

# Passagens de Parâmetros

Roberto Rocha

# Parâmetros!

# Parâmetros

A utilização de variável global auxilia muito no desenvolvimento de programas. Porém possui desvantagens sendo a principal delas a utilização do mesmo nome por parte do programa chamador e da rotina.

O processo de passagem de parâmetro permite estabelecer uma linha de comunicação entre os conteúdos dos trechos de sub-rotinas e dos trechos de programa que chamam essas sub-rotinas, principalmente quando se usam variáveis com escopo local.

A passagem de parâmetro é efetivada quando se usam variáveis locais, e a variável local de uma sub-rotina necessita receber ou enviar conteúdo para a variável local de outra sub-rotina. A passagem de parâmetro ocorre entre o trecho de programa chamador e o trecho de programa chamado. Um trecho de programa chamador pode passar um conteúdo para o trecho de programa chamado processar, e um trecho de programa chamado pode receber de volta o conteúdo do trecho do programa chamador.

# Passagem de Parâmetro por Valor

O conteúdo passado como parâmetro é copiado para o módulo que neste caso assume o papel de variável local da sub-rotina.

Qualquer modificação no conteúdo do parâmetro na rotina não afeta o valor do conteúdo do parâmetro no módulo chamador, ou seja, o processamento é executado apenas dentro da sub-rotina, ficando o resultado dessa operação "preso" na própria sub-rotina.

Quando se utiliza passagem de parâmetros em uma sub-rotina, é necessário mencionar no título da sub-rotina o uso desse parâmetro entre parênteses.

**procedimento** <nomeprocedimento> ([var] **parâmetros**)

**var**

<declaração das variáveis locais ao procedimento>

**início**

<lista de comandos>

**fimprocedimento**

# Passagem de Parâmetro por Valor

Exemplo: Elaborar um procedimento que escreva o sucessor de um número dado. Em seguida fazer um programa que chame esse procedimento.

```
procedimento sucessor(numero:inteiro)
  var
  inicio
    numero ← numero +1
    escreva("O sucessor é = ",numero)
  fimprocedimento
```

Algoritmo "ex01"

```
var n:inteiro
Inicio
  leia(n)
  sucessor(n)
finalgoritmo
```

# Passagem de Parâmetro por Valor

Exemplo: Elaborar um procedimento que escreva o sucessor de um número dado. Em seguida fazer um programa que chame esse procedimento.

→ procedimento sucessor(numero:inteiro)  
    var  
    inicio  
→     numero ← numero +1  
→     escreva("O sucessor é = ",numero)  
→ fimprocedimento

Algoritmo "ex01"  
→ var n:inteiro  
  Inicio  
→   leia(n)  
→   sucessor(n)  
→ fimalgoritmo

Local	Variável	
Pprincipal	n	5
sucessor	numero	5 6

O sucessor é = 6

# Passagem de Parâmetro por Valor

```
procedimento sucessor(numero:inteiro)
var
inicio
    numero ← numero +1
    escreva("O sucessor é = ",numero)
fimprocedimento
```

Algoritmo "ex01"

```
var n:inteiro
Inicio
    leia(n)
    sucessor(n)
finalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4
5  int main()
6  {
7      setlocale(LC_ALL, "portuguese");
8      int n;
9      printf("Digite um valor:");
10     scanf("%d", &n);
11     sucessor(n);
12     return 0;
13 }
14 void sucessor(int numero)
15 {
16     numero = numero+1;
17     printf("O sucessor é = %d\n", numero);
18 }
19
```

```
Digite um valor:5
O sucessor é = 6
```

```
Process returned 0 (0x0)   execution time : 14.253 s
Press any key to continue.
```

# Passagem de Parâmetro por Valor

ler dois valores no programa principal e passar como parâmetro esses valores imprimir no procedimento a soma desses valores

```
procedimento sucessor(numero:inteiro)
var
inicio
    numero ← numero +1
    escreva("O sucessor é = ",numero)
fimprocedimento
```

