

Passagens de Parâmetros

Roberto Rocha



E se o parâmetro voltasse valor?

Passagem de parâmetro por referência

Qualquer modificação feita no conteúdo do parâmetro formal implica alteração imediata do conteúdo da variável correspondente.

A palavra VAR em frente ao nome do parâmetro indica que a passagem será por referência.

```
procedimento <nomeprocedimento> ([var] parâmetros)
var
  <declaração das variáveis locais ao procedimento>
inicio
  <lista de comandos>
fimprocedimento
```

Passagem de parâmetro por referência

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

```
procedimento sucessor(var numero:inteiro)
  var
  inicio
    numero ← numero +1
fimprocedimento
```

Algoritmo “ex01”

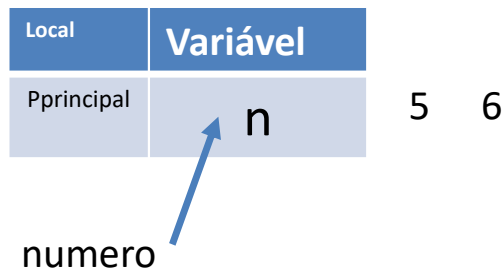
```
var n:inteiro
Inicio
  leia(n)
  sucessor(n)
  escreva(“O sucessor é = “,n)
finalgoritmo
```

Passagem de parâmetro por referência

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

→ procedimento sucessor(var numero:inteiro)
 var
 inicio
 → numero ← numero + 1
 → fimprocedimento

Algoritmo "ex02"
→ var n:inteiro
Início
→ leia(n)
→ sucessor(n)
→ escreva("O sucessor é = ",n)
→ fimalgoritmo



O sucessor é = 6

Passagem de parâmetro por referência

procedimento sucessor(var numero:inteiro)

var

inicio

numero \leftarrow numero +1

fimprocedimento

Algoritmo "ex02"

var n:inteiro

Inicio

leia(n)

sucessor(n)

escreva("O sucessor é = ",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     printf("O sucessor é = %d\n", n);
13     return 0;
14 }
15 void sucessor(int *numero)
16 {
17     *numero = *numero+1;
18 }
```

Chamada &

* Indicando
passagem por
referência

```
Digite um valor:5
O sucessor é = 6
Process returned 0 (0x0)   execution time : 24.019 s
Press any key to continue.
```

Passagem de parâmetro por referência

procedimento sucessor(var numero:inteiro)

var

inicio

numero \leftarrow numero +1

fimprocedimento

Algoritmo "ex02"

var n:inteiro

Inicio

leia(n)

sucessor(n)

escreva("O sucessor é = ",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     printf("O sucessor é = %d\n", n);
13     return 0;
14 }
15 void sucessor(int *numero)
16 {
17     *numero = *numero+1;
18 }
```

Chamada &

* Indicando
passagem por
referência

Exercício: Faça um procedimento que receba dois valores e devolva a soma desses valores

Exercício: Faça um procedimento que leia um conjunto de valores e devolva a média desses valores

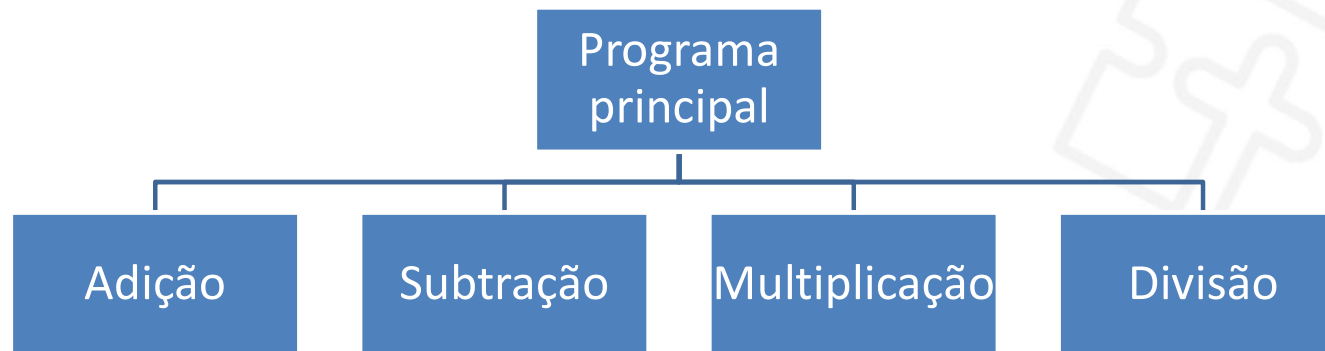
Procedimentos

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 e referência.

