

Capítulo 3 – Estruturas de Controle

Estruturas básicas de controle do fluxo de execução:

- Estruturas de sequência
- Estruturas de decisão
- Estruturas de repetição

Estruturas de sequência

Em uma estrutura de sequência, as ações são separadas por ponto-e-vírgula (;) e executadas de cima para baixo e da esquerda para a direita.

Cuidado! Quanto à ordem de execução deve-se atentar à precedência de operadores e a peculiaridades de algumas linguagens de programação (JavaScript).

Bloco de código:

```
início  
  // declaração de variáveis  
  ação-1;  
  ação-2;  
  ...  
fim.
```

Exemplo: Algoritmo para calcular a média aritmética das quatro notas de um aluno.

```
início  
  real: n1, n2, n3, n4, // notas bimestrais  
           média; // média anual  
  leia(n1, n2, n3, n4);  
  média ← (n1 + n2 + n3 + n4) / 4;  
  escreva(média);  
fim.
```

Exemplo: Algoritmo para calcular o volume de uma esfera.

$$V = \frac{4}{3} \cdot \pi R^3$$

```
início  
  real: R, V; // raio e volume da esfera  
  real: PI = 3.14159; // constante PI  
  leia(R);  
  V ← 4 / 3 * PI * pot(R, 3);  
  escreva(V);  
fim.
```

Estruturas de decisão

Uma estrutura de decisão permite que uma ou mais ações sejam executadas com base no resultado (V ou F) de uma expressão lógica.

Estrutura de decisão simples:

```
se (expressão-lógica) então  
    ação-1;  
fim-se;
```

```
se (expressão-lógica) então  
    início  
        ação-1;  
        ação-2;  
        ...  
    fim;  
fim-se;
```

Estrutura de decisão composta:

```
se (expressão-lógica) então  
    ação-1;  
senão  
    ação-2;  
fim-se;
```

```
se (expressão-lógica) então  
    início  
        ação-1.1;  
        ação-1.2;  
        ...  
    fim;  
senão  
    início  
        ação-2.1;  
        ação-2.2;  
        ...  
    fim;  
fim-se;
```

```
se (expressão-lógica) então  
    ação-1;  
senão  
    início  
        ação-2.1;  
        ação-2.2;  
        ...  
    fim;  
fim-se;
```

```
se (expressão-lógica) então  
    início  
        ação-1.1;  
        ação-1.2;  
        ...  
    fim;  
senão  
    ação-2;  
fim-se;
```

Exemplo: Algoritmo para calcular a média aritmética das quatro notas de um aluno e informar se o aluno foi aprovado.

```
início
  real: n1, n2, n3, n4, // notas bimestrais
           média; // média anual
  leia(n1, n2, n3, n4);
  média ← (n1 + n2 + n3 + n4) / 4;
  escreva(média);
  se (média >= 7.0) então
    escreva("Aluno aprovado");
  fim-se;
fim.
```

Exemplo: Algoritmo para calcular a média aritmética das quatro notas de um aluno e informar se o aluno foi aprovado ou reprovado.

```
início
  real: n1, n2, n3, n4, // notas bimestrais
           média; // média anual
  leia(n1, n2, n3, n4);
  média ← (n1 + n2 + n3 + n4) / 4;
  escreva(média);
  se (média >= 7.0) então
    escreva("Aluno aprovado");
  senão
    escreva("Aluno reprovado");
  fim-se;
fim.
```

Exemplo: Algoritmo para calcular as raízes de uma equação do segundo grau.

$$ax^2 + bx + c = 0 \quad \Delta = b^2 - 4ac$$
$$\Delta < 0 \Rightarrow \nexists \text{ raiz real} \quad \Delta = 0 \Rightarrow x = \frac{-b}{2a} \quad \Delta > 0 \Rightarrow x_1 = \frac{-b + \sqrt{\Delta}}{2a} ; x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

```
início
  real: a, b, c; // coeficientes da equação do segundo grau
  real: delta;
  real: x, x1, x2; // raízes da equação
  leia(a, b, c);
  delta ← b * b - 4 * a * c;
  se (delta < 0) então
    escreva("Não existe raiz real");
  senão
    se (delta = 0) então
      início
        x ← -b / (2 * a);
        escreva(x);
      fim;
    senão
      início
        x1 ← (-b + rad(delta)) / (2 * a);
        x2 ← (-b - rad(delta)) / (2 * a);
        escreva(x1, x2);
      fim;
    fim-se;
  fim-se;
fim.
```

