

CAPÍTULO 2

PROCESSAMENTO DE PROGRAMAS: VISÃO GERAL CONSTRUÇÃO DE ALGORITMOS: ESTRUTURAS DE CONTROLE

1.1. INTRODUÇÃO

Neste capítulo discutiremos brevemente as estruturas de controle utilizadas na construção de algoritmos. Na seqüência mostraremos alguns exemplos de codificação de algoritmos em Pascal. Para finalizar o capítulo, apresentaremos o modo de se processar um programa Pascal utilizando o ambiente do Delphi 5, como uma aplicação do tipo Console (ou seja, não voltada para o ambiente Windows).

1.2. SOLUÇÃO DE PROBLEMAS: ALGORITMOS

1.2.1. ESTRUTURAS DE CONTROLE

a) estrutura sequencial

ação 1;
ação 2;
⋮
ação n;

Ex.: volume de uma esfera:

$$V = \frac{4}{3} \pi r^3$$

1. Leia o raio;
2. Calcule o volume;
3. Apresente o valor obtido;

b) estrutura de seleção

b.1) se condição

então
 seqüência de comandos
fim se

Ex.:

1. Leia o raio;
2. Se raio < 0
 então
 2.1 gere mensagem;
 2.2 pare;
 fim se;
3. Calcule o volume;
4. Apresente o valor obtido;
5. retorne ao passo 1;

Ação 2: desvio condicional do fluxo de controle.

Ação 5: desvio incondicional do fluxo de controle.

b.2) Se condição
então
sequencia de comandos 1
senão
sequencia de comandos 2
fim se

Obs.:

se raio < 0
ENDENTAÇÃO → $\left\{ \begin{array}{l} \text{gere mensagem;} \\ \text{pare;} \end{array} \right.$
fim se;

c) estrutura de repetição

c.1) repetição com variável contadora

para v = v_{inic} até v_{final} passo p
faça
sequência de comandos
fim para;

c.2) repetição com teste de condição

c.2.1) teste no início da repetição

Enquanto condição
faça
sequência de comandos
fim enquanto;

c.2.2) teste no final da repetição

repita
sequência de comandos
até que condição;

Ex.:

1. Leia o raio;
2. Se raio < 0
então
 - 2.1 gere mensagem;
 - 2.2 pare;
 - 2.3 calcule o volume;
 - 2.4 apresente o valor obtido;fim se;
3. retorne ao passo 1;

Ex.:

leia nc; (* número de círculos *)
para i = 1 até 1000 passo 1
leia o raio;
se raio < 0
então
gere mensagem;
pare;
fim-se;
calcule o volume;
apresente o valor obtido;
fim para;

Ex.:

leia o raio;
Enquanto raio > 0
faça
calcule o volume
apresente o valor obtido,
leia o raio;
fim enquanto;

Ex.:

leia o raio;
se raio < 0 então pare;
repita
calcule o volume;
apresente o valor obtido;
leia o raio;
até que raio < 0;

1.2.2. EXEMPLOS DE ALGORITMOS

Exemplo 1: A tabela abaixo traz as faixas de contribuição para cálculo do Imposto de Renda retido na fonte. Esta tabela também é utilizada para cálculo do imposto a pagar através do carnê Leão.

Imposto de Renda – Tabela progressiva mensal		
Base de Cálculo	Alíquota(%)	Parcela a deduzir
Até 900,00	Isento	-
Acima de 900,00 até 1800,00	15,0	135,00
Acima de 1800,00	27,5	360,00

Vamos construir um algoritmo para calcular o valor devido de imposto, para um dado valor de salário.