Algoritmo "ex01"

```
var n:inteiro
Inicio
    leia(n)
    sucessor(n)
finalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4
5  int main()
6  {
7      setlocale(LC_ALL, "portuguese");
8      int n;
9      printf("Digite um valor:");
10     scanf("%d", &n);
11     sucessor(n);
12     return 0;
13 }
14 void sucessor(int numero)
15 {
16     numero = numero+1;
17     printf("O sucessor é = %d\n", numero);
18 }
19
```

```
Digite um valor:5
O sucessor é = 6
```

```
Process returned 0 (0x0)   execution time : 14.253 s
Press any key to continue.
```



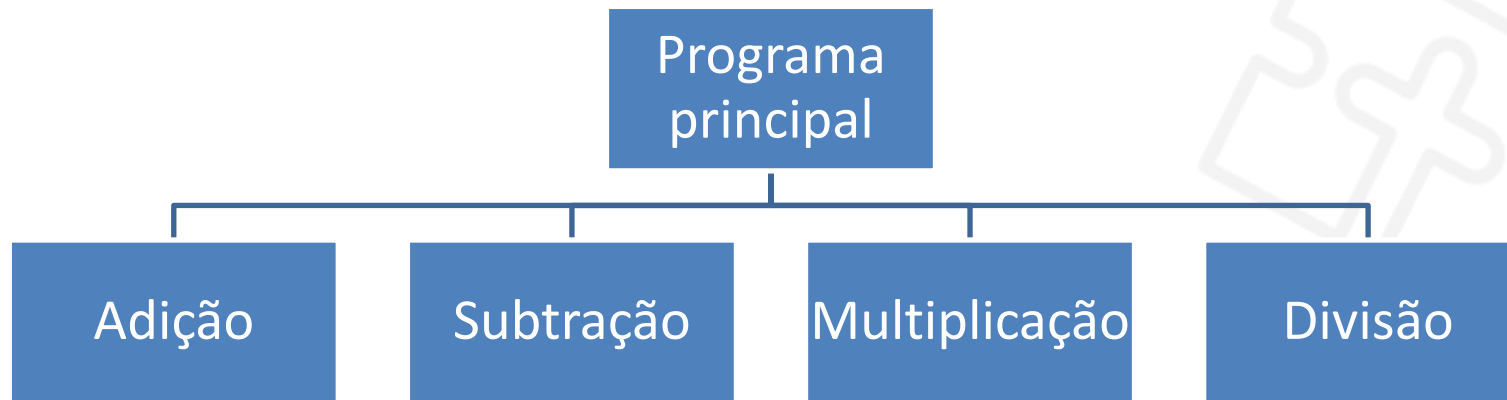
# Procedimentos

## Exercício de fixação:

Desenvolver um programa de computador que simule as operações básicas de uma calculadora que opere com a entrada de dois valores do tipo real após a escolha da operação a ser executada.

O programa deve apresentar uma lista de opções (menu) com as operações matemáticas de **adição**, **subtração**, **multiplicação** e **divisão**, além de uma opção de saída do programa. Escolhida a opção desejada, deve ser solicitada a entrada de dois valores numéricos - **NO PROGRAMA PRINCIPAL** - para que seja possível executar o processamento escolhido. Após a execução da operação, o **programam principal** deve apresentar o resultado. Após a execução de qualquer uma das operações de calculo, o programa deve voltar para o menu de seleção. Utilize passagem de parâmetros por valor.

