

# Lógica de Programação

## Capítulo 3

### Estruturas de Controle

Lógica de Programação – Forbellone / Eberspacher

# Estruturas de Controle

- ✦ Neste capítulo
  - ✦ Estrutura Seqüencial
  - ✦ Estrutura de Seleção
  - ✦ Estrutura de Repetição

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

# Estrutura Seqüencial

- ✦ O Fluxo de Controle segue a mesma seqüência linear da nossa escrita, ou seja:
  - ✦ De cima para baixo;
  - ✦ Da esquerda para direita
- ✦ Cada ação é seguida de um ;
  - ✦ Objetiva separar uma ação da outra
  - ✦ Indica que a próxima ação da seqüência deve ser executada

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

# Estrutura Seqüencial

Algoritmo 3.2 - Média Aritmética

```
início
// declaração de variáveis
real: N1, N2, N3, N4, // notas bimestrais
      MA; // média anual
// entrada de dados
leia (N1, N2, N3, N4);
// processamento
MA ← (N1 + N2 + N3 + N4) / 4;
// saída de dados
escreva (MA);
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

# Estruturas de Seleção

- ✦ São aquelas que permitem alterar o Fluxo de Execução, de forma a selecionar qual parte deve ser executada
- ✦ Essa “decisão” de execução é tomada a partir de uma condição, que pode resultar apenas em Verdade ou Falsidade
- ✦ Uma condição é representada por expressões relacionais ou lógicas
- ✦ As estruturas de seleção podem ser classificadas em simples, compostas ou encadeadas

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

# Seleção Simples

```
se <condição> então
início // início do bloco verdade
comando 1;
comando 2;
...
comando n;
fim; // fim do bloco verdade
fimse;
```

- ✦ Quando a <condição> for verdadeira o “bloco verdade” é executado
- ✦ Quando a <condição> for falsa o “bloco verdade” não é executado

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Simples

Algoritmo 3.4 - Média Aritmética com Aprovação

```
início
// declaração de variáveis
real: N1, N2, N3, N4, // notas bimestrais
MA; // média anual
// entrada de dados
leia (N1, N2, N3, N4);
// processamento
MA ← (N1 + N2 + N3 + N4) / 4;
// saída de dados
escreva (MA);
se (MA >= 7) então
    escreva ("Aluno Aprovado!");
fimse;
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Composta

```
se <condição> então
início // início do bloco verdade
    comando 1;
    comando n;
fim; // fim do bloco verdade
senão
início // início do bloco falsidade
    comando 1;
    comando n;
fim; // fim do bloco falsidade
fimse;
```

- ✦ Quando a <condição> for verdadeira o "bloco verdade" é executado
- ✦ Quando a <condição> for falsa o "bloco falsidade" é executado

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Composta

Algoritmo 3.5 - Média Aritmética com aprovação e reprovação

```
início
// declaração de variáveis
real: N1, N2, N3, N4, // notas bimestrais
MA; // média anual
leia (N1, N2, N3, N4);
MA ← (N1 + N2 + N3 + N4) / 4;
escreva (MA);
se (MA >= 7) então
início
    escreva ("Aluno Aprovado!");
    escreva ("Parabéns!");
fim;
senão
início
    escreva ("Aluno Reprovado!");
    escreva ("Estude mais!");
fim;
fimse;
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Encadeada

- ✦ Ocorre quando uma seleção tem como ação uma outra seleção
- ✦ Uma seleção encadeada pode ser:
  - ✦ Heterogênea: Quando não é possível identificar padrão de comportamento
  - ✦ Homogênea: Quando é possível identificar padrão de comportamento
    - se – então – se: quando depois de cada **então** ocorre outro **se**
    - se – senão – se: quando depois de cada **senão** ocorre outro **se**

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Encadeada Heterogênea

Algoritmo 3.6 – Tipos de Triângulo