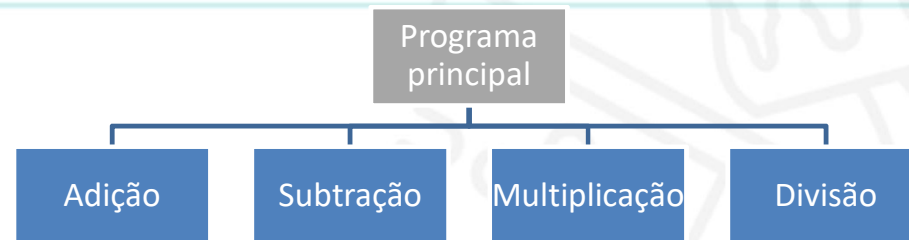
Calculadora



Calculadora

Algoritmo "Calculadora"

```
var
  op:inteiro
  a,b,r:real
Inicio
  faça
    leValores(a,b)
    menu(op)
    escolha (op)
      1: somaValores(a,b,r)
      2: subtraiValores(a,b,r)
      3: multiplicaValores(a,b,r)
      4:divideValores(a,b,r)
      5:leValores(a,b)
    fimescolha
  se (op<>9 e op<>5) entao
    escreva(r)
  fimse
  enquanto (op<>9)
finalgoritmo
```



procedimento menu(var item:inteiro)

```
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")
    leia(item)
  fimprocedimento
```

procedimento leValores(var a,b :real)

```
var
  inicio
    leia(a,b)
  fimprocedimento
```

Calculadora

Algoritmo "Calculadora"

var

op: inteiro

a,b,r: real

Início

faça

leValores(a,b)

menu(op)

escolha (op)

1: somaValores(a,b,r)

2: subtraiValores(a,b,r)

3: multiplicaValores(a,b,r)

4: divideValores(a,b,r)

5: leValores(a,b)

fimescolha

se (op<>9 e op<>5) então

escreva(r)

fimse

enquanto (op<>9)

finalgoritmo

```
12 int main()
13 {   setlocale(LC_ALL, "portuguese");
14     int op;
15     float a,b,r;
16     leValores(&a,&b);
17     do
18     {   menu(&op);
19         switch (op)
20         {case 1:
21             somaValores(a,b,&r);
22             break;
23         case 2:
24             subtraiValores(a,b,&r);
25             break;
26         case 3:
27             multiplicaValores(a,b,&r);
28             break;
29         case 4:
30             divideValores(a,b,&r);
31             break;
32         case 5:
33             leValores(&a,&b);
34             break;
35         }
36         if (op!=9 && op!=5)
37         {   printf("O resultado da operação = %5.2f\n",r);
38             system("PAUSE");
39         }
40     }
41     while(op!=9);
42     return 0;
43 }
```

Chamada por referência
Utilizar & na frente da
variável

Calculadora

```
procedimento leValores(var a,b :real)
var
  inicio
    leia(a,b)
  fimprocedimento
```

```
procedimento menu(var item:inteiro)
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")
    leia(item)
  fimprocedimento
```

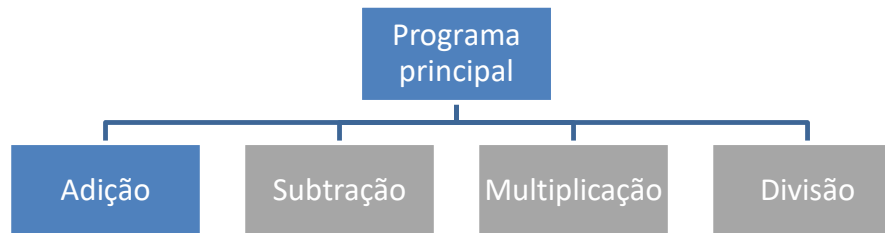
```
44 void leValores(int *a,int *b)
45 { printf("Valores para calcular:\n");
46   printf("Digite o 1 valor:");
47   scanf("%f",a);
48   printf("Digite o 2 valor:");
49   scanf("%f",b);
50 }
51 void menu(int *item)
52 {
53   system("CLS");
54   printf("menu:\n");
55   printf("1 - adição\n");
56   printf("2 - subtração\n");
57   printf("3 - multiplicação\n");
58   printf("4 - divisão\n");
59   printf("5 - novos numeros\n");
60   printf("9 - sair do programa\n");
61   printf("Escolha:");
62   scanf("%d",item);
63 }
```

O parâmetro já era
ponteiro!!!

passagem por referência
Utilizar * na frente da
variável

Por que aqui não
precisou utilizar & na
chamada do scanf?

Calculadora - adição

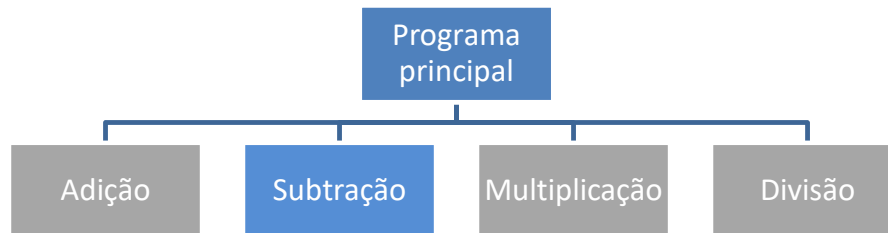


deve ser solicitada a entrada de dois valores numéricos para que seja possível executar o processamento escolhido e mostrar o resultado.

```
procedimento somaValores(n1,n2:real;var soma:real);  
  var  
  inicio  
    soma ← n1 + n2  
  fimprocedimento
```

```
64 void somaValores(float n1, float n2, float *soma)  
65 {  
66     *soma = n1 + n2;  
67 }
```