Algoritmo:

início;

Solicite o salário;

leia o salário;

se salário \leq 900

então

gere mensagem de isenção;

fim se;

se salario $>$ 900 e salario \leq 1800

então

 imposto = salario x 0.15 - 135;

imprima imposto;

fim se;

se salario $>$ 1800

então

 imposto = salario x 0.275 - 360

imprima imposto;

fim se;

pare;

fim;

Solução com "aninhamento" de estruturas:

Se salario \leq 900

então

gere mensagem de isenção;

pare;

senão /* salario $>$ 900 */

se salario \leq 1800

então

 imposto = salario x 0.15 - 135

senão /* salario $>$ 1800 */

$$\text{imposto} = \text{salario} \times 0.275 - 360$$

fim se

fim se;

Exemplo 2: seja obter uma solução algorítmica para calcular e imprimir os n primeiros termos seqüência

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots$$

$t_i = i\text{-ésimo termo}$

i	t_i	
1	$1 = \frac{1}{2^0} = \frac{1}{2^{1-1}}$	
2	$\frac{1}{2} = \frac{1}{2^1} = \frac{1}{2^{2-1}}$	$\therefore i\text{-ésimo termo:}$
4	$\frac{1}{4} = \frac{1}{2^2} = \frac{1}{2^{3-1}}$	$\frac{1}{2^{i-1}}$
\vdots	\vdots	

Soluções algorítmicas alternativas:

solicite n;

leia n;

para i de 1 até n passo 1

$$\text{tr} = \frac{1}{2^{i-1}}$$

imprima tr

fim para;

pare;

solicite n;

leia n;

i = 1;

Enquanto i ≤ n

faça

$$\text{tr} = \frac{1}{2^{i-1}}$$

imprima tr;

i = i + 1;

fim enquanto;

pare;

solicite n;

leia n;

se n ≤ 0

então pare;

i = 1;

repita

$$\text{tr} = \frac{1}{2^{i-1}};$$

imprima tr;

i = i + 1;

até que i > n;

pare;

1. solicite n;

2. leia n;

3. i = 1;

4. se i ≤ n

então

$$\text{tr} = \frac{1}{2^{i-1}};$$

imprima tr;

i = i + 1;

retorne ao passo 4;

senão

pare;

fim se;

1.2.3. COMPLEMENTO: CODIFICAÇÃO EM PASCAL DE UM ALGORITMO PARA CALCULAR O MONTANTE PRODUZIDO POR UM CAPITAL

Expressão de Cálculo:

$$M = C(1 + I)^N$$

onde: C é o capital
 I é a taxa unitária, e
 N é o número de períodos da aplicação.

Codificação:

```
program montante (input, output);
(* Programa p/ calculo do Montante *)
(* produzido por um capital      *)
uses
  WinCrt;
var M,C,I: real;
var N : integer;
begin (* programa principal *)

  (* entrada de dados*)
  ClrScr;    (* limpa a tela *)
  writeln (' Entre com o capital = ');
  readln (c);
  writeln ('Entre com a taxa = ');
  readln (I);
  writeln (' Entre com o número de períodos =');
  readln (N);

  (* calculo do montante *)
  M := exp(ln (c) + N * ln (1 + I));

  (* saída montante *)
  writeln (' Montante Calculado = ', M : 7 : 2);
  readln;
end. (* Programa principal *)
```

Modificações:

(i) Uso do construtor de repetição *while*:

```
program montante (input, output);
uses
  WinCrt;
var M,C,I: real;
var N : integer;
begin (* programa principal *)
```

```

(* entrada de dados *)
ClrScr;    (* limpa a tela *)
    (* entrada do primeiro valor de capital *)
writeln ('Entre com o capital (<= 0 para terminar) = ');
readln (c);
while (c > 0) do
begin
    (* demais dados *)
    writeln ('Entre com a taxa = ');
    readln (I);
    writeln (' Entre com o número de períodos =');
    readln (N);

    (* calculo do montante *)
    M: = exp (ln (c) + N * ln (1 + I));

    (* saída montante *)
    writeln (' Montante Calculado = ', M : 7 : 2);
    readln;

    (* proximo capital *)
    writeln ('Entre com o capital (<= 0 para terminar) = ');
    readln (c);
end;
readln;
end.