```
início
inteiro: A, B, C; // tamanho dos lados
leia (A, B, C);
se (A<B+C) e (B<A+C) e (C<A+B) então
se (A=B) e (B=C) então
    escreva ("Triângulo Equilátero");
senão
se (A=B) ou (B=C) ou (A=C) então
    escreva ("Triângulo Isósceles");
senão
    escreva ("Triângulo Escaleno");
fimse;
fimse;
senão
    escreva ("Estes valores não formam um triângulo");
fimse;
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Encadeada Homogênea

```
se – então – se
se <Cond1> então
se <Cond2> então
se <Cond3> então
se <Cond4> então W;
fimse;
fimse;
fimse;
fimse;
```

Cond1	Cond2	Cond3	Cond4	Ação
V	V	V	V	W

É equivalente a:  
se <Cond1> e <Cond2> e <Cond3> e <Cond4> então W;  
fimse;

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção Encadeada Homogênea

```

se X=V1 então
  C1;
fimse;
se X=V2 então
  C2;
fimse;
se X=V3 então
  C3;
fimse;
se X=V4 então
  C4;
fimse;

se X=V1
  então C1;
senão se X=V2
  então C2;
senão se X=V3
  então C3;
senão se X=V4
  então C4;
fimse;

```

**se – senão – se**

X=V1	X=V2	X=V3	X=V4	Ação
V	F	F	F	C1
F	V	F	F	C2
F	F	V	F	C3
F	F	F	V	C4

X=V1	X=V2	X=V3	X=V4	Ação
V	-	-	-	C1
F	V	-	-	C2
F	F	V	-	C3
F	F	F	V	C4

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção de Múltipla Escolha

- Seleções encadeadas homogêneas se-senão-se são bastante frequentes para o tratamento de listas de valor
- Para simplificar a escrita, pode-se utilizar o comando escolha.
- Adaptando o algoritmo anterior:

```

escolha X
caso V1: C1;
caso V2: C2;
caso V3: C3;
caso V4: C4;
fimescolha;

```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Seleção de Múltipla Escolha

Algoritmo 3.7 – Múltipla Escolha

```

início
real: Preço;
inteiro: Origem;
leia (Preço, Origem);
escolha Origem
caso 1: escreva (Preço, " – produto do Sul");
caso 2: escreva (Preço, " – produto do Norte");
caso 3: escreva (Preço, " – produto do Leste");
caso 4: escreva (Preço, " – produto do Oeste");
caso 7, 8, 9: escreva (Preço, " – produto do Sudeste");
caso 10..20: escreva (Preço, " – produto do Centro-Oeste");
caso 5, 6, 25..50: escreva (Preço, " – produto do Nordeste");
caso contrário: escreva (Preço, " – produto importado");
fimescolha;
fim.

```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Estruturas de Repetição

- São aquelas que permitem executar mais de uma vez (repetir) um determinado trecho do algoritmo
- O trecho do algoritmo em repetição é também chamado de laço (ou "loop")
- As repetições devem ser sempre finitas
- Quanto a quantidade de repetições, os laços podem ser
  - Pré-determinados: Sabe-se antes a quantidade de execuções
  - Indeterminados: Não se conhece a quantidade de execuções
- Quanto ao critério de parada, os laços podem utilizar
  - Teste no início
  - Teste no final
  - Variável de controle

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Início

- Laço que verifica antes de cada execução, se é "permitido" executar o trecho do algoritmo
- Trata-se de um laço que se mantém repetindo enquanto uma dada condição permanecer verdadeira

```

enquanto <condição> faça
  comando 1;
  comando 2;
  ...
  comando n;
fimenquanto;

```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Início

- Contador:** Variável que reproduz o processo de contagem

```

início
inteiro: CON;
CON ← 0;
enquanto CON < 3 faça
  CON ← CON + 1;
fimenquanto;
fim.

