

# Curso de C

## *Estruturas Condicionais*



# Estruturas Condicionais

## Objetivos:

- Aprender:
  - Como vincular a execução de blocos de sentenças à condições
  - Quando utilizar cada uma das estruturas disponíveis

# Estruturas Condicionais

## Roteiro:

- Exemplo
- Condições e operadores relacionais
- Comando `if`
- Comando `if ... else`
- Operadores lógicos
- Comandos `if ... else if` em cascata
- Comando `switch`

# Estruturas Condicionais

## Motivação:

```
int main(int argc, char* argv[]) {  
  
    double pi = 3.141592;  
    double raio, area, perimetro;  
  
    printf("Digite o raio: ");  
    scanf("%lf", &raio);  
  
    area = pi * (raio * raio);  
    perimetro = 2.0 * pi * raio;  
  
    printf("Raio: %lf \n", raio);  
    printf("Area: %lf \n", area);  
    printf("Perimetro: %lf \n", perimetro);  
    return 0;  
}
```

Seqüencial:  
⇒ Algoritmos Simples

Condições:  
⇒ Como decidir se  
deve executar ou  
não um bloco

# Estruturas Condicionais

*Condições*





# Condições

## O que são condições:

- Expressões matemáticas convencionais!
- Testam validade de uma afirmação
- Resultado: número inteiro
- Interpretação:
  - 0: significa *falso*
  - não 0: significa *verdadeiro*
- Operadores especiais:

<    <=    >    >=    ==    !=

# Condições

## Operador de Comparação:

Maior que:

esquerda > direita

Resultado:

1: se esq > dir

0: caso contrário

Valor constante

Variável

Outra expressão



# Condições

## Operadores de comparação:

Expressão	Condição	Resultado
$a > b$	se $a > b$	1 (verdadeiro)
	se $a \leq b$	0 (falso)
$a \geq b$	se $a \geq b$	1 (verdadeiro)
	se $a < b$	0 (falso)
$a < b$	se $a < b$	1 (verdadeiro)
	se $a \geq b$	0 (falso)
$a \leq b$	se $a \leq b$	1 (verdadeiro)
	se $a > b$	0 (falso)
$a == b$	se $a = b$	1 (verdadeiro)
	se $a \neq b$	0 (falso)
$a != b$	se $a \neq b$	1 (verdadeiro)
	se $a = b$	0 (falso)

Atenção

Cuidado!

Não confundir igualdade (==) com atribuição (=)!



# Condições

## Exemplo:

```
int a, b;  
...  
a = (1 < 2);  
b = (3 <= 2);
```

~~aa~~ == ? 1 (verdade)  
~~bb~~ == ? 0 (falso)

```
int c = 3;  
int d = 10;  
int e, f;  
...  
e = (c == d);  
f = (c != d);
```

~~ee~~ == ? 0 (falso)  
~~ff~~ == ? 1 (verdade)

# Estruturas Condicionais

*if...*



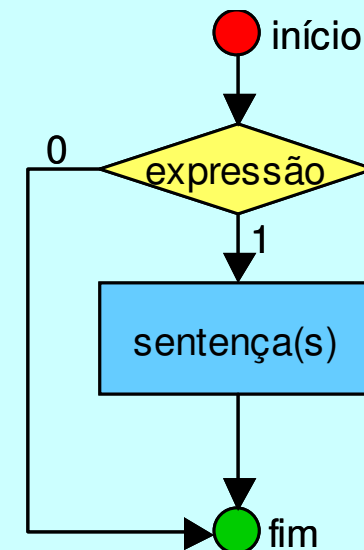
# if...

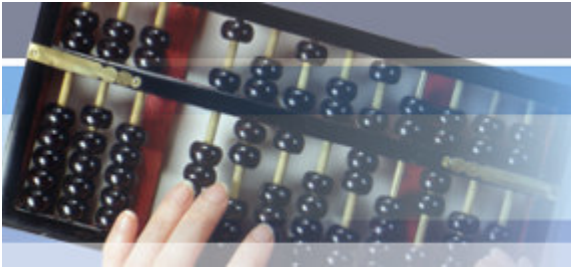
## Estrutura if...