```

(ii) Uso do construtor de repetição ***repeat until***:

```

program montante (input, output);
(* programa para calculo do montante *)
uses
    WinCrt;
var M,C,I: real;
var N : integer;

begin (* programa principal *)

    (* entrada do primeiro valor de capital *)
    ClrScr;    (* limpa a tela *)
    writeln ('Entre com o capital (<= 0 para terminar) = ');
    readln (c);
    (* teste primeiro valor negativo *)
    if (c <= 0)
    then begin
        writeln ('Termino Execução');
        readln;
        exit;
    end;

```

```

repeat
    (* demais dados *)
    writeln ('Entre com a taxa = ');
    readln (I);
    writeln (' Entre com o número de períodos =');
    readln (N);

    (* calculo do montante *)
    M: = exp (ln (c) + N * ln (1 + I));

    (* saída montante *)
    writeln (' Montante Calculado = ', M : 7 : 2);
    readln;

    (* proximo capital *)
    writeln ('Entre com o capital (<= 0 para terminar) = ');
    readln (c);
until (c <= 0);

readln;
end.

```

(iii) Uso de uma estrutura do tipo *for*:

```

program montante (input, output);
uses
    WinCrt;
var M,C,I : real;
var N, J, NCalc : integer;

begin (* programa principal *)

    (* entrada do número de calculos a efetuar *)
    (* controle de reexecução *)
    ClrScr; (* limpa a tela *)
    writeln (' Quantos Calculos?');
    readln (NCalc);
    for j : = 1 to NCalc do
    begin

        (* entrada de dados*)
        ClrScr; (* limpa a tela *)
        writeln (' Entre com o capital = ');
        readln (c);
        writeln ('Entre com a taxa = ');
        readln (I);
        writeln (' Entre com o número de períodos =');
        readln (N);
    end;
end;

```

```

(* calculo do montante *)
M: = exp (ln (c) + N * ln (1 + I));

(* saída montante *)
writeln (' Montante Calculado = ', M : 7 : 2);
readln;
end;
readln;
end.

```

1.3. PROCESSAMENTO DE UM PROGRAMA PASCAL UTILIZANDO O DELPHI 5

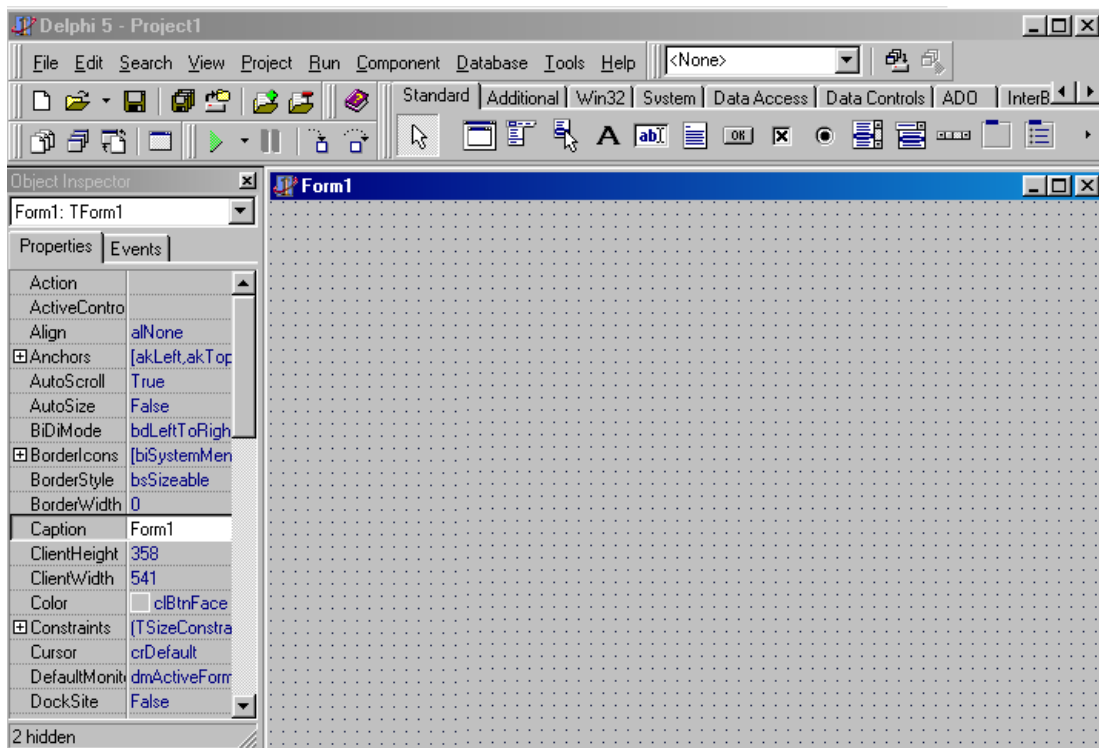
Mostraremos nesta seção como proceder para processar o exemplo de cálculo do montante produzido por um capital.

