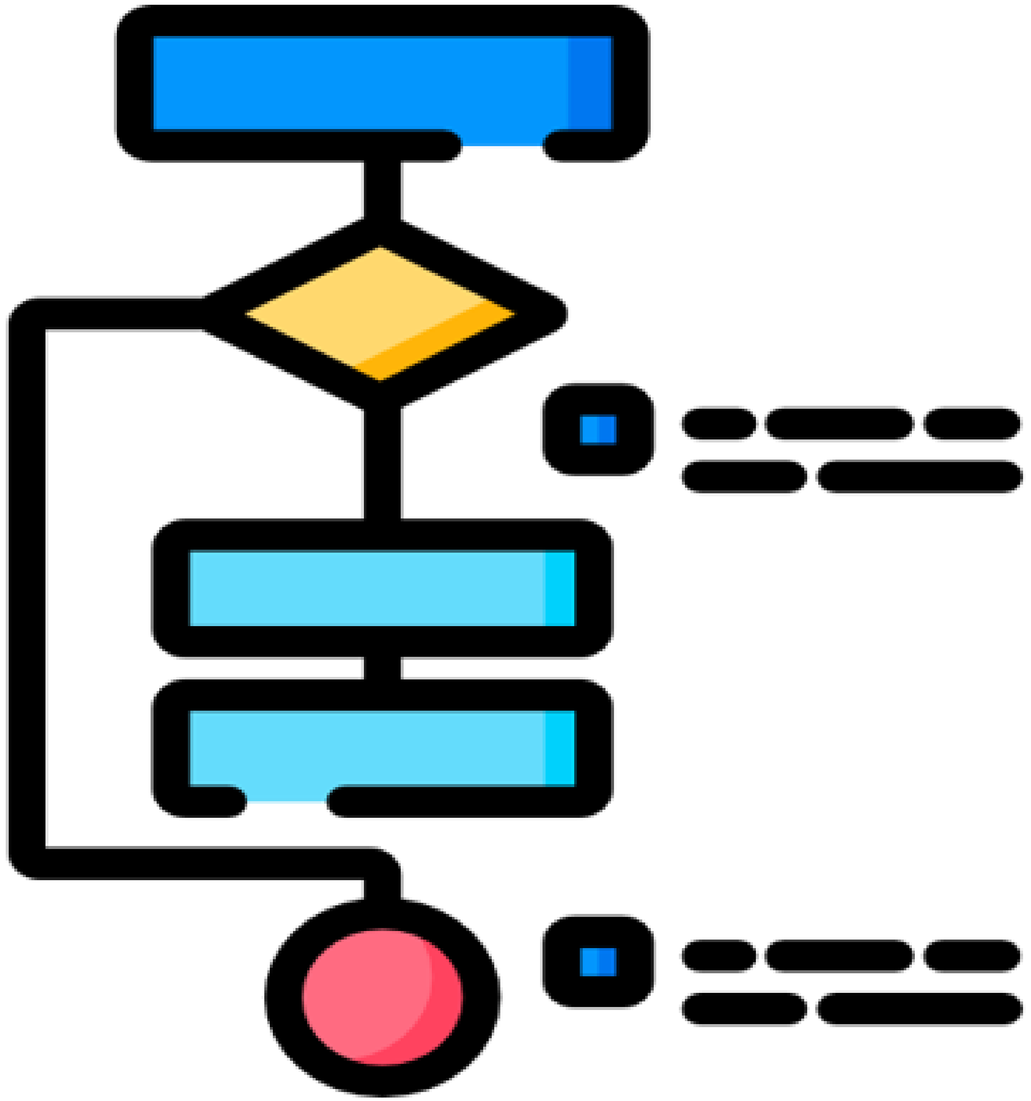
Em linguagens orientada a objeto como java, C++ e C#, funções e procedimentos são chamados de **MÉTODO**.

Podem receber parâmetros e retornam ou não um resultado.

vnt/school
powered by venturus

Coffee time!

