



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

## 1. Teoria

**Subprograma** é um programa que auxilia o programa principal através da realização de uma determinada **subtarefa**. Também costuma receber os nomes de **sub-rotina**, **procedimento**, **método** ou **módulo**. Os subprogramas são chamados dentro do corpo do programa principal como se fossem **comandos**. 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**.

Há um caso particular de subprograma que recebe o nome de **função**. Uma **função**, além de executar uma determinada tarefa, **retorna um valor para quem a chamou**, que é o resultado da sua execução. Por este motivo, a chamada de uma função aparece no corpo do programa principal como uma **expressão**, e não como um comando.

Cada **subprograma**, além de ter acesso às variáveis do programa que o chamou (são as **variáveis globais**), pode ter suas próprias variáveis (são as **variáveis locais**), que existem apenas durante sua chamada.

Ao se chamar um subprograma, também é possível passar-lhe determinadas informações que recebem o nome de **parâmetros** (são valores que, na linha de chamada, ficam entre os parênteses e que estão separados por vírgulas). A quantidade dos parâmetros, sua sequência e respectivos tipos não podem mudar: devem estar de acordo com o que foi especificado na sua correspondente declaração.

Para se criar subprogramas, é preciso descrevê-los após a declaração das variáveis e antes do corpo do programa principal. O VisuAlg possibilita declaração e chamada de subprogramas como procedimentos e funções com passagem de parâmetros por valor ou referência.

### Procedimentos

Em VisuAlg, procedimento é um subprograma que não retorna nenhum valor (corresponde ao **procedure** do Pascal). Sua declaração, que deve estar entre o final da declaração de variáveis e a linha início do programa principal, segue a sintaxe abaixo:

```
procedimento <nome-de-procedimento>[(<sequência-de-declarações-de-parâmetros>)]  
// Seção de Declarações Internas  
início  
// Seção de Comandos  
fimprocedimento
```

O **<nome-de-procedimento>** obedece as mesmas regras de nomenclatura das variáveis. Por outro lado, a **<sequência-de-declarações-de-parâmetros>** é uma sequência de **[var] <sequência-de-parâmetros>: <tipo-de-dado>** separadas por ponto e vírgula. A presença (opcional) da palavra-chave **var** indica passagem de parâmetros por referência; caso contrário, a passagem será por valor.

Por sua vez, **<sequência-de-parâmetros>** é uma sequência de nomes de parâmetros (também obedecem a mesma regra de nomenclatura de variáveis) separados por vírgulas.

De modo análogo ao programa principal, a seção de declaração interna começa com a palavra-chave **var**, e continua com a seguinte sintaxe:

```
<lista-de-variáveis> : <tipo-de-dado>
```



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

A passagem de parâmetros é o modo em que o subprograma simplesmente recebe um valor que utiliza durante sua execução. Durante essa execução, os parâmetros passados por valor são análogos às suas variáveis locais, mas com uma única diferença: receberam um valor inicial no momento em que o subprograma foi chamado.

### Funções

Em VisuAlg, função é um subprograma que retorna um valor. De modo análogo aos procedimentos, sua declaração deve estar entre o final da declaração de variáveis e a linha início do programa principal, e segue a sintaxe abaixo:

```
funcao <nome-de-função>(<sequência-de-declarações-de-parâmetros>): <tipo-de-dado>  
// Seção de Declarações Internas  
início  
// Seção de Comandos  
fimfuncao
```

O **<nome-de-função>** obedece as mesmas regras de nomenclatura das variáveis. Por outro lado, a **<sequência-de-declarações-de-parâmetros>** é uma sequência de **[var] <sequência-de-parâmetros>: <tipo-de-dado>** separadas por ponto e vírgula. A presença (opcional) da palavra-chave var indica passagem de parâmetros por referência; caso contrário, a passagem será por valor.

Por sua vez, **<sequência-de-parâmetros>** é uma sequência de nomes de parâmetros (também obedecem a mesma regra de nomenclatura de variáveis) separados por vírgulas.

O valor retornado pela função será do tipo especificado na sua declaração (logo após os dois pontos). Em alguma parte da função (de modo geral, no seu final), este valor deve ser retornado através do comando retorne.

De modo análogo ao programa principal, a seção de declaração interna começa com a palavra-chave var, e continua com a seguinte sintaxe:

```
<lista-de-variáveis>:<tipo-de-dado>
```

### Passagem de Parâmetros por Referência

Há ainda uma outra forma de passagem de parâmetros para subprogramas: é a passagem por referência. Neste caso, o subprograma não recebe apenas um valor, mas sim o **endereço** de uma variável global, também chamado de **Ponteiro**. Portanto, qualquer modificação que for realizada no conteúdo deste parâmetro afetará também a variável global que está associada a ele. Durante a execução do subprograma, os parâmetros passados por referência são análogos às variáveis globais. No VisuAlg essa passagem é feita através da palavra **var** na declaração do parâmetro.

### Recursão e Aninhamento

A atual versão do VisuAlg permite recursão, isto é, a possibilidade de que um subprograma possa chamar a si mesmo. A função do exemplo abaixo calcula recursivamente o fatorial do número inteiro que recebe como parâmetro:



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

*funcao fatorial (v: inteiro): inteiro*

*inicio*

*se v <= 2 entao*

*retorne v*

*senao*

*retorne v \* fatorial(v-1)*

*fimse*

*fimfuncao*

O VisuAlg não é permitido o aninhamento de subprogramas, isto é, cada subprograma também pode ter seus próprios subprogramas. No entanto, esta característica dificulta a elaboração dos compiladores e, na prática, não é muito importante.