Executa código somente se uma condição for verdadeira.  
(resultado da expressão diferente de zero)

Sintaxe:

```
início;  
if (expressão) {  
    sentença;  
    sentença;  
    ...  
}  
fim;
```





# if...

## Exemplo if...

```
int main(int argc, char *argv[]) {  
    int idade;  
  
    printf("Digite sua idade: ");  
    scanf("%d", &idade);  
  
    if (idade >= 18) {  
        printf("Já pode obter habilitação!");  
    }  
  
    return 0;  
}
```

*EstruturasCondicionais\Idade01\Idade01.vcproj*

# if...

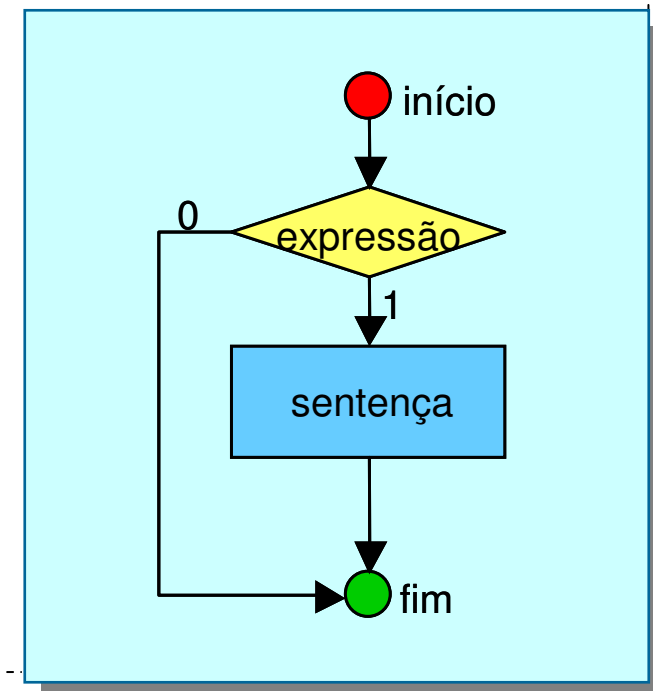
## Estrutura if...

Sintaxe simplificada:

- Uma única sentença
- Sem bloco

Sintaxe:

```
início;  
if (expressão)  
    sentença;  
fim;
```



# Estruturas Condicionais

*if...else...*

A background image showing a close-up of a traditional Chinese abacus (suanpan) with black beads on wooden rods. Two hands are visible, with fingers positioned to move the beads. The image is slightly faded and serves as a backdrop for the text.

# if...else...

## Exemplo:

```
int main(int argc, char *argv[]) {  
    int idade;  
  
    printf("Digite sua idade: ");  
    scanf("%d", &idade);  
  
    if (idade >= 18) {  
        printf("Você já pode obter habilitação!");  
    }  
    if (idade <= 17) {  
        printf("Espere mais alguns anos!");  
    }  
    return 0;  
}
```

Condições  
mutuamente  
exclusivas

*EstruturasCondicionais\ldade02\ldade02.vcproj*



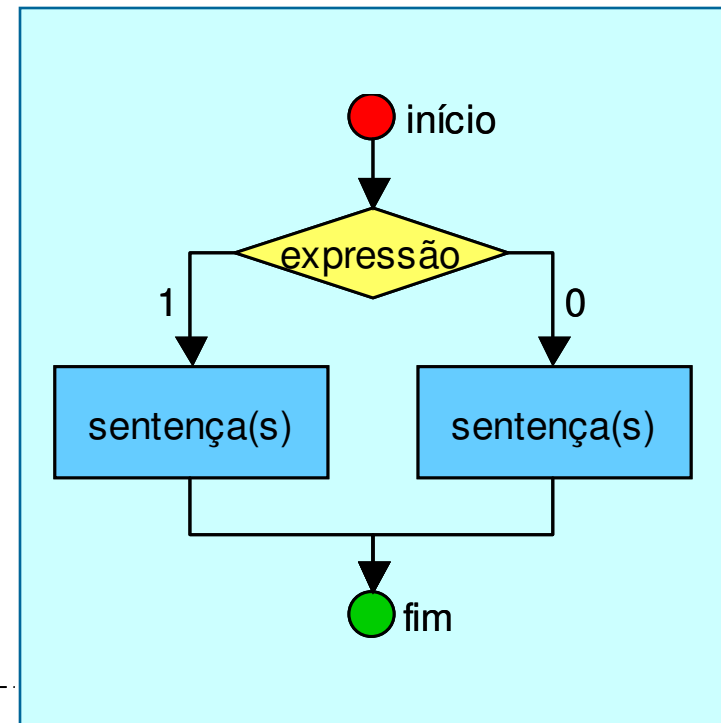
# if...else...