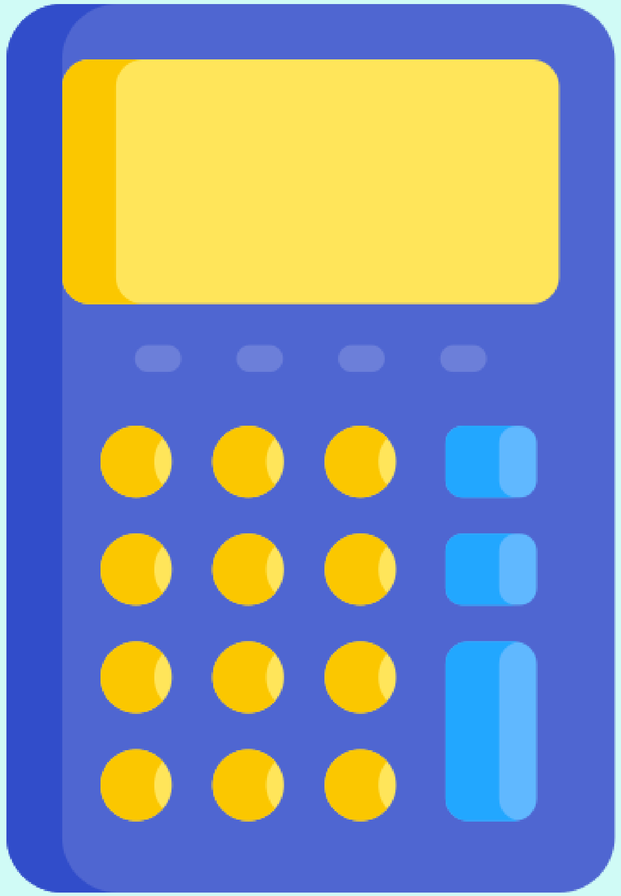
Exercícios



vnt /school
powered by venturus

Prática

Exercícios 1: Faça um algoritmo para subtrair dois valores usando procedimentos.
Imprima o resultado.



vnt/school
powered by venturus

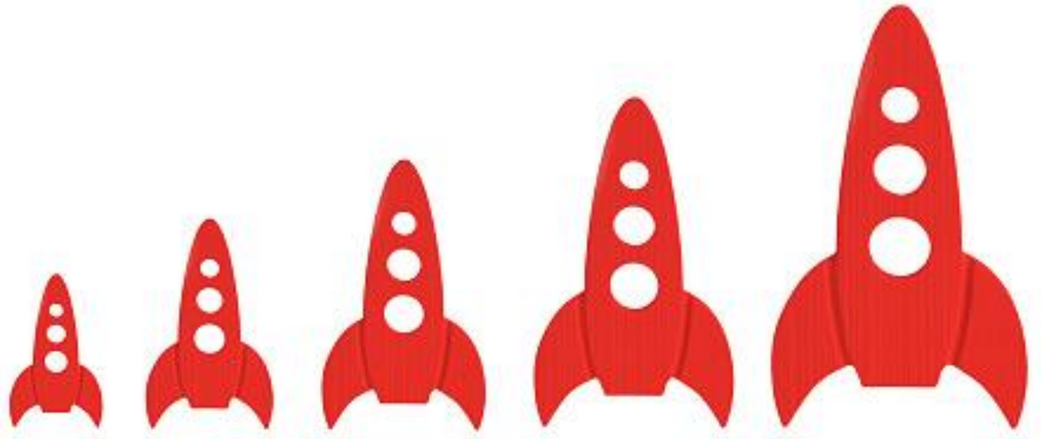
Prática

Exercícios 2: Algoritmo para verificar se um valor e par ou ímpar usando rotinas internas ou procedimentos.

vnt/school
powered by venturus

Prática

Exercícios 3: Faça um programa que leia 4 números e em seguida imprima o resultado em ordem crescente.
Use o conceito de Procedimento para desenvolver o código.



vnt/school
powered by venturus

Prática

Exercício 4: Escreva o algoritmo, usando função, que calcule o fatorial de um número.