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 r real);
```

```
var
```

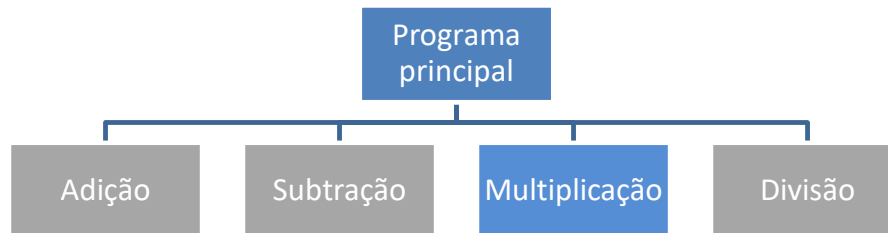
```
inicio
```

```
    r ← numero1 – numero2
```

```
fimprocedimento
```

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

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 multiplica real);

var

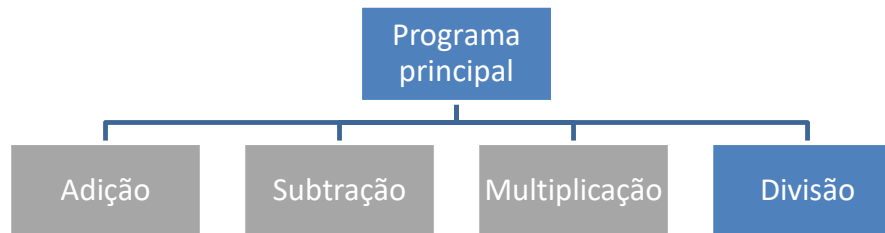
inicio

 multiplica \leftarrow valor1 * valor2

fimprocedimento

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

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 divide:real);

var

inicio

$r \leftarrow a / b$

fimprocedimento

```
76 void divideValores(float a, float b, float *divide)
77 {
78     *divide = (float) a / b;
79 }
80
```


Calculadora

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

```
44 void leValores(int *a, int *b)
45 {
46     printf("Valores para calcular:\n");
47     printf("Digite o 1 valor:");
48     scanf("%f", a);
49     printf("Digite o 2 valor:");
50     scanf("%f", b);
51 }
52 void menu(int *item)
53 {
54     system("CLS");
55     printf("menu:\n");
56     printf("1 - adição\n");
57     printf("2 - subtração\n");
58     printf("3 - multiplicação\n");
59     printf("4 - divisão\n");
60     printf("5 - novos numeros\n");
61     printf("9 - sair do programa\n");
62     printf("Escolha:");
63     scanf("%d", item);
64 }
```

```
64 void somaValores(float n1, float n2, float *soma)
65 {
66     *soma = n1 + n2;
67 }
68 void subtraiValores(float numero1, float numero2, float *r)
69 {
70     *r = numero1 - numero2;
71 }
72 void multiplicaValores(float valor1, float valor2, float *multiplica)
73 {
74     *multiplica = valor1 * valor2;
75 }
76 void divideValores(float a, float b, float *divide)
77 {
78     *divide = (float) a / b;
79 }
```

```
12 int main()
13 {
14     setlocale(LC_ALL, "portuguese");
15     int op;
16     float a, b, r;
17     leValores(&a, &b);
18     do
19     {
20         menu(&op);
21         switch (op)
22         {
23             case 1:
24                 somaValores(a, b, &r);
25                 break;
26             case 2:
27                 subtraiValores(a, b, &r);
28                 break;
29             case 3:
30                 multiplicaValores(a, b, &r);
31                 break;
32             case 4:
33                 divideValores(a, b, &r);
34                 break;
35             case 5:
36                 leValores(&a, &b);
37                 break;
38             case 9:
39                 return 0;
40         }
41     } while (op != 9);
42     return 0;
43 }
```

Exercício de fixação

Crie um procedimento utilizando passagem de parâmetro por valor e produza o somatório do número 1 até valor passado por parâmetro, devolva o resultado em um parâmetro por referência.

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 receba em outro parâmetro na modalidade referência e imprima o resultado esperado.

Exercício de fixação

Elabore um procedimento para calcular o Fatorial de um número passado por parâmetro por valor. A resposta deverá ser retornada em uma variável passada por parâmetro - referência. Faça também um programa principal que chame esse procedimento.

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 passada por referência.

Faça também um programa principal que chame esse procedimento.

2 – Elabore um procedimento que receba um valor passado por parâmetro. Esse procedimento deverá devolver em uma variável passada por referência 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 em uma variável passada por referência 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 devolver a soma de $1+1+2+3+5+8$, ou seja, 20.

4 – Elabore um procedimento que receba dois valores passados por parâmetro. Esse procedimento deverá em uma variável passada por referência, 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 2, pois existem os números ímpares 3 e 5 no intervalo.

5 - Elabore um procedimento que receba dois valores passados por parâmetro. Esse procedimento em uma variável passada por referência, 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 8, pois a soma dos números ímpares 3 e 5 no intervalo é 8.



PUC Minas
Virtual