## Estrutura if...else...

Condição verdadeira: executa o primeiro bloco.  
Caso contrário: executa o segundo bloco.

Sintaxe:

```
início;  
if (expressão) {  
    sentença;  
    ...  
} else {  
    sentença;  
    ...  
}  
fim;
```



# if...else...

## Exemplo if...else...

```
int main(int argc, char *argv[]) {
    int idade, diferenca_tempo;
    printf("Digite sua idade: ");
    scanf("%d", &idade);

    if (idade >= 18) {
        diferenca_tempo = idade - 18;
        printf("Voce tem habilitacao ha %d ano(s)",
            diferenca_tempo);
    } else {
        diferenca_tempo = 18 - idade;
        printf("Espere mais %d ano(s)!\n",
            diferenca_tempo);
    }
    return 0;
}
```

*EstruturasCondicionais\ldade03\ldade03.vcproj*

# if...else...

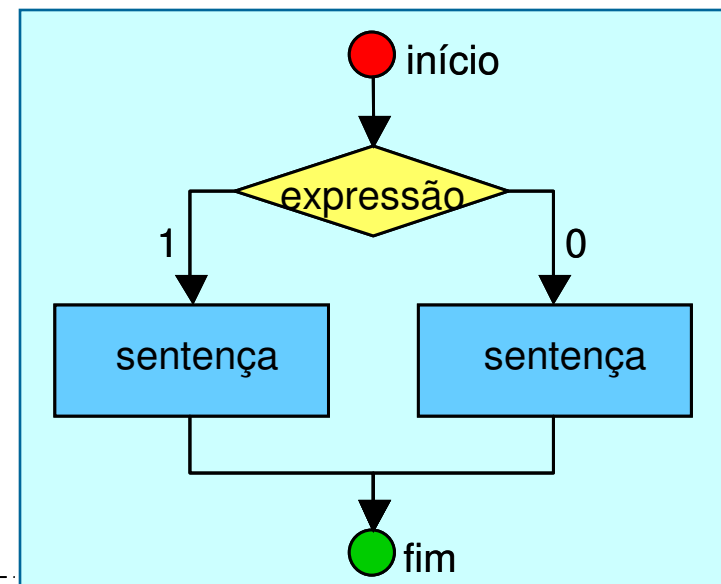
## Estrutura if...else...

Sintaxe simplificada:

- Uma única sentença
- Sem bloco

Sintaxe:

```
início;  
if (expressão)  
    sentença;  
else  
    sentença;  
fim;
```



# Estruturas Condicionais

*Operadores Lógicos*



# Operadores Lógicos

## O que são Operadores Lógicos:

- Objetivo:
  - Condições com mais de um teste
- Combinação: (E / AND)
  - Duas expressões precisam ser verdadeiras
- Alternativas: (OU / OR)
  - Uma das duas expressões precisa ser verdadeira
- Negação: (NÃO / NOT)
  - A condição precisa ser falsa

# Operadores Lógicos

## Exemplos conceituais:

- Carteira de Habilitação:
  - Idade maior ou igual que 18 anos
  - Aprovação nos exames
  - Combinação E / AND
- Cargo de boa remuneração:
  - Excelente currículo profissional
  - Indicação de uma pessoa influente
  - Alternativa OU / OR



# Operadores Lógicos

## Operadores Lógicos:

Expressão	Condição	Resultado
<code>a &amp;&amp; b</code>	se <i>a</i> não 0 e <i>b</i> não 0	1 (verdadeiro) (0 caso contrário)
<code>a    b</code>	se <i>a</i> não 0 ou <i>b</i> não 0	1 (verdadeiro) (0 caso contrário)
<code>! a</code>	se <i>a</i> é 0	1 (verdadeiro) (0 caso contrário)