A definição de fatorial é:

F(n) = 1, se n = 0 ou n = 1

F(n) = n.F(n-1), se n>1

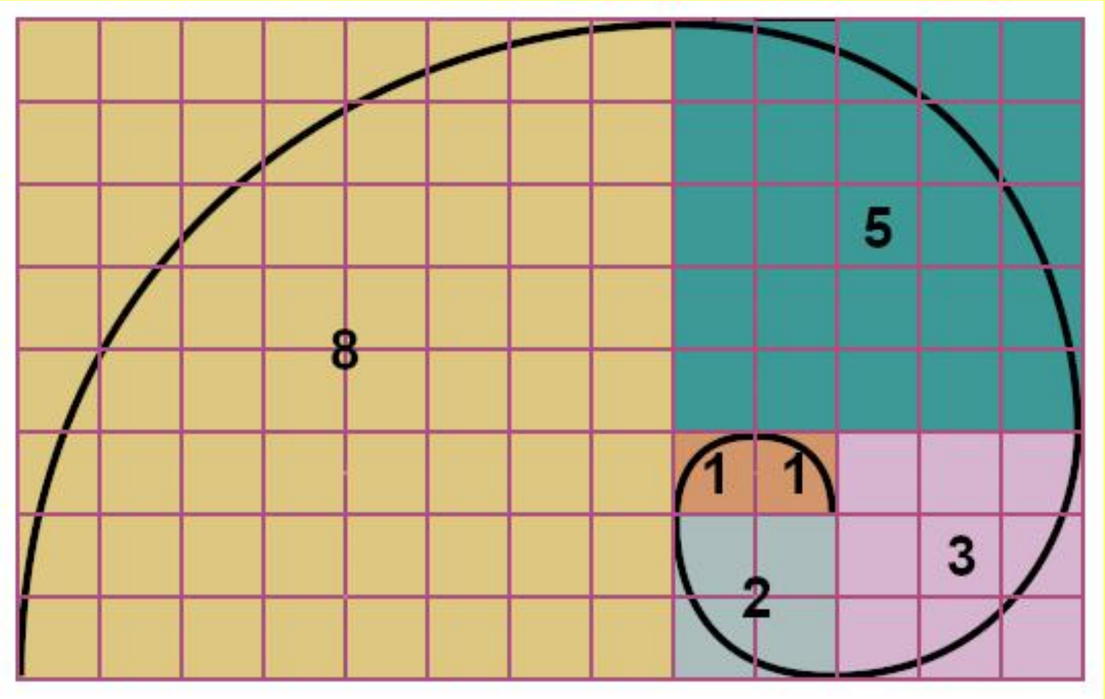
onde n é um número inteiro positivo. Uma propriedade (facilmente verificável) dos fatoriais é que: n! = n . (n-1)! Esta propriedade é chamada de propriedade recursiva: o fatorial de um número pode ser calculado através do fatorial de seu antecessor.

vnt/school
powered by venturus

Prática

Exercícios 5)

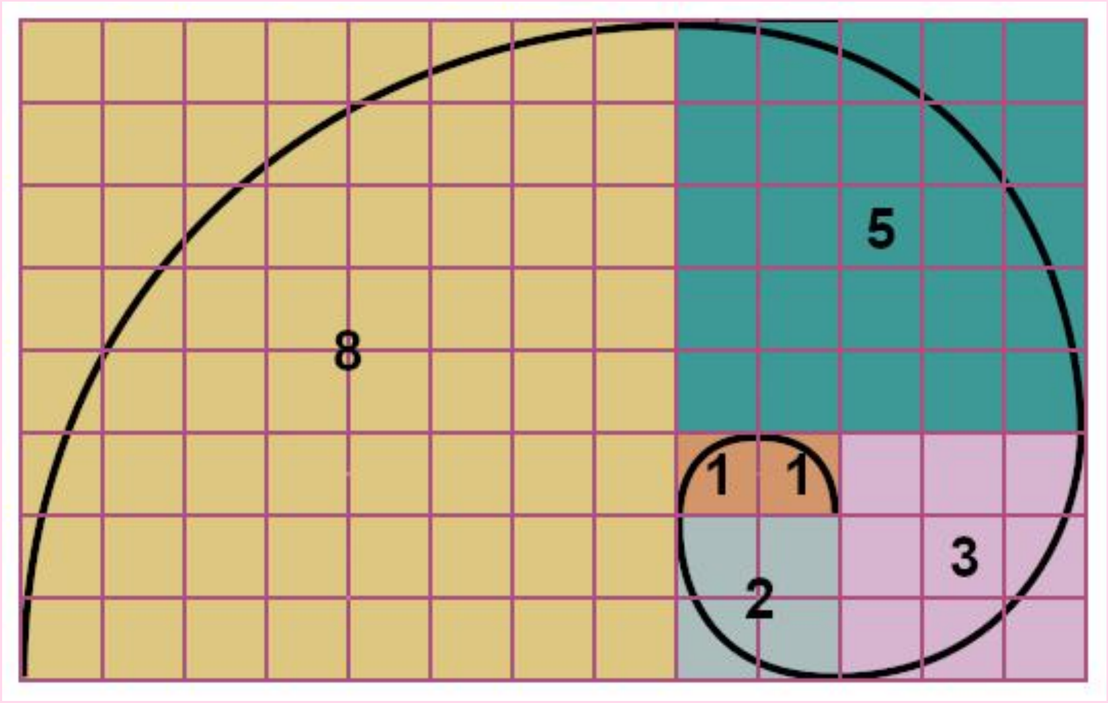
Faça um algoritmo para exibir os 10 primeiros elementos de uma série de Fibonacci usando **procedimentos** com passagem de parâmetros por referência.



