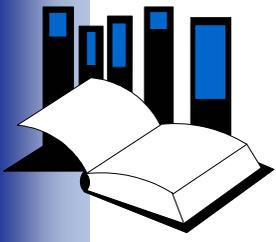




## Estruturas de Decisão



- Nesta aula são introduzidos conceitos básicos sobre estruturas de controle não seqüencial:
  - Seleções
  - Laços (*loops*)

José Augusto Baranauskas  
 Departamento de Física e Matemática – FFCLRP-USP  
 Sala 226 – Bloco P2 – Fone (16) 3602-4361

E-mail: [augusto@ffclrp.usp.br](mailto:augusto@ffclrp.usp.br)  
 URL: <http://fmp.usp.br/augusto>

## Estruturas de Decisão

- Condições
- Operadores Relacionais
- Operadores Lógicos
- Seleção
- Laços (repetição)
  - Condicionais
  - Contados

## Condição

- Uma condição é uma expressão na qual o resultado é booleano (ou lógico), ou seja, seu resultado é **verdadeiro** (*true*) ou **falso** (*false*)
- Exemplos
  - $3 > 2$ : resulta em **verdadeiro**
  - $2 < 3$ : resulta em **verdadeiro**
  - $3 > 7$ : resulta em **falso**
  - $7 < 3$ : resulta em **falso**
  - $3 = 3$ : resulta em **verdadeiro**
  - $3 \leq 3$ : resulta em **verdadeiro**
  - $3 \geq 3$ : resulta em **verdadeiro**
  - $3 \neq 3$ : resulta em **falso**

## Operadores Relacionais

- Um conjunto especial de operadores é necessário para expressar condições: operadores relacionais

Operador	Significado
>	maior que
<	menor que
≥	maior ou igual a
≤	menor ou igual a
≠	diferente de
=	igual a

## Operadores Relacionais em C/C++

Operador	Significado
>	maior que
<	menor que
>=	maior ou igual a
<=	menor ou igual a
!=	diferente de
==	igual a

## Operadores Relacionais em C/C++

Operador	Significado
>	maior que
<	menor que
>=	maior ou igual a
<=	menor ou igual a
!=	diferente de
==	igual a

**Atenção em C/C++:**  
 = é o operador de atribuição  
 == é o operador relacional de igualdade  
 É um erro de programação utilizar  
 "=" quando se deve utilizar "=="  
 (e vice-versa)

## Operadores Lógicos

- ❑ Condições simples podem ser combinadas para formar condições complexas
- ❑ Exemplos
  - $3 > 2$  e  $7 > 3$ : resulta em **verdadeiro**
  - $3 < 2$  e  $7 > 3$ : resulta em **falso**
  - $3 < 2$  ou  $7 > 3$ : resulta em **verdadeiro**
  - $\text{não}(3 < 2 \text{ e } 7 > 3)$ : resulta em **verdadeiro**

7

## Operadores Lógicos

- ❑ Um conjunto especial de operadores é necessário para combinar condições simples criando condições complexas: operadores lógicos

Operador	Significado
não	negação
e	conjunção
ou	disjunção

8

## Tabela Verdade dos Operadores Lógicos

- ❑ Sejam A e B duas variáveis lógicas, que assumem valores
  - **verdadeiro (V)** ou
  - **falso (F)**

A	não A
V	F
F	V

A	B	A e B
V	V	V
V	F	F
F	V	F
F	F	F

A	B	A ou B
V	V	V
V	F	V
F	V	V
F	F	F

9

## Qual o resultado?

- ❑ Assuma  $A = 2$ ;  $B = 3$ ;  $C = 10$
- ❑ não ( $A > B$ )
- ❑ não ( $B < C$ )
- ❑  $A < B$  e  $C > B$
- ❑  $A < B$  e  $C < B$
- ❑  $A > B$  e  $C > B$
- ❑  $A > B$  e  $C < B$
- ❑  $A < B$  e  $C > B$  e  $A > 0$
- ❑  $A < B$  e  $C > B$  e  $A < 0$
- ❑  $A < B$  ou  $C > B$
- ❑  $A < B$  ou  $C < B$
- ❑  $A > B$  ou  $C > B$
- ❑  $A > B$  ou  $C < B$
- ❑  $A < B$  ou  $C > B$  ou  $A > 0$
- ❑  $A < B$  ou  $C > B$  ou  $A < 0$
- ❑ não ( $A < B$  e  $C > B$ )
- ❑ não ( $A < B$  ou  $C > B$ )
- ❑ não ( $A > B$ ) e  $C > B$
- ❑ não ( $A > B$ ) ou  $B > C$

10

## Qual o resultado?

- ❑ Assuma  $A = 2$ ;  $B = 3$ ;  $C = 10$
- ❑ não ( $A > B$ )
- ❑ não ( $B < C$ )
- ❑  $A < B$  e  $C > B$
- ❑  $A < B$  e  $C < B$
- ❑  $A > B$  e  $C > B$
- ❑  $A > B$  e  $C < B$
- ❑  $A < B$  e  $C > B$  e  $A > 0$
- ❑  $A < B$  e  $C > B$  e  $A < 0$
- ❑  $A < B$  ou  $C > B$
- ❑  $A < B$  ou  $C < B$
- ❑  $A > B$  ou  $C > B$
- ❑  $A > B$  ou  $C < B$
- ❑  $A < B$  ou  $C > B$  ou  $A > 0$
- ❑  $A < B$  ou  $C > B$  ou  $A < 0$
- ❑ não ( $A < B$  e  $C > B$ )
- ❑ não ( $A < B$  ou  $C > B$ )
- ❑ não ( $A > B$ ) e  $C > B$
- ❑ não ( $A > B$ ) ou  $B > C$