# Operadores Lógicos

## Tabelas Verdade:

cond1 && cond2 (E / AND)		cond1	
		<i>falso</i>	<i>verdadeiro</i>
cond2	<i>falso</i>	<i>falso</i>	<i>falso</i>
	<i>verdadeiro</i>	<i>falso</i>	<i>verdadeiro</i>

cond1    cond2 (OU / OR)		cond1	
		<i>falso</i>	<i>verdadeiro</i>
cond2	<i>falso</i>	<i>falso</i>	<i>verdadeiro</i>
	<i>verdadeiro</i>	<i>verdadeiro</i>	<i>verdadeiro</i>

cond	! cond
<i>falso</i>	<i>verdadeiro</i>
<i>verdadeiro</i>	<i>falso</i>



# Operadores Lógicos

## Exemplos:

Média pelo menos 7.0 e frequência de pelo menos 40 aulas:

```
aprovado = (nota >= 7.0) && (frequencia >= 40)
```

```
nota = 5.0; frequencia = 30; ⇒ aprovado = 0;
```

```
nota = 8.0; frequencia = 30; ⇒ aprovado = 0;
```

```
nota = 5.0; frequencia = 50; ⇒ aprovado = 0;
```

```
nota = 8.0; frequencia = 50; ⇒ aprovado = 1;
```



# Operadores Lógicos

## Exemplos:

Média superior a 5.0 **ou** frequência superior a 30 aulas:

```
aprovado = (nota > 5.0) || (frequencia > 30)
```

```
nota = 3.0; frequencia = 20; ⇒ aprovado = 0;
```

```
nota = 8.0; frequencia = 20; ⇒ aprovado = 1;
```

```
nota = 3.0; frequencia = 50; ⇒ aprovado = 1;
```

```
nota = 8.0; frequencia = 50; ⇒ aprovado = 1;
```

# Operadores Lógicos

## Exemplo:

```
int main(int argc, char *argv[]) {
    int idade;
    float media;
    printf("Digite sua idade: ");
    scanf("%d", &idade);
    printf("Digite sua media nos exames: ");
    scanf("%f", &media);

    if ( (idade >= 18) && (media >= 5.0) ) {
        printf("Voce esta aprovado!");
    } else {
        printf("Ainda nao aprovado!");
    }
    return 0;
}
```

*EstruturasCondicionais\ldade04\ldade04.vcproj*

# Estruturas Condicionais

*if...else if...else*

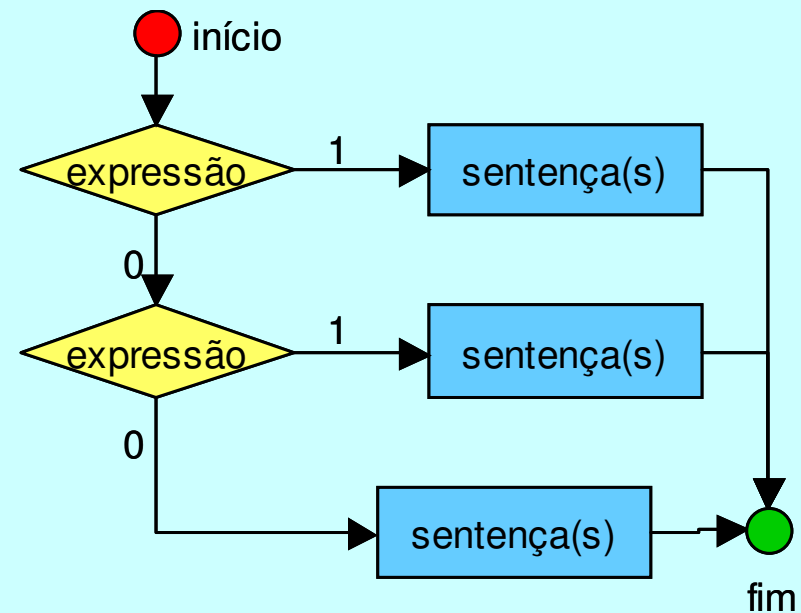
# if...else if...else

## Estrutura if...else if...else...

Múltiplas decisões mutuamente exclusivas

Sintaxe:

```
início;  
if (expressão) {  
    sentença;  
    ...  
} else if (expressão) {  
    sentença;  
    ...  
} else {  
    sentença;  
    ...  
}  
fim;
```