Prática

Exercícios 6)

Faça um algoritmo para exibir os 10 primeiros elementos de uma série de Fibonacci usando **função** com passagem de parâmetros por referência.



vnt/school
powered by venturus

Prática

Exercícios 7)

Escreva um programa que lê um valor inteiro (maior ou igual a 1 e menor ou igual a 10) e exibe a tabuada (até 10) de multiplicação do número lido.

Funcao LeNumero(n1,n2:inteiro**):**inteiro;

Prática

Exercícios 8)

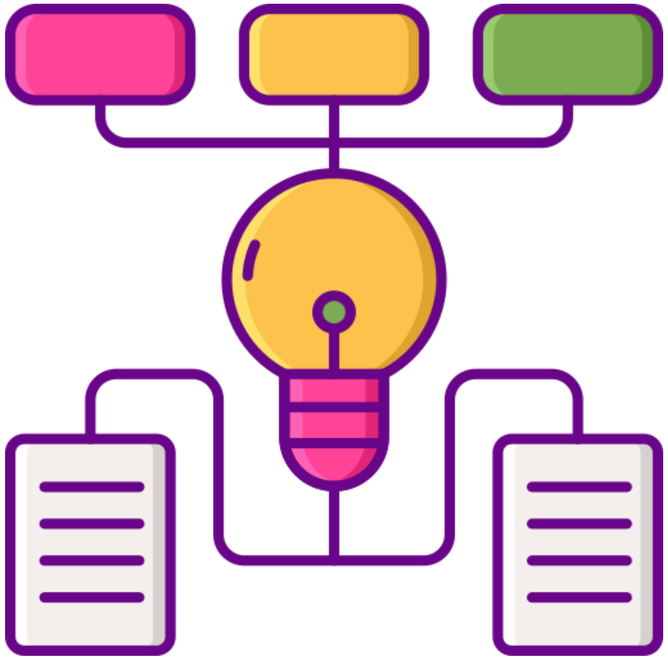
Faça um algoritmo para demonstrar as principais funcões de manipulacão de strings do visualg.