11

## Operadores Lógicos em C/C++

Operador	Significado
!	negação
&&	conjunção
	disjunção

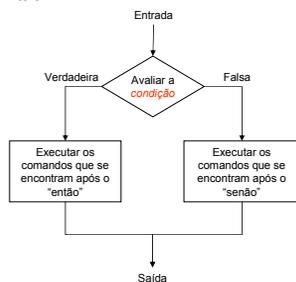
12

## Seleção (if-then-else)

### Pseudo-código

Se *condição* Então  
 ...  
 Senão  
 ...  
 Fim Se

### Fluxograma



### Diagrama de Nassi-Shneiderman



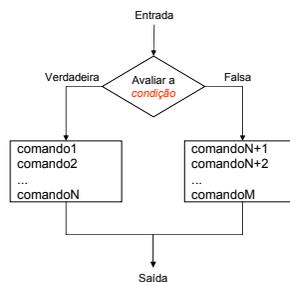
13

## Seleção (if-then-else)

### Pseudo-código

Se *condição* Então  
 comando1  
 comando2  
 ...  
 comandoN  
 Senão  
 comandoN+1  
 comandoN+2  
 ...  
 comandoM  
 Fim Se

### Fluxograma



### Diagrama de Nassi-Shneiderman



14

## Exemplo

Algoritmo Média. Este algoritmo lê duas notas, calcula a média aritmética delas e imprime uma mensagem indicativa se o aluno foi aprovado ou não, juntamente com a média obtida.

### Início

declare N1, N2, Media : real

Escreva ("Notas? ")

Leia (N1,N2)

Media ← (N1 + N2) / 2.0

Se (Media >= 5.0) Então

Escreva ("Aprovado com média",Media)

Senão

Escreva ("Reprovado com média",Media)

Fim Se

Fim

15

## Seleção (if-then-else) em C/C++

□ **if(*condição*)**  
 comando1;  
**else**  
 comando2;  
 ...  
 comandoN;  
**if(*condição*)**  
 { comando1;  
 ...  
 comandoN;  
 }  
**else**  
 comandoM;  
**if(*condição*)**  
 comando1;  
**else**  
 { comandoN+1;  
 ...  
 comandoM;  
 }

□ **if(*condição*)**  
 { comando1;  
 comando2;  
 ...  
 comandoN;  
 }  
**else**  
 { comandoN+1;  
 comandoN+2;  
 ...  
 comandoM;  
 }

□ Assim, se houver mais de um comando dentro de um comando **if** (tanto na parte **then** como **else**), eles devem ser colocados entre chaves **{ e }**

16

## Exemplo em C++

```
#include <iostream>
using namespace std;
```

```
/* Algoritmo Média. Este algoritmo lê duas notas, calcula a
média aritmética delas e imprime uma mensagem indicativa se
o aluno foi aprovado ou não, juntamente com a média obtida.
*/
```

```
int main()
```

```
{ float N1,N2,Media;
```

```
cout << "Notas? ";
```

```
cin >> N1 >> N2;
```

```
Media = (N1 + N2) / 2.0;
```

```
if (Media >= 5.0)
```

```
cout << "Aprovado com média " << Media << endl;
```

```
else
```

```
cout << "Reprovado com média " << Media << endl;
```

```
return 0;
```

```
}
```

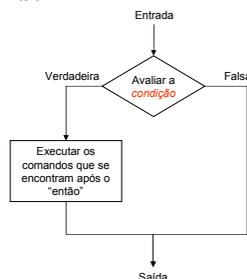
18

## Seleção (if-then)

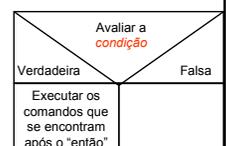
### Pseudo-código

Se *condição* Então  
 ...  
 Fim Se

### Fluxograma



### Diagrama de Nassi-Shneiderman



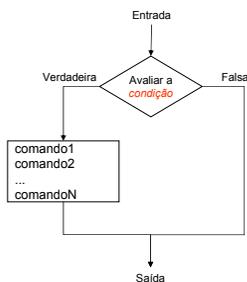
19

## Seleção (if-then)

### Pseudo-código

**Se** *condição* **Então**  
 comando1  
 comando2  
 ...  
 comandoN  
**Fim Se**

### Fluxograma



### Diagrama de Nassi-Shneiderman



20

## Exemplo

Algoritmo Premio. Este algoritmo lê o valor de venda de um vendedor e calcula seu prêmio. Se o valor do prêmio for maior do que 2000, o programa imprime uma mensagem de congratulação.

### Início

declare venda, premio : real

Escreva("Valor da venda? ")

Leia(venda)

premio ← 0.1 \* venda

**Se** (premio > 2000) **Então**

    Escreva("Congratulações pelo premio de ",premio)

**Fim Se**

Escreva("Valor da venda = ",venda, "Premio = ",premio)

**Fim**

21

## Seleção (if-then) em C/C++

❑ **if**(*condição*)  
 comando1;

❑ **if**(*condição*)  
 { comando1;  
 ...  
 comandoN;  
 }

❑ Assim, se houver mais de um comando dentro de um comando **if**, eles devem ser colocados entre chaves **{ e }**

22

## Exemplo em C++

```

#include <iostream>
using namespace std;

/* Algoritmo Premio. Este algoritmo lê o valor de venda de um vendedor
e calcula seu prêmio. Se o valor do prêmio for maior do que 2000, o
programa imprime uma mensagem de congratulação.
*/
int main()
{ float venda,premio;

  cout << "Valor da venda? ";
  cin >> venda;
  premio = 0.1 * venda;
  if (premio > 2000)
    cout << "Congratulacoes pelo premio de " << premio
    << endl;
  cout << "Valor da venda = " << valor << endl
    << "Premio = " << premio << endl;
  return 0;
}
    
```