```

CON

3

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Início

Algoritmo 3.8 - Média Aritmética para 50 alunos

```
início
// declaração de variáveis
real: N1, N2, N3, N4, // notas bimestrais
      MA; // média anual
inteiro: CON; // contador
CON ← 0; // inicialização do contador
enquanto (CON < 50) faça // teste da condição de parada
  leia (N1, N2, N3, N4);
  MA ← (N1 + N2 + N3 + N4) / 4;
  escreva (MA);
  se (MA >= 7) então
    escreva ("Aluno Aprovado. Parabéns!");
  senão
    escreva ("Aluno Reprovado. Estude mais!");
  fimse;
  CON ← CON + 1; // incremento do contador
fimenquanto;
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Início

➤ **Acumulador:** Variável que reproduz o processo de acumulação

```
início
inteiro: CON, X, ACM;
CON ← 0;
ACM ← 0;
enquanto CON < 3 faça
  CON ← CON + 1;
  leia (X);
  ACM ← ACM + X;
fimenquanto;
fim.
```

CON	ACM	X
3	11	4

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Início

Algoritmo 3.9 - Média Aritmética da turma de 50 alunos

```
início
// declaração de variáveis
real: MA, // média anual de dado aluno
      ACM, // Acumulador
      MAT; // Média Anual da Turma
inteiro: CON; // contador
CON ← 0; // inicialização do contador
ACM ← 0; // inicialização do acumulador
enquanto (CON < 50) faça // teste da condição de parada
  leia (MA);
  ACM ← ACM + MA; // soma em ACM os valores lidos em MA
  CON ← CON + 1; // incremento do contador
fimenquanto;
MAT ← ACM / 50; // calculo da média anual da turma
escreva ("média anual da turma = ", MAT);
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Final

- Laço que verifica depois de cada execução, se é "permitido" continuar executando o trecho do algoritmo
- Trata-se de um laço que se mantém repetindo até que uma dada condição se torne verdadeira

```
repita
  comando 1;
  comando 2;
  ...
  comando n;
até <condição>;
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Final

Algoritmo 3.12 - Média Aritmética da turma com Repita

```
início
// declaração de variáveis
real: MA, // média anual de dado aluno
      ACM, // Acumulador
      MAT; // Média Anual da Turma
inteiro: CON; // contador
CON ← 0; // inicialização do contador
ACM ← 0; // inicialização do acumulador
repita
  leia (MA);
  ACM ← ACM + MA; // soma em ACM os valores lidos em MA
  CON ← CON + 1; // incremento do contador
até (CON >= 50); // teste da condição de parada
MAT ← ACM / 50; // calculo da média anual da turma
escreva ("média anual da turma = ", MAT);
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Variável de Controle

- Laço simplificado para utilização em repetições de quantidade predeterminada
- Incorpora internamente o funcionamento de um contador de repetições

```
para V de v1 até v2 passo p faça
  comando 1;
  comando 2;
  ...
  comando n;
fimpara;
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Repetição com Teste no Final

Algoritmo 3.12 - Média Aritmética da turma com Para

```
início
// declaração de variáveis
real: MA, // média anual de dado aluno
      ACM, // Acumulador
      MAT; // Média Anual da Turma
inteiro: V; // contador
ACM ← 0; // inicialização do acumulador
para V de 1 até 50 passo 1 faça
  leia (MA);
  ACM ← ACM + MA; // soma em ACM os valores lidos em MA
fimpara;
MAT ← ACM / 50; // calculo da média anual da turma
escreva ("média anual da turma = ", MAT);
fim.
```

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3

## Comparação entre Estruturas de Repetição

- Aprendemos 3 maneiras de construir laços de repetição
- É importante perceber que existem laços mais adequados ou convenientes para cada situação

Estrutura	Condição	Quantidade de Execuções	Condição de Existência
Enquanto	Início	zero ou muitas	Condição verdadeira
Repita	Final	uma ou muitas	Condição falsa
Para	Não tem	$((vf - vi) \text{ div } \text{passo}) + 1$	$vi \leq vf$

Lógica de Programação – Forbellone / Eberspacher – Capítulo 3