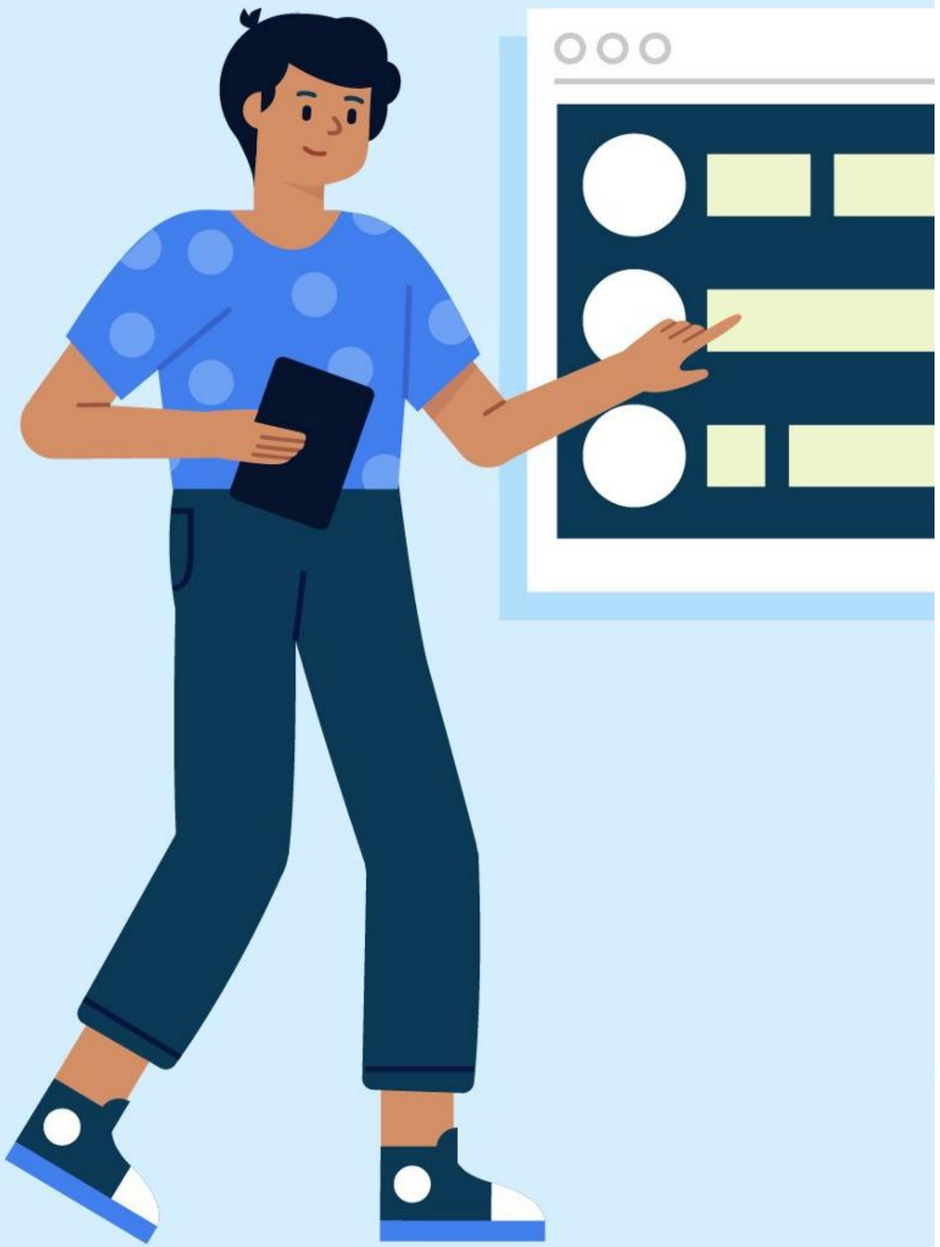
vnt/school
powered by venturus

Dica de hoje

Pressionando (CTRL+J): o visualg mostra uma Lista de funções predefinidas, a sua utilização é muito simples basta selecionar a função desejada e dar um Enter, depois é só passar os parâmetros desejados.

Pressione (F1): basta selecionar "Funções do Visualg". Todas as funções serão descritas detalhadamente.





Comunidade VNT

Referências

[1] A. Goldman, F. Kon, Paulo J. S. Silva; Introdução à Ciência da Computação com Java e Orientação a Objetos (USP). 2006. Ed. USP.
[2] Algoritmo e lógica de programação. Acessado julho/2022: https://visualg3.com.br/
[3] G. Silveira; Algoritmos em Java; Ed. Casa do Código.
[4] M. T. Goodrich, R. Tamassia; Estrutura de dados e algoritmos em Java. Ed Bookman. 2007.
[5] Algoritmo e lógica de programação. Acessado julho/2022: https://www.cursoemvideo.com/
[6] P. Silveira, R. Turini; Java 8 Pratico: lambdas, streams e os novos recursos da linguagem. Ed. Casa do Código.
[7] Linguagem Java: Curso acessado em agosto/2022: https://www.udemy.com/
[8] Linguagem Java: Curso acessado em setembro/2022: https://www.cursoemvideo.com/