23

## Exemplo em C++

```

#include <iostream>
using namespace std;

/* Algoritmo Premio. Este algoritmo lê o valor de venda de um vendedor
e calcula seu prêmio. Se o valor do prêmio for maior do que 2000, o
programa imprime uma mensagem de congratulação.
*/
int main()
{ float venda,premio;

  cout << "Valor da venda? ";
  cin >> venda;
  premio = 0.1 * venda;
  if (premio > 2000)
    { cout << "Congratulacoes pelo premio de ";
      cout << premio << endl;
    }
  cout << "Valor da venda = " << valor << endl
    << "Premio = " << premio << endl;
  return 0;
}
    
```

24

## Exercícios

1. Preparar um algoritmo para ler um valor inteiro e determinar se ele é par ou ímpar.
2. Preparar um algoritmo para ler a base e a altura de um triângulo e imprimir a área (area = 0.5 \* base \* altura). Durante a preparação dos dados para este algoritmo é possível que se cometa um erro e entrem valores negativos para a base ou a altura. Isto é indesejável, pois a saída impressa será negativa. Prever no algoritmo a possibilidade de verificar valores negativos na entrada de dados. Se um valor negativo for fornecido, imprimir uma mensagem identificando este valor como a base ou a altura (isso permite corrigir o erro mais facilmente). Testar o algoritmo cuidadosamente. Tomar cuidado no caso de ambos os valores serem negativos. Isso produziria uma área positiva e o erro não seria detectado.

27

## Solução Exercício 1

Algoritmo ParÍmpar. Este algoritmo lê um valor inteiro N e determina se ele é par ou ímpar.

```

Início
declare N : inteiro

Escreva("Forneca um numero: ")
Leia(N)
Se (N % 2 = 0) Então
    Escreva("Numero ",N," e' par")
Senão
    Escreva("Numero ",N," e' ímpar")
Fim Se
Fim
    
```

28

## Solução Exercício 1 em C++

```

#include <iostream>
using namespace std;
/* Algoritmo ParÍmpar. Este algoritmo lê um valor
inteiro N e determina se ele é par ou ímpar.
*/

int main()
{ int N;

    cout << "Forneca um numero: ";
    cin >> N;
    if (N % 2 == 0)
        cout << "Numero " << N << " e' par" << endl;
    else
        cout << "Numero " << N << " e' ímpar" << endl;
    return 0;
}
    
```

30

## Else pendente em C/C++

❑ Como os comandos seguintes são interpretados pelo compilador? (a) ou (b)?

```

if(condição1)
if(condição2)
comando1;
else
comando2;
    
```

(a)	(b)
<pre> if(condição1) {     if(condição2)         comando1;     else         comando2; }                 </pre>	<pre> if(condição1) {     if(condição2)         comando1;     else         comando2; }                 </pre>

31

## Else pendente em C/C++

❑ Para resolver essa ambigüidade do *else* pendente adota-se o critério que o *else* sempre se refere ao comando *if* mais interno (mais próximo)

❑ Solução (a)

```

if(condição1)
{
    if(condição2)
        comando1;
    else
        comando2;
}
    
```

32

## Seleção (case)

Caso (variável)

```

const1:
comando1
const2:
comando2
comando3
    
```

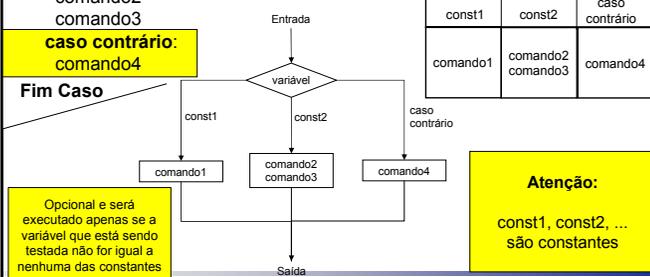
**caso contrário:**  
comando4

Fim Caso

Fluxograma

Diagrama de Nassi-Shneiderman

variável		
const1	const2	caso contrário
comando1	comando2 comando3	comando4



Opcional e será executado apenas se a variável que está sendo testada não for igual a nenhuma das constantes

**Atenção:**  
const1, const2, ...  
são constantes

33

## Seleção (case) em C/C++

❑ **switch**(variável)

```

{ case const1:
    comando1;
  break;
  case const2:
    comando2;
    comando3;
    break;
  default:
    comando4;
}
    
```

O comando *break* faz com que o *switch* seja interrompido assim que um dos *case* seja executado. Se após a execução de um dos *case* não houver um *break*, a execução continua.

Opcional e será executado apenas se a variável que está sendo testada não for igual a nenhuma das constantes

34

## Exemplo em C++

```

/* Imprime o valor de um número como
nenhum, um, dois,
alguns e muitos
*/
...
cout << "Forneça um número";
cin >> numero;
switch (numero)
{
  case 0:
    cout << "Nenhum" << endl;
    break;
  case 1:
    cout << "Um" << endl;
    break;
  case 2:
    cout << "Dois" << endl;
    break;
  case 3:
  case 4:
  case 5:
    cout << "Alguns" << endl;
    break;
  default:
    cout << "Muitos" << endl;
}
...

```

36

## Exemplo em C++

```

/* Imprime o valor de um número como
nenhum, um, dois, alguns e muitos
*/
...
cout << "Forneça um número";
cin >> numero;
switch (numero)
{
  case 0:
    cout << "Nenhum" << endl;
    break;
  case 1:
    cout << "Um" << endl;
    break;
  case 2:
    cout << "Dois" << endl;
    break;
  case 3:
  case 4:
  case 5:
    cout << "Alguns" << endl;
    break;
  default:
    cout << "Muitos" << endl;
}
...

```

Forneça um número 0  
Nenhum

Forneça um número 1  
Um

Forneça um número 2  
Dois

Forneça um número 3  
Alguns

Forneça um número 4  
Alguns

Forneça um número 5  
Alguns

Forneça um número 6  
Muitos

37

## Exemplo em C++

```

/* Imprime o valor de um número
como nenhum, um, dois, alguns
e muitos
*/
...
cout << "Forneça um número";
cin >> numero;
switch (numero)
{
  case 0:
    cout << "Nenhum" << endl;
  case 1:
    cout << "Um" << endl;
  case 2:
    cout << "Dois" << endl;
  case 3:
  case 4:
  case 5:
    cout << "Alguns" << endl;
  default:
    cout << "Muitos" << endl;
}
...