Considere agora o seguinte exemplo...

```
início
caractere: operação;
leia(operação);
se (operação = "+") então
    escreva("Adição");
senão
    se (operação = "-") então
        escreva("Subtração");
    senão
        se (operação = "*") então
            escreva("Multiplicação");
        senão
            se (operação = "/") então
                escreva("Divisão");
            senão
                escreva("Operação matemática inválida");
        fim-se;
    fim-se;
fim-se;
fim-se;
fim-se;
fim-se;
fim-se;
fim.
```

Estrutura de decisão de múltipla escolha:

```
escolha variável
caso valor-1:
    ação-1;
caso valor-2:
    início
        ação-2.1;
        ação-2.2;
        ...
    fim;
caso m..n: // intervalo numérico
    ação-3;
caso i, j, k:
    ação-4;
caso contrário:
    ação-5;
fim-escolha;
```

Exemplo: Utilizando a estrutura de decisão de múltipla escolha para o problema anterior, temos...

```
início
caractere: operação;
leia(operação);
escolha operação
    caso "+": escreva("Adição");
    caso "-": escreva("Subtração");
    caso "*": escreva("Multiplicação");
    caso "/": escreva("Divisão");
    caso contrário: escreva("Operação matemática inválida");
fim-escolha;
fim.
```

Estruturas de repetição

Uma estrutura de repetição permite que uma ou mais ações sejam executadas mais de uma vez. Blocos de código são implicitamente criados por estruturas de repetição.

Estrutura de repetição com teste no início:

```
enquanto (expressão-lógica) faça  
    ação-1;  
    ação-2;  
    ...  
fim-enquanto;
```

Exemplo: Algoritmo para somar dez números.

```
início  
    real: soma, número;  
    inteiro: i;  
    soma ← 0;  
    i ← 0;  
    enquanto (i < 10) faça  
        leia(número);  
        soma ← soma + número;  
        i ← i + 1;  
    fim-enquanto;  
    escreva(soma);  
fim.
```

Estrutura de repetição com teste no final:

```
repita  
    ação-1;  
    ação-2;  
    ...  
até (expressão-lógica);
```

Exemplo: Algoritmo para calcular a média aritmética das quatro notas de um aluno e continuar a execução até que o usuário escolha sair.

```
início  
    real: n1, n2, n3, n4, média;  
    caractere: sair;  
    repita  
        leia(n1, n2, n3, n4);  
        média ← (n1 + n2 + n3 + n4) / 4;  
        escreva(média);  
        leia(sair);  
    até (sair = "sim");  
fim.
```

Exemplo: Algoritmo para tentar adivinhar um número que um colega imaginou e digitou como entrada.

```
início  
    inteiro: número, chute, tentativas;  
    leia(número);  
    tentativas ← 0;  
    repita  
        leia(chute);  
        tentativas ← tentativas + 1;  
        se (chute > número) então  
            escreva("Chutou alto!");  
        senão  
            se (chute < número) então  
                escreva("Chutou baixo!");  
            fim-se;  
        fim-se;  
    até (chute = número);  
    escreva(tentativas);  
fim.
```

Estrutura de repetição com variável de controle:

```
para variável de valor-inicial até valor-final passo p faça  
    ação-1;  
    ação-2;  
    ...  
fim-para;
```

Exemplo: Algoritmo para mostrar os números pares de 0 a 10.

```
início  
    inteiro: i;  
    para i de 0 até 10 passo 2 faça  
        escreva(i);  
    fim-para;  
fim.
```

Tarefa: Ler capítulo 4 do livro para a próxima aula

Desafio! Escreva um algoritmo para calcular a média aritmética das quatro notas de um aluno e informar o conceito final obtido com base na tabela a seguir.

A = 9.0 até 10

B = 8.0 até abaixo de 9.0

C = 7.0 até abaixo de 8.0

F = Abaixo de 7.0

Desafio! Escreva um algoritmo para ler três números e mostrar o menor deles.

Desafio! Escreva um algoritmo para ler três números e mostrá-los em ordem crescente.