Passo 1:

Iniciar a execução do ambiente do Delphi 5:

Opções de menu: Iniciar/ Programas/ Borland Delphi 5/ Delphi 5.

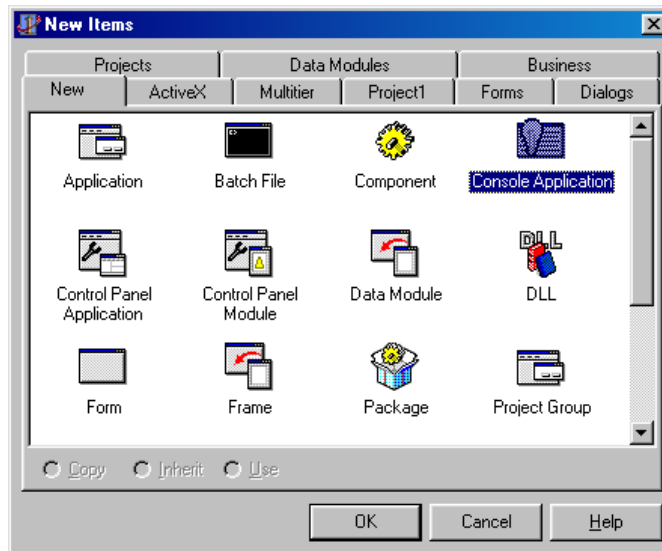
O ambiente será como mostrado na figura a seguir:



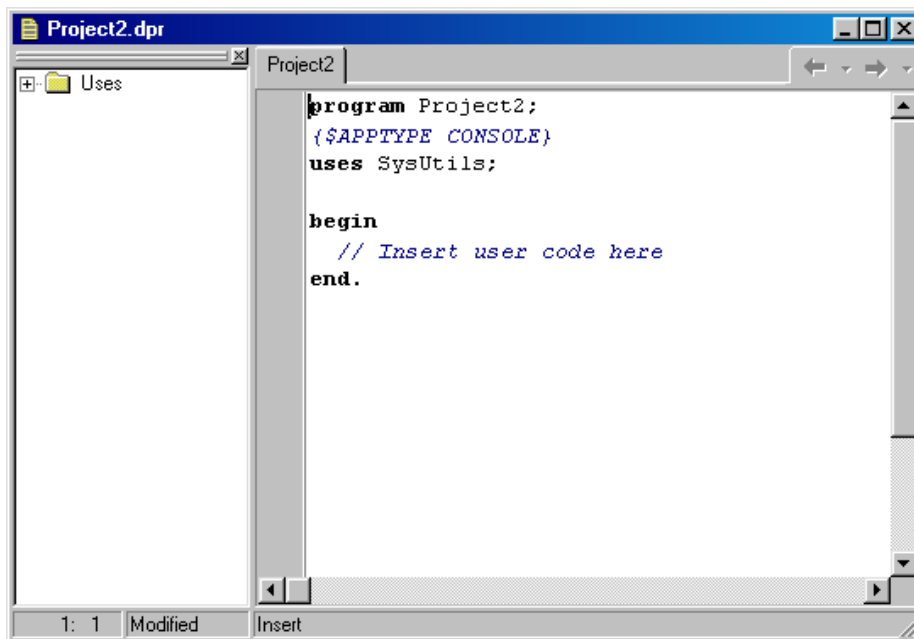
Passo 2:

Criar uma aplicação em Delphi do tipo *Console*:

A partir do menu *File*: opção *New*; selecionar *Console Application*, como mostrado na figura a seguir:



O aspecto da janela de edição será como mostrado na figura a seguir:



Passo 3:

Digitar o programa segundo o esquema:

Após `uses SysUtils`: digitar a parte do programa relativa à declaração de variáveis:

```
var M,C,I: real;  
var N : integer;
```

Na parte da janela de edição em que se lê:

`// Insert user code here`

digitar o código do programa como mostrado abaixo:

```

(* entrada de dados*)
writeln ('Entre com o capital: ');
readln (c);
writeln ('Entre com a taxa: ');
readln (I);
writeln ('Entre com o numero de periodos: ');
readln (N);

(* calculo do montante *)
M := exp(ln (c) + N * ln (1 + I));

(* saída montante *)
writeln (' Montante Calculado = ', M : 7 : 2);
readln;

```

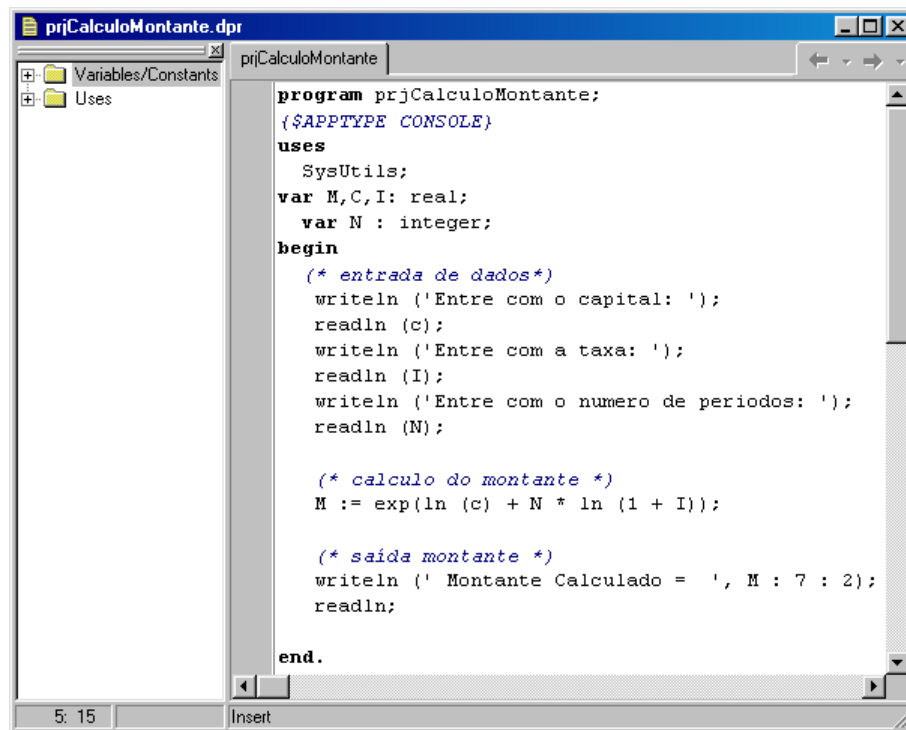
Passo 4:

Com o auxílio do Windows Explorer, criar uma pasta de nome *projeto_Calculo_de_Montante*.

Salvar o projeto nesta pasta:

Opções de menu: File/Save All; salvar o projeto (Project2) sob o nome: *prjCalculoMontante*.

A figura a seguir mostra como fica a janela de edição após as etapas anteriores:

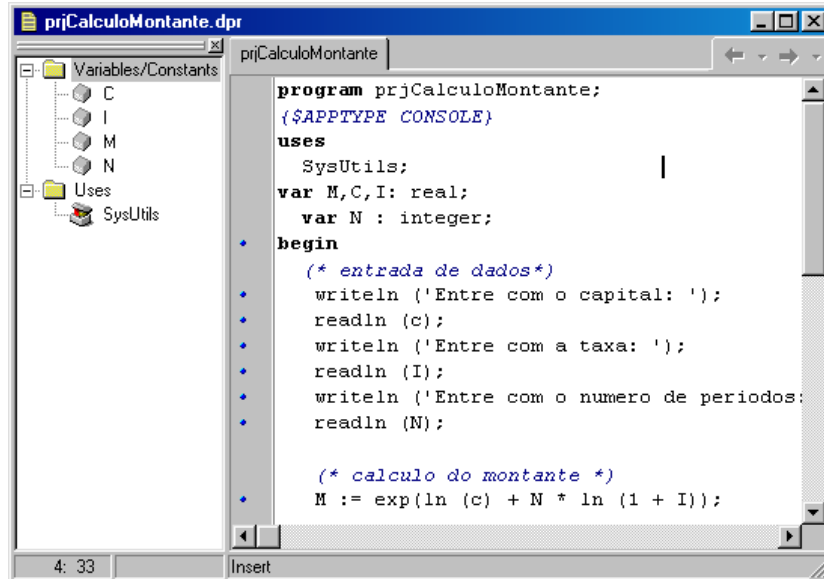


Passo 5:

Compilar o programa:

Opções de menu: Project/Compile prjCalculoMontante.

Após a compilação, caso não haja nenhum erro na codificação do programa, a janela terá o aspecto mostrado na figura a seguir. Notar a presença de um “marcador”, um pequeno círculo azul, à esquerda de cada linha de comando. Em caso de erro, corrigir o erro, salvar o projeto e recompilar novamente.



Passo 6:

Após a compilação, efetuar a link-edição:

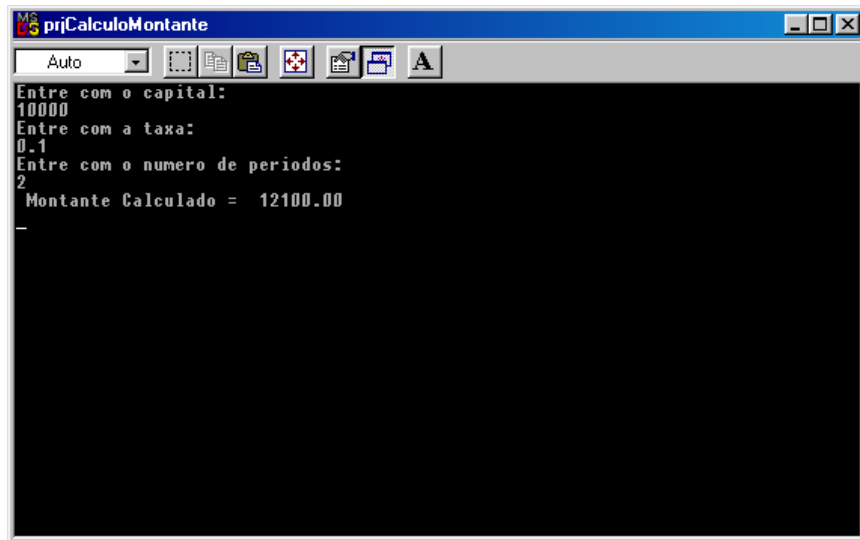
Menu: Project, opção: Build prjCalculoMontante.

Passo 7:

Após a link-edição, executar o projeto:

Menu: Run, opção Run (ou F9).

A figura a seguir mostra uma tela de execução para este projeto.



Observar o conteúdo da pasta *projeto_Calculo_de_Montante* e anotar na tabela abaixo o nome dos arquivos lá encontrados. Assinalar o arquivo correspondente ao projeto bem como o arquivo que corresponde ao executável gerado pelo Delphi após a compilação do projeto.

Arquivo	Função

Para evitar conflitos e por questões de organização, é sempre aconselhável criar uma nova pasta para cada novo projeto a ser desenvolvido no Delphi.

REFERÊNCIAS

1. Tanenbaum, A.S. **Structured Computer Organization**. Prentice-Hall, 1984.
2. Farrer, H. e outros. **Algoritmos Estruturados**. Guanabara Dois, 1985.
3. Schmitz, E.A. e Souza Teles, A.A. **Pascal e Técnicas de Programação**. Livros Técnicos e Científicos Editora, 1988.
4. Cantù, Marco **Dominando o Delphi 2**. Makron Books, 1996.