# if...else if...else

## Exemplo if...else if...else...

```
int main(int argc, char *argv[]) {  
    int idade;  
  
    printf("Digite sua idade: ");  
    scanf("%d", &idade);  
  
    if ( (idade >= 0) && (idade < 18) ) {  
        printf("Nao possui habilitacao.\n");  
    } else if ( (idade >= 18) && (idade < 65) ) {  
        printf("Renove exames a cada 5 anos.\n");  
    } else if (idade >= 65) {  
        printf("Renove exames a cada 3 anos.\n");  
    }  
    return 0;  
}
```

*EstruturasCondicionais\ldade05\ldade05.vcproj*



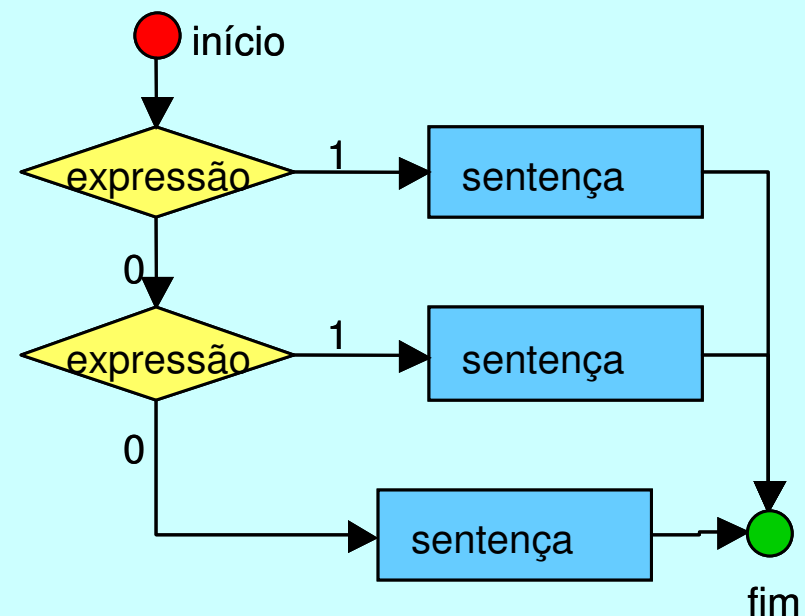
# if...else if...else

## Estrutura if...else if...else...

Sintaxe simplificada:

Sintaxe:

```
início;  
if (expressão)  
    sentença;  
else if (expressão)  
    sentença;  
else  
    sentença;  
fim;
```