```

Forneça um número 0  
Nenhum  
Um  
Dois  
Alguns  
Muitos

Forneça um número 1  
Um  
Dois  
Alguns  
Muitos

Forneça um número 2  
Dois  
Alguns  
Muitos

Forneça um número 3  
Alguns  
Muitos

38

## Exercício

- Elabore um algoritmo que leia o código de um determinado produto e mostre sua classificação. Utilize a seguinte tabela como referência:

Código	Classificação
1	Alimento perecível
2 ou 3	Alimento não-perecível
4, 5 ou 6	Vestuário
7	Higiene pessoal
8 até 12	Limpeza
Qualquer outro código	Inválido

39

## Solução

Algoritmo Produto. Este algoritmo lê o código de um determinado produto e mostra sua classificação

Início

declare código : inteiro

Escreva("Codigo do produto? ")

Leia(código)

**Caso (código)**

1: Escreva("Alimento perecível")  
 2,3: Escreva("Alimento não-perecível")  
 4,5,6: Escreva("Vestuário")  
 7: Escreva("Higiene pessoal")  
 8,9,10,11,12: Escreva("Limpeza")  
**caso contrário:**  
 Escreva("Codigo invalido")

**Fim Caso**

Fim

40

## Solução em C++

```

#include <iostream>
using namespace std;

/* Algoritmo Produto. Este algoritmo lê
o código de um determinado produto e
mostra sua classificação.
*/

int main()
{
  int código;

  cout << "Codigo do produto? ";
  cin >> código;
  switch (código)
  {
    case 1:
      cout << "Alim.perecível";
      break;
    case 2:
    case 3:
      cout << "Alim.não-perecível";
      break;
    case 4:
    case 5:
    case 6:
      cout << "Vestuário";
      break;
    case 7:
      cout << "Higiene pessoal";
      break;
    case 8:
    case 9:
    case 10:
    case 11:
    case 12:
      cout << "Limpeza";
      break;
    default:
      cout << "Codigo invalido";
  }
  return 0;
}

```

42

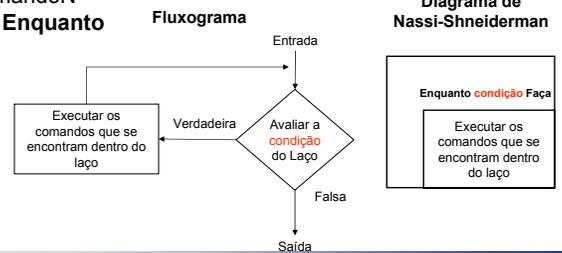
## Laços ou *Loops*

- ❑ Laços permitem que uma seqüência de comandos seja executada repetidas vezes
- ❑ Tipos de Laços
  - Condicionais
    - ❖ Com teste no início (tipo “enquanto”)
    - ❖ Com teste no final (tipo “repita...”)
  - Contados
    - ❖ Com teste no início (tipo “para  $i \leftarrow 1,2,\dots,N$ ”)

43

## Laços Condicionais/Teste no Início

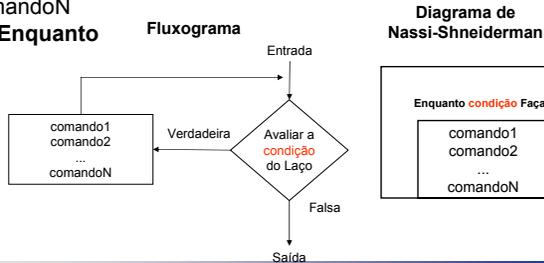
- ❑ Enquanto **condição** Faça
    - comando1
    - comando2
    - ...
    - comandoN
- Fim Enquanto**



44

## Laços Condicionais/Teste no Início

- ❑ Enquanto **condição** Faça
    - comando1
    - comando2
    - ...
    - comandoN
- Fim Enquanto**



45

## Exemplo

Algoritmo MédiaClasse. Esse algoritmo calcula a média das notas de uma classe com 10 alunos.

```

Início
declare contador : inteiro
declare soma,nota,media : real

contador ← 1
soma ← 0.0
Enquanto contador <= 10 Faça
  Escreva("Nota do aluno ", contador," : ")
  Leia(nota)
  soma ← soma + nota
  contador ← contador + 1
Fim Enquanto
media ← soma / 10.0
Escreva("Média da classe = ",media)
Fim
    
```

46

## Laços Condicionais em C/C++

- ❑ **while(condição)**
  - comando1;
- ❑ **while(condição)**
  - { comando1;
  - comando2;
  - ...
  - comandoN;
  - }
- ❑ Assim, se houver mais de um comando dentro de um comando **while**, eles devem ser colocados entre chaves **{ e }**

47

## Exemplo em C++

```

#include <iostream>
using namespace std;
// Algoritmo MédiaClasse. Esse algoritmo calcula a média
// das notas de uma classe com 10 alunos.
int main()
{ int contador;
  float soma,nota,media;

  contador = 1;
  soma = 0.0;
  while(contador <= 10)
  { cout << "Nota do aluno " << contador << " : ";
    cin >> nota;
    soma = soma + nota;
    contador = contador + 1;
  }
  media = soma / 10.0;
  cout << "Média da classe = " << media << endl;
  return 0;
}
    
```

49

## Exercício

- Altere o Algoritmo MédiaClasse para que ele funcione para uma classe com N alunos. Assuma  $N > 0$ .