**2. Objetivos:** Familiarizar com variáveis indexadas e sua aplicação.

### 3. Procedimentos

**3.1 Procedimento:** 1) Abra o aplicativo VisuAlg. 2) Copiar o seguinte algoritmo. 3) Verifique a execução através do comando de linha F8 e o resultado da execução na área das variáveis e o seu conteúdo.

*Algoritmo "Procedimento sem passagem de parâmetro"*

*// Disciplina : MIC*

*// Função : Exemplo de uso de Procedimento*

*// Autor(a) : Referência ao Manual do Visualg da UFPR*

*// Data atual : 19/09/2018*

*Var*

*// Seção de Declarações das variáveis*

*n, m, soma, res: inteiro*

*procedimento soma*

*var aux: inteiro*

*inicio*

*//n, m e res são variáveis globais*

*aux <- n + m*

*res <- aux*

*fimprocedimento*

*Inicio*

*// Seção de Comandos, procedimento, funções, operadores, etc...*

*n <- 4*

*m <- -9*

*soma*

*escreval(res)*

*Fimalgoritmo*

**Escreva as Suas observações:** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_  
\_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

**3.2 Procedimento:** 1) Abra o aplicativo VisuAlg. 2) Copiar o seguinte algoritmo. 3) Verifique a execução através do comando de linha F8 e o resultado da execução na área das variáveis e o seu conteúdo.

```
Algoritmo "Função com passagem de parâmetros"
// Disciplina : MIC
// Função : Exemplo de uso de Função
// Autor(a) : Referência ao Manual do Visualg da UFPR
// Data atual : 19/09/2018
Var
// Seção de Declarações das variáveis
n, m, soma, res: inteiro

procedimento soma (x,y: inteiro)
inicio
//res é variável global
res <- x + y
fimprocedimento

Inicio

n <- 4
m <- -9
soma(n,m)
escreval(res)

Fimalgoritmo
```

**3.3. Procedimento:** 1) Abra o aplicativo VisuAlg. 2) Copiar o seguinte algoritmo. 3) Verifique a execução através do comando de linha F8 e o resultado da execução na área das variáveis e o seu conteúdo.

```
Algoritmo "Função sem passagem de parâmetros"
// Disciplina : MIC
// Função : Exemplo de uso de Função
// Autor(a) : Referência ao Manual do Visualg da UFPR
// Data atual : 19/09/2018
Var
// Seção de Declarações das variáveis
n, m, soma, res: inteiro

funcao soma: inteiro
var aux: inteiro
inicio
// n, m e res são variáveis globais
aux <- n + m
retorne aux
fimfuncao
```



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

*Início*

```
n <- 4  
m <- -9  
res <- soma  
escreval(res)
```

*Fimalgoritmo*

**3.4. Procedimento:** 1) Abra o aplicativo VisuAlg. 2) Copiar o seguinte algoritmo. 3) Verifique a execução através do comando de linha F8 e o resultado da execução na área das variáveis e o seu conteúdo.

```
Algoritmo "Função com passagem de parâmetros"  
// Disciplina : MIC  
// Função : Exemplo de uso de Função  
// Autor(a) : Referência ao Manual do Visualg da UFPR  
// Data atual : 19/09/2018  
Var  
// Seção de Declarações das variáveis  
n, m, soma, res: inteiro
```

```
funcao soma (x,y: inteiro): inteiro  
inicio  
retorne x + y  
fimfuncao
```

*Início*

```
n <- 4  
m <- -9  
res <- soma(n,m)  
escreva(res)
```

*Fimalgoritmo*

**3.5. Procedimento:** 1) Abra o aplicativo VisuAlg. 2) Copiar o seguinte algoritmo. 3) Verifique a execução através do comando de linha F8 e o resultado da execução na área das variáveis e o seu conteúdo.

```
Algoritmo "Recursividade"  
// Disciplina : MIC  
// Função : Exemplo de uso de Recursividade  
// Autor(a) : Referência ao Manual do Visualg da UFPR  
// Data atual : 19/09/2018  
Var  
// Seção de Declarações das variáveis  
x: inteiro
```

```
funcao fatorial (v: inteiro): inteiro  
inicio  
se v <= 2 entao
```



### Lab5: Algoritmo – Procedimentos e Funções

Alunos: \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

```
    retorne v
senao
    retorne v * fatorial(v-1)
fimse
fimfuncao

Inicio

escreval("Entre com um inteiro")
leia(x)
escreval(fatorial(x))

Fimalgoritmo
```

**4. Conclusão das atividades:** Neste experimento verificou-se como o Visualg possibilita trabalhar com subprogramas, também conhecidos como sub-rotinas. Aprendemos como declarar Procedimentos e Funções e a execução dos mesmos e a verificação de resultados.

**5. Escreva as suas conclusões:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Exercício de aplicação:** Fazer um algoritmo no VisuAlg para ler 3 notas (entre 0 e 20) de um aluno, calcular a média das 3 notas e apresentar a classificação qualitativa:

- média entre 0 e 4: Mau
- média entre 5 e 9: Medíocre
- média entre 10 e 13: Suficiente
- média entre 14 e 17: Bom
- média entre 18 e 20: Muito Bom

**OBS.: Utilizar uma função para calcular a média e um procedimento para escrever a classificação qualitativa.**