# Estruturas Condicionais

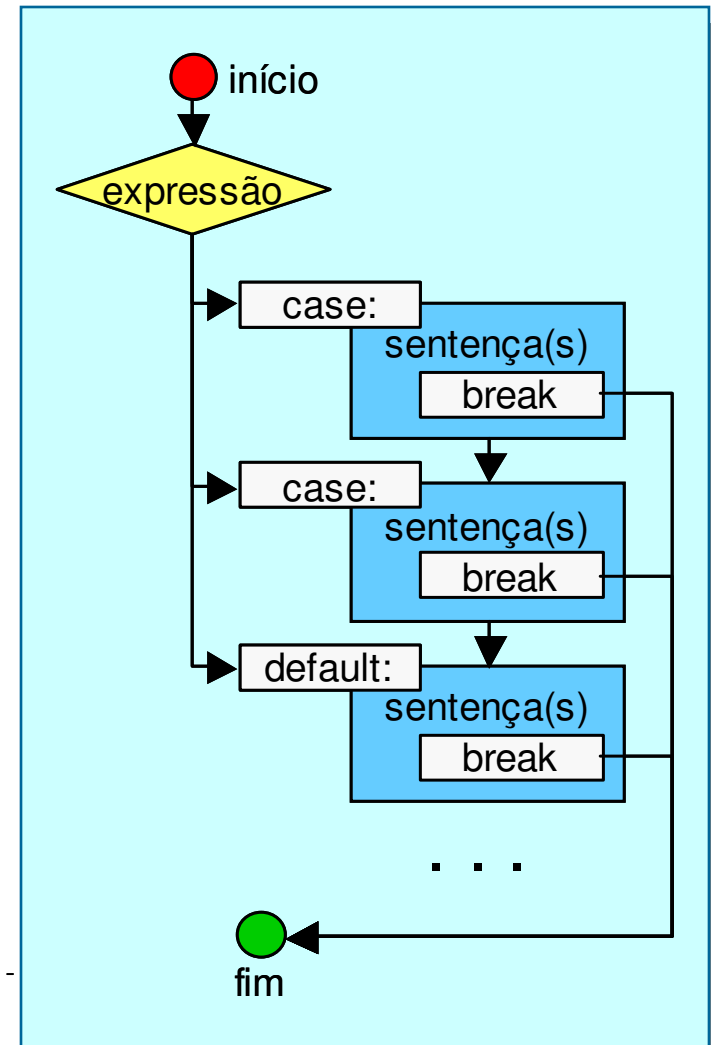
*switch*



# switch

## Conceitos:

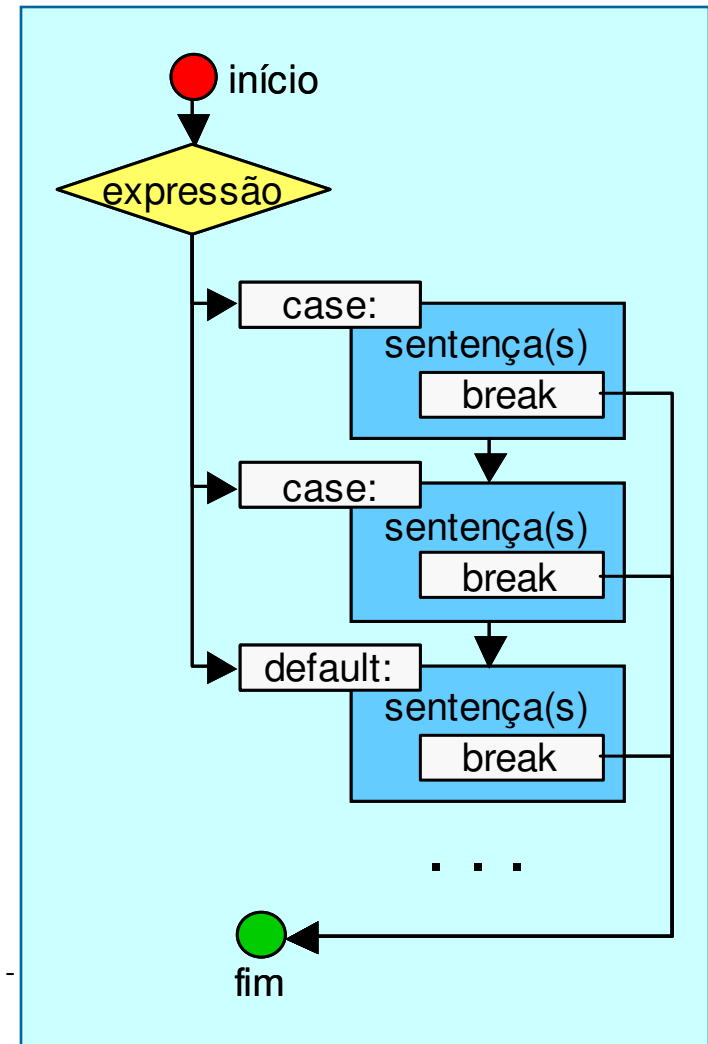
- Várias alternativas (case)
  - Valores constantes
- Avalia expressão
  - Compara com cada case
  - Entra no case correspondente
- default: se não encontra alternativa
- break – Finaliza switch
- Execução prossegue através dos cases!



# switch

Sintaxe:

```
início;  
switch (expressão) {  
    case valor1:  
        sentença(s);  
        break;  
    case valor2:  
        sentença(s);  
        break;  
    case valor3:  
        sentenças;  
        break;  
    default:  
        sentença(s);  
        break;  
}  
fim;
```





# switch

## Exemplo switch...

```
int main(int argc, char *argv[]) {  
    float preco, preco_final;  
    char categoria;  
  
    printf("Digite o preço do ingresso: ");  
    scanf("%f", &preco);  
    printf("E - estudante, \n");  
    printf("A - aposentado, \n");  
    printf("N - normal\n");  
    printf("Digite a categoria do cliente (E/A/N): ");  
    scanf("%c", &categoria);  
  
    switch (categoria) ...  
  
    return 0;  
}
```



# switch

## Exemplo switch...

```
switch