50

## Solução

Algoritmo MédiaClasse2. Esse algoritmo calcula a média das notas de uma classe com N alunos

Início

```
declare contador,N : inteiro
declare soma,nota,media : real

contador ← 1
soma ← 0.0
Escreva("Numero de alunos: ")
Leia(N)
Enquanto contador <= N Faça
  Escreva("Nota do aluno ",contador,",": ")
  Leia(nota)
  soma ← soma + nota
  contador ← contador + 1
Fim Enquanto
media ← soma / N
Escreva("Média da classe = ",media)
Fim
```

51

## Solução em C++

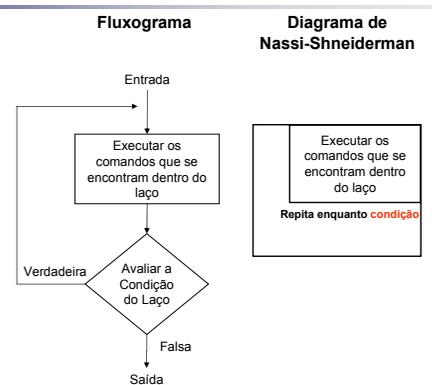
```
#include <iostream>
using namespace std;
// Algoritmo MédiaClasse2 Esse algoritmo calcula a média
// das notas de uma classe com N alunos
int main()
{ int contador,N;
  float soma,nota,media;

  contador = 1;
  soma = 0.0;
  cout << "Numero de alunos: ";
  cin >> N;
  while(contador <= N)
  { cout << "Nota do aluno " << contador << ": ";
    cin >> nota;
    soma = soma + nota;
    contador = contador + 1;
  }
  media = soma / N;
  cout << "Média da classe = " << media << endl;
  return 0;
}
```

53

## Laços Condicionais/Teste no Fim

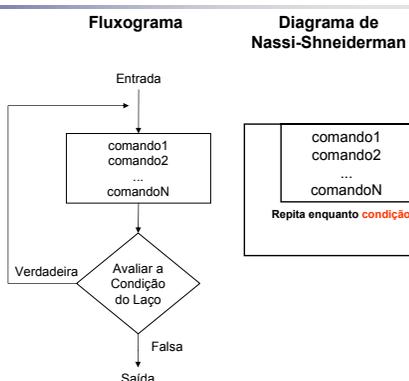
- Repita  
comando1  
comando2  
...  
comandoN  
**Enquanto condição**



54

## Laços Condicionais/Teste no Fim

- Repita  
comando1  
comando2  
...  
comandoN  
**Enquanto condição**



55

## Exemplo

Algoritmo MédiaClasse3. Esse algoritmo calcula a média das notas de uma classe com N alunos. O algoritmo só aceita valores de notas entre 0.0 e 10.0

Início

```
declare contador,N : inteiro
declare soma,nota,media : real

contador ← 1
soma ← 0.0
Escreva("Numero de alunos: ")
Leia(N)
Enquanto contador <= N Faça
  Repita
    Escreva("Nota do aluno ",contador,",": ")
    Leia(nota)
  Enquanto (nota < 0 ou nota > 10)
    soma ← soma + nota
    contador ← contador + 1
Fim Enquanto
media ← soma / N
Escreva("Média da classe = ",media)
Fim
```

56

## Laços Condicionais em C/C++

- ❑ **do**

```
{ comando1;
  comando2;
  ...
  comandoN;
} while(condição);
```
- ❑ **do**

```
comando1;
comando2;
...
comandoN;
while(condição);
```
- ❑ Assim, evite **do/while** sem **{ e }** pois pode é possível confundir o comando **do/while** com o comando **while**

57

## Exemplo em C++

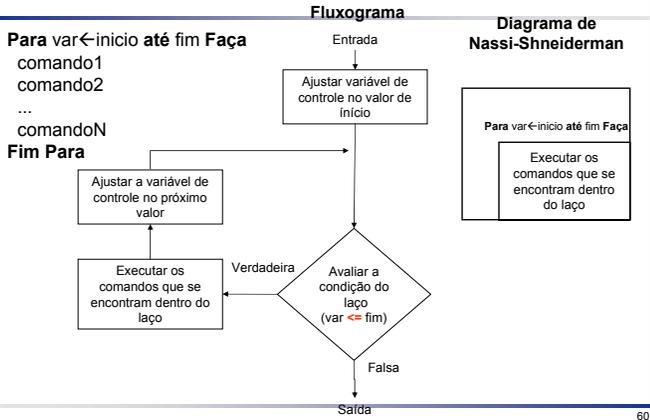
```
#include <iostream>
using namespace std;
//Algoritmo MédiaClasse3. Esse algoritmo calcula a média das notas de uma classe
//com N alunos. O algoritmo só aceita valores de notas entre 0 e 10.

int main()
{ int contador,N;
  float soma,nota,media;

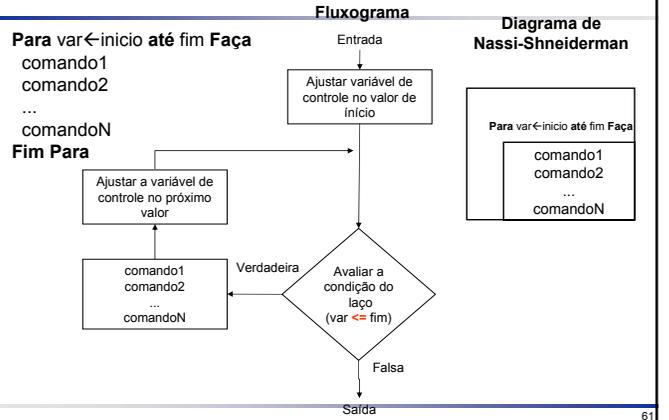
  contador = 1;
  soma = 0.0;
  cout << "Numero de alunos: ";
  cin >> N;
  while(contador <= N)
  { do
    { cout << "Nota do aluno " << contador << ": ";
      cin >> nota;
    } while (nota < 0 || nota > 10);
    soma = soma + nota;
    contador = contador + 1;
  }
  media = soma / N;
  cout << "Média da classe = " << media;
  return 0;
}
```

59

## Laços Contados



## Laços Contados



## Exemplo

Algoritmo MédiaClasse4. Esse algoritmo calcula a média das notas de uma classe com N alunos.