# Calculadora



# Calculadora

Variáveis globais

var  
r:real

Algoritmo "Calculadora"

var

op:inteiro

a,b:real

procedimento leValores()

var

inicio

leia(a,b)

fimprocedimento

Inicio

faça

leValores()

menu()

leia(op)

escolha (op)

1: somaValores(a,b)

2: subtraiValores(a,b)

3: multiplicaValores(a,b)

4:divideValores(a,b)

5:leValores()

fimescolha

se (op<>9 e op<>5) entao

escreva(r)

fimse

enquanto (op<>9)

finalgoritmo

Procedimento interno  
ao programa principal

Programa  
principal

Adição

Subtração

Multiplicação

Divisão

procedimento menu()

var

inicio

escreva("menu:")

escreva("1 – adição")

escreva("2 – subtração")

escreva("3 – multiplicação")

escreva("4 – divisão")

escreva("5 – novos números")

escreva("9 – sair do programa")

fimprocedimento

# Calculadora

var

r:real

Algoritmo "Calculadora"

var

op:inteiro

a,b:real

procedimento leValores()

var

inicio

leia(a,b)

fimprocedimento

Inicio

faça

leValores()

menu()

leia(op)

escolha (op)

1: somaValores(a,b)

2: subtraiValores(a,b)

3: multiplicaValores(a,b)

4: divideValores(a,b)

5: leValores()

fimescolha

se (op<>9 e op<>5) então

escreva(r)

fimse

enquanto (op<>9)

finalgoritmo

```
12 int main()
13 {
14     setlocale(LC_ALL, "portuguese");
15     int op;
16     float a,b;
17     void leValores()
18     {
19         printf("Valores para calcular:\n");
20         printf("Digite o 1 valor:");
21         scanf("%f", &a);
22         printf("Digite o 2 valor:");
23         scanf("%f", &b);
24     }
25     leValores();
26     do
27     {
28         menu();
29         printf("Escolha:");
30         scanf("%d", &op);
31         switch (op)
32         {
33             case 1:
34                 somaValores(a,b);
35                 break;
36             case 2:
37                 subtraiValores(a,b);
38                 break;
39             case 3:
40                 multiplicaValores(a,b);
41                 break;
42             case 4:
43                 divideValores(a,b);
44                 break;
45             case 5:
46                 leValores();
47                 break;
48         }
49         if (op!=9 && op!=5)
50         {
51             printf("O resultado da operação = %5.2f\n", r);
52             system("PAUSE");
53         }
54     }
55     while (op!=9);
56     return 0;
57 }
```

procedimento menu()

var

inicio

escreva("menu:")

escreva("1 – adição")

escreva("2 – subtração")

escreva("3 – multiplicação")

escreva("4 – divisão")

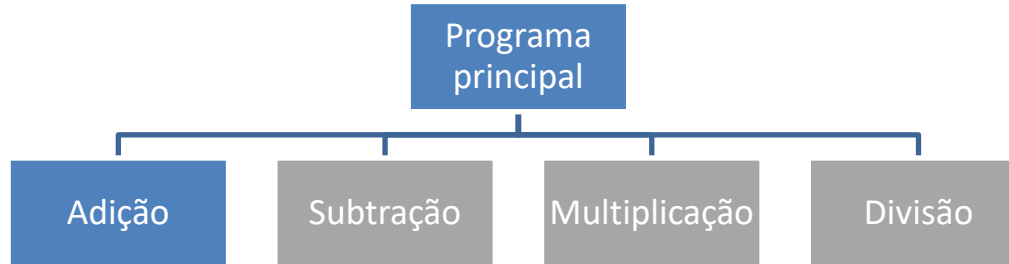
escreva("5 – novos números")

escreva("9 – sair do programa")

fimprocedimento

```
54 void menu()
55 {
56     system("CLS");
57     printf("menu:\n");
58     printf("1 - adição\n");
59     printf("2 - subtração\n");
60     printf("3 - multiplicação\n");
61     printf("4 - divisão\n");
62     printf("5 - novos numeros\n");
63     printf("9 - sair do programa\n");
64 }
```