Início

```
declare contador,N : inteiro
declare soma,nota,media : real
```

```
soma ← 0.0
Escreva ("Numero de alunos: ")
Leia (N)
Para contador ← 1 até N Faça
  Escreva ("Nota do aluno ",contador," : ")
  Leia(nota)
  soma ← soma + nota
Fim Para
media ← soma / N
Escreva ("Média da classe = ",media)
Fim
```

62

## Laços Contados em C/C++

- for(<inicialização>; <condição>; <alteração>)**
- ❑ <inicialização>
    - O comando colocado nesta parte é realizado incondicionalmente
    - Utilizado para atribuir o valor inicial à variável de controle do laço
  - ❑ <condição>
    1. Após a execução da inicialização, a <condição> é avaliada
    2. Se a <condição> for falsa, o comando "for" termina (ou seja, os comandos colocados dentro do laço "for" não são executados)
    3. Se a <condição> for verdadeira, os comandos dentro do laço "for" são executados
    4. Ao final da execução do último comando dentro do laço, o comando de <alteração> é executado (automaticamente) e a <condição> é novamente avaliada (repetindo-se o processo a partir do passo 2)
  - ❑ <alteração>
    - Indica como a variável de controle do laço deve ser alterada
    - Normalmente é um incremento (variável++) ou decremento (variável--), mas outras alterações são possíveis (tais como variável = variável + 2, etc)

63

## Laços Contados em C/C++

❑ **for(i=inicio; i<= fim; i++)**

```
{ comando1;  
  comando2;  
  ...  
  comandoN;  
}
```

❑ **for(i=inicio; i<= fim; i++)**

```
  comando1;
```

❑ Assim, se houver mais de um comando dentro de um comando **for**, eles devem ser colocados entre chaves **{ e }**

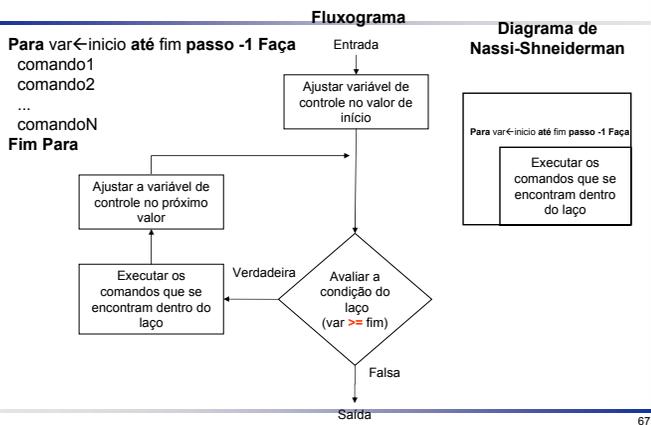
64

## Exemplo em C++

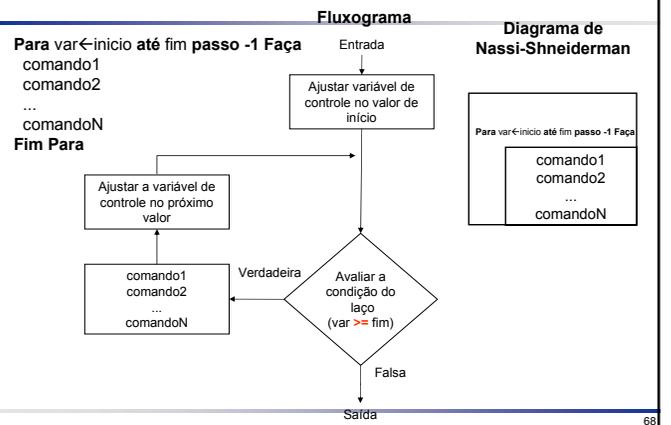
```
#include <iostream>  
using namespace std;  
/* Algoritmo MédiaClasse4. Esse algoritmo calcula a média das notas de  
uma classe com N alunos.  
*/  
int main()  
{ int contador,N;  
  float soma,nota,media;  
  
  soma = 0.0;  
  cout << "Numero de alunos: ";  
  cin >> N;  
  for(contador = 1; contador <= N; contador++)  
  { cout << "Nota do aluno " << contador << ": ";  
    cin >> nota;  
    soma = soma + nota;  
  }  
  media = soma / N;  
  cout << "Média da classe = " << media;  
  return 0;  
}
```

65

## Laços Contados



## Laços Contados



## Exemplo

Algoritmo MédiaClasse5. Esse algoritmo calcula a média das notas de uma classe com N alunos.

Início

```
declare contador,N : inteiro  
declare soma,nota,media : real
```

```
soma ← 0.0
```

```
Escreva ("Numero de alunos: ")
```

```
Leia (N)
```

```
Para contador ← N até 1 passo -1 Faça
```

```
  Escreva ("Nota do aluno ",contador," : ")
```

```
  Leia (nota)
```

```
  soma ← soma + nota
```

```
Fim Para
```

```
media ← soma / N
```

```
Escreva ("Média da classe = ",media)
```

Fim

69

## Laços Contados em C/C++

❑ **for(i=inicio; i>= fim; i--)**

```
{ comando1;  
  comando2;  
  ...  
  comandoN;  
}
```

❑ **for(i=inicio; i>= fim; i--)**

```
  comando1;
```

70

## Exemplo em C++

```
#include <iostream>
using namespace std;
/* Algoritmo MédiaClasse5. Esse algoritmo calcula a média das notas de
uma classe com N alunos.
*/
int main()
{ int contador,N;
  float soma,nota,media;

  soma = 0.0;
  cout << "Numero de alunos: ";
  cin >> N;
  for(contador = N; contador >= 1; contador--)
  { cout << "Nota do aluno " << contador << ": ";
    cin >> nota;
    soma = soma + nota;
  }
  media = soma / N;
  cout << "Média da classe = " << media;
  return 0;
}
```

72

## Exercício

1. Elabore um programa para calcular o fatorial de um número inteiro  $n$ . Por definição:
  - $0! = 1$
  - $n! = n * (n - 1)!$
2. Um programador está preocupado em relação a seu desempenho no curso de computação. Em seu primeiro programa ele cometeu um erro; em seu segundo, dois erros; no terceiro, quatro erros; assim por diante. Ele está cometendo, por programa, duas vezes o número de erros que cometeu no programa anterior. O curso dura treze semanas, com dois problemas por semana. Preparar um algoritmo para calcular o número de erros que este programador espera cometer em seu programa final.

73

## Solução 1 Exercício 1

Algoritmo Fatorial1. Calcula o fatorial de um número  $n \geq 0$ .

Início

declare  $n, i, \text{fatorial}$  : inteiro

Escreva("Valor de  $n$ : ")

Leia( $n$ )

$\text{fatorial} \leftarrow 1$

Para  $i \leftarrow 2$  até  $n$  Faça

$\text{fatorial} \leftarrow \text{fatorial} * i$

Fim Para

Escreva("fatorial de ", $n$ ," = ", $\text{fatorial}$ )

Fim

74

## Solução 1 Exercício 1 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Fatorial1. Calcula o fatorial de um
// número n >= 0
int main()
{ int n,i,fatorial;

  cout << "Valor de n: ";
  cin >> n;
  fatorial = 1;
  for(i=2; i<=n; i++)
    fatorial = fatorial * i;
  cout << "fatorial de " << n << " = " << fatorial << endl;
  return 0;
}
```

75

## Solução 2 Exercício 1

Algoritmo Fatorial2. Calcula o fatorial de um número  $N \geq 0$ .

Início

declare  $N, i, \text{fatorial}$  : inteiro

Escreva("Valor de  $N$ : ")

Leia( $N$ )

$\text{fatorial} \leftarrow 1$

$i \leftarrow 1$

Enquanto  $i \leq N$  Faça

$\text{fatorial} \leftarrow \text{fatorial} * i$

$i \leftarrow i + 1$

Fim Para

Escreva("fatorial de ", $N$ ," = ", $\text{fatorial}$ )

Fim

77

## Solução 2 Exercício 1 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Fatorial2. Calcula o fatorial de
// um número N >= 0
int main()
{ int N,i,fatorial;

  cout << "Valor de N: ";
  cin >> N;
  fatorial = 1;
  i = 1;
  while(i <= N)
  { fatorial = fatorial * i;
    i++;
  }
  cout << "fatorial de " << N << " = " << fatorial << endl;
  return 0;
}
```

79

## Solução Exercício 2

Algoritmo Erros. Calcula o número de erros de um programador, num curso de 13 semanas, 2 problemas por semana.

```
Início
declare semanas,pps,total_problemas,p,erros : inteiro

semanas ← 13
pps ← 2 // problemas por semana
total_problemas ← semanas * pps
erros ← 1 // primeiro problema
Para p ← 2 até total_problemas Faça
    erros ← 2 * erros
Fim Para
Escreva("Numero de erros = ",erros)
Fim
```

80

## Solução Exercício 2 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Erros. Calcula o número de erros de um
// programador, num curso de 13 semanas, 2 problemas por
// semana.
int main()
{ int semanas, pps, total_problemas, p, erros;

    semanas = 13;
    pps = 2; // problemas por semana
    total_problemas = semanas * pps;
    erros = 1; // primeiro problema
    for(p = 2; p <= total_problemas; p++)
        erros = 2 * erros;
    cout << "Numero de erros = " << erros << endl;
    return 0;
}
```

Numero de erros = 0

84

## Solução Exercício 2 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Erros. Calcula o número de erros de um
// programador, num curso de 13 semanas, 2 problemas por
// semana.
int main()
{ int semanas, pps, total_problemas, p;
  long int erros;

    semanas = 13;
    pps = 2; // problemas por semana
    total_problemas = semanas * pps;
    erros = 1; // primeiro problema
    for(p = 2; p <= total_problemas; p++)
        erros = 2 * erros;
    cout << "Numero de erros = " << erros << endl;
    return 0;
}
```

Numero de erros = 33554432

85

## Exercícios Adicionais

- Um número inteiro positivo é, por definição, primo se ele tem exatamente dois divisores distintos entre si. Por exemplo, os primeiros cinco números primos são 2, 3, 5, 7 e 11. Preparar um algoritmo para ler um número e determinar se ele é ou não um número primo.
- Um distribuidor de brinquedos fez um acordo de compra de 10.000 brinquedos pequenos embalados em caixas de formato retangular de tamanhos variados. Ele pretende re-embalar estas caixas em esferas coloridas de plástico e revendê-las como pacotes surpresas. As esferas são fornecidas com quatro diâmetros diferentes: 10, 15, 20 e 25 cm. Para pedir as esferas, ele precisa saber quantas de cada diâmetro ele necessita. Sabendo que a diagonal de uma caixa retangular com as dimensões A, B e C, dada por

$$D = \sqrt{A^2 + B^2 + C^2}$$

é sua maior medida, o distribuidor deve calcular os comprimentos das diagonais das caixas e determinar o número de caixas com diagonal igual ou menor do que 10 cm, maior do que 10, mas menor ou igual a 15 e assim por diante. As dimensões de cada caixa são digitadas em linhas separadas. Preparar um algoritmo para ler estes dados e determinar o número de esferas de cada tamanho necessárias para re-embalar os brinquedos.