# Calculadora - adição

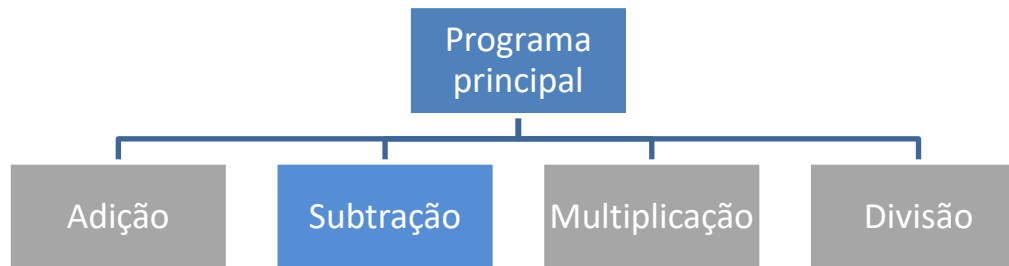


deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

```
procedimento somaValores(n1,n2:real);  
  var  
  inicio  
     $r \leftarrow n1 + n2$   
  fimprocedimento
```

```
65 void somaValores(float n1, float n2)  
66 {  
67     r = n1 + n2;  
68 }
```

# Calculadora - Subtração

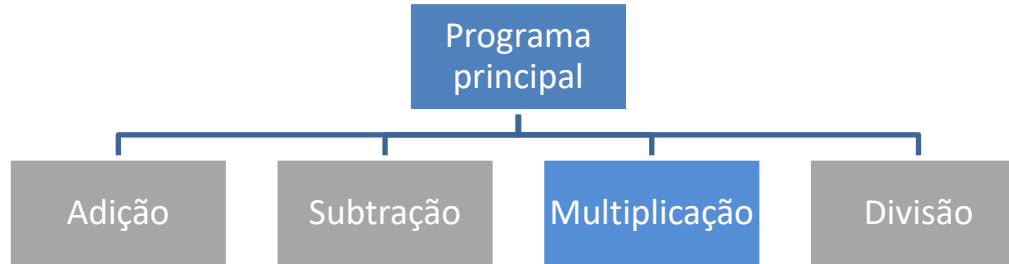


deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

```
procedimento subtraiValores(numero1,numero2:real);  
  var  
  inicio  
     $r \leftarrow \text{numero1} - \text{numero2}$   
  fimprocedimento
```

```
69 void subtraiValores(float numero1, float numero2)  
70 {  
71     r = numero1 - numero2;  
72 }
```

# Calculadora - Multiplicação



deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

procedimento multiplicaValores(valor1,valor2:real);

var

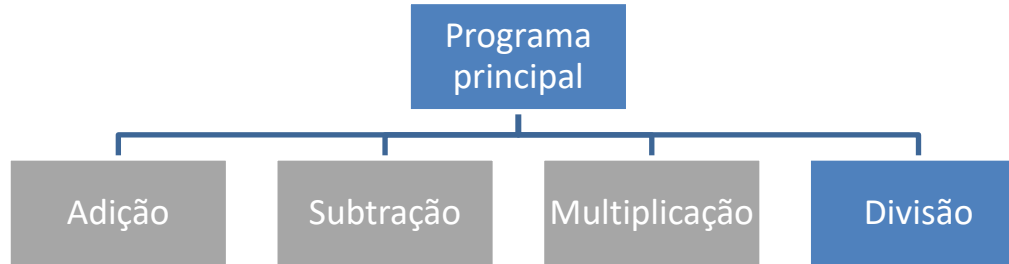
inicio

$r \leftarrow \text{valor1} * \text{valor2}$

fimprocedimento

```
73 void multiplicaValores(float valor1,float valor2)
74 {
75     r= valor1 * valor2;
76 }
```

# Calculadora - Divisão



deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

```
procedimento divideValores(a,b:real);  
  var  
  inicio  
     $r \leftarrow a / b$   
  fimprocedimento
```

```
77 void divideValores(float a, float b)  
78 {  
79     r = (float) a / b;  
80 }  
81
```



# Calculadora

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <locale.h>
4
5 void menu();
6 void somaValores(float n1, float n2);
7 void subtraiValores(float numero1, float numero2);
8 void multiplicaValores(float valor1, float valor2);
9 void divideValores(float a, float b);
10
11 float r;
```