86

## Solução 1 Exercício 1

Algoritmo Primo. Lê um número N e determina se ele é ou não um número primo.

```
Início
Escreva("Numero?")
Leia(N)
Se N = 1 Então
    Escreva("1 nao e' primo")
Senão
    Calcule o número de divisores de N
    Se divisores = 2 Então
        Escreva(N, " e' primo")
    Senão
        Escreva(N, " nao e' primo")
Fim Se
Fim
```

87

## Solução 1 Exercício 1

Algoritmo Primo. Lê um número N e determina se ele é ou não um número primo.

```
Início
Escreva("Numero?")
Leia(N)
Se N = 1 Então
    Escreva("1 nao e' primo")
Senão
    divisores ← 2 // todo número N é divisível por 1 e N
    Para i ← 2 até N-1 Faça
        Se (N % i = 0) Então // N é divisível por i?
            divisores ← divisores + 1
    Fim Para
    Se divisores = 2 Então // Primo tem apenas 2 divisores
        Escreva(N, " e' primo")
    Senão
        Escreva(N, " nao e' primo")
Fim Se
Fim
```

88

## Solução 1 Exercício 1 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Primo. Lê um número N e determina se ele é ou não um
// número primo

int main()
{ int N,divisores,i;

  cout << "Numero?";
  cin >> N;
  if(N == 1)
    cout << "1 nao e' primo";
  else
  { divisores = 2; // todo número N é divisível por 1 e N
    for(i=2; i<=N-1; i++)
      if(N % i == 0) // N é divisível por i?
        divisores++;
    if(divisores == 2) // Primo tem apenas 2 divisores
      cout << N << "e' primo" << endl;
    else
      cout << N << "nao e' primo" << endl;
  }
  return 0;
}
```

90

## Solução 2 Exercício 1

- ❑ A solução apresentada não considera o fato que qualquer número par não é primo, exceto o número dois
- ❑ Assim, antes de calcular o número de divisores de um número, é possível avaliar se isso realmente ocorre; em caso afirmativo, o programa pode informar (sem calcular o número de divisores) que o número não é primo
- ❑ Fica como exercício você alterar a solução apresentada para refletir essa melhoria

91

## Solução Exercício 2

Algoritmo Esferas. Calcula qtde de esferas de 10, 15, 20 e 25 cm de diâmetros.

```
Início
esferas10 ← esferas15 ← esferas20 ← esferas25 ← esferas_grandes ← 0
Para i ← 1 até 10000 Faça
  Escreva("Dimensões A, B, C da caixa?")
  Leia(A,B,C) // dimensões da caixa
  D ← sqrt(A^2 + B^2 + C^2) // diagonal da caixa
  Se D <= 10 Então
    esferas10 ← esferas10 + 1
  Senão
    Se D <= 15 Então
      esferas15 ← esferas15 + 1
    Senão
      Se D <= 20 Então
        esferas20 ← esferas20 + 1
      Senão
        Se D <= 25 Então
          esferas25 ← esferas25 + 1
        Senão
          esferas_grandes ← esferas_grandes + 1
    Fim Se
  Fim Se
Fim Se
Fim Para
Escreva("Total de esferas com 10 cm =",esferas10)
Escreva("Total de esferas com 15 cm =",esferas15)
Escreva("Total de esferas com 20 cm =",esferas20)
Escreva("Total de esferas com 25 cm =",esferas25)
Escreva("Total de esferas com >25 cm =",esferas_grandes)
Fim
```

92

## Solução Exercício 2 em C++

```
#include <iostream>
using namespace std;
// Algoritmo Esferas. Calcula qtde de esferas de 10, 15, 20 e 25 cm de diâmetros.
int main()
{ int i,esferas10, esferas15, esferas20, esferas25, esferas_grandes;
  float A, B, C, D;

  esferas10 = esferas15 = esferas20 = esferas25 = esferas_grandes = 0;
  for(i=1; i<= 10000; i++)
  { cout << "Dimensões A, B, C da caixa?";
    cin >> A >> B >> C; // dimensões da caixa
    D = sqrt(A*A + B*B + C*C); // diagonal da caixa
    if(D <= 10)
      esferas10++;
    else
      if(D <= 15)
        esferas15++;
      else
        if(D <= 20)
          esferas20++;
        else
          if(D <= 25)
            esferas25++;
          else
            esferas_grandes++;
    }
  }
  cout << "Total de esferas com 10 cm = " << esferas10 << endl;
  cout << "Total de esferas com 15 cm = " << esferas15 << endl;
  cout << "Total de esferas com 20 cm = " << esferas20 << endl;
  cout << "Total de esferas com 25 cm = " << esferas25 << endl;
  cout << "Total de esferas com >25 cm = " << esferas_grandes << endl;
  return 0;
}
```

94

## Algumas Normas de Programação

- ❑ O tempo de leitura e compreensão de um programa é muito maior do que o tempo para escrevê-lo. A leitura do programa deve ser o mais fácil possível
- ❑ Todo comando em C/C++ termina com um ponto-e-vírgula (;)
- ❑ Alinhe um { na mesma coluna que seu } correspondente. Todos os comandos entre { e } devem ser deslocados dois espaços para a direita
- ❑ Ao usar um comando de seleção (if, switch) ou repetição (for, while, do-while), desloque dois espaços para a direita (indentation) os comandos contidos neles. Se há vários níveis de indentation, cada nível deve ser deslocado à direita por dois espaços
- ❑ Colocar um ponto-e-vírgula logo após um comando de seleção ou repetição constitui um erro. Por exemplo, a mensagem "a == b" sempre será impressa, independentemente se a é ou não igual a b nos comandos seguintes:  
if (a == b);  
cout << "a == b";
- ❑ Evite colocar mais de um comando em uma mesma linha

95

## Resumo

- ❑ Uma variável é uma entidade que possui um valor, sendo conhecida por um nome (ou identificador)
- ❑ Expressão é uma combinação válida de variáveis, constantes e operadores
- ❑ Condição é uma expressão na qual o resultado é verdadeiro ou falso
- ❑ As três estruturas de controle em programação são:
  - Seqüência (um comando é executado após o comando anterior)
  - Seleção (se-então-senão, se-então, caso)
  - Repetição ou Laço
    - ❖ Contado (para variável ← início até final faça)
    - ❖ Condicional (enquanto, repita/enquanto)

96