```
54 void menu()
55 {
56     system("CLS");
57     printf("menu:\n");
58     printf("1 - adição\n");
59     printf("2 - subtração\n");
60     printf("3 - multiplicação\n");
61     printf("4 - divisão\n");
62     printf("5 - novos numeros\n");
63     printf("9 - sair do programa\n");
64 }
65 void somaValores(float n1, float n2)
66 {
67     r = n1 + n2;
68 }
69 void subtraiValores(float numero1, float numero2)
70 {
71     r = numero1 - numero2;
72 }
73 void multiplicaValores(float valor1, float valor2)
74 {
75     r = valor1 * valor2;
76 }
77 void divideValores(float a, float b)
78 {
79     r = (float) a / b;
80 }
81
```

```
12 int main()
13 {
14     setlocale(LC_ALL, "portuguese");
15     int op;
16     float a, b;
17     void leValores()
18     {
19         printf("Valores para calcular:\n");
20         printf("Digite o 1 valor:");
21         scanf("%f", &a);
22         printf("Digite o 2 valor:");
23         scanf("%f", &b);
24     }
25     leValores();
26     do
27     {
28         menu();
29         printf("Escolha:");
30         scanf("%d", &op);
31         switch (op)
32         {
33             case 1:
34                 somaValores(a, b);
35                 break;
36             case 2:
37                 subtraiValores(a, b);
38                 break;
39             case 3:
40                 multiplicaValores(a, b);
41                 break;
42             case 4:
43                 divideValores(a, b);
44                 break;
45             case 5:
46                 leValores();
47                 break;
48         }
49         if (op != 9 && op != 5)
50         {
51             printf("O resultado da operação = %.2f\n", r);
52             system("PAUSE");
53         }
54     } while (op != 9);
55     return 0;
56 }
```

# Exercício de fixação

Crie uma variável também global (Soma) do tipo inteiro.

Crie um procedimento utilizando passagem de parâmetro por valor e produza o somatório do número 1 até valor passado por parâmetro, e coloque o resultado na variável global Soma.

Crie um programa principal que solicite a entrada de um valor e chame o procedimento criado acima passando esse valor por parâmetro na modalidade valor e imprima o valor de Soma.

# Exercício de fixação

- 1 - Elabore um procedimento para calcular o Fatorial de um número passado por parâmetro. A resposta deverá ser retornada em uma variável definida como global de nome FAT. Faça também um programa principal que chame esse procedimento.
- 2 – Elabore um procedimento que receba um valor passado por parâmetro. Esse procedimento deverá devolver na variável global Fib o termo correspondente ao valor passado da serie de Fibonacci. Exemplo suponha que seja passado o valor 6, então deverá retorna o 6º termo da série ou seja 1,1,2,3,5,8, ou seja, deverá atribuir a variável Fib o valor 8.
- 3 – Elabore um procedimento que receba um valor passado por parâmetro. Esse procedimento deverá devolver na variável global Fib a soma dos termos da serie de Fibonacci do primeiro termo até o termo passado. Exemplo suponha que seja passado o valor 6, então deverá atribuir a variável Fib a soma de  $1+1+2+3+5+8$ , ou seja,  $Fib = 20$ .
- 4 – Elabore um procedimento que receba dois valores passados por parâmetro. Esse procedimento deverá atribuir a variável impar, a quantidade de números ímpares entre o intervalo fechado correspondente aos valores passados. Por exemplo, suponha que os valores passados seja, 3 e 6, deverá devolver  $impar=2$ , pois existem os números ímpares 3 e 5 no intervalo.
- 5 - Elabore um procedimento que receba dois valores passados por parâmetro. Esse procedimento deverá atribuir a variável impar, a soma dos números ímpares entre o intervalo fechado correspondente aos valores passados. Por exemplo, suponha que os valores passados seja, 3 e 6, deverá devolver  $impar=8$ , pois a soma dos números ímpares 3 e 5 no intervalo é 8.



**PUC Minas**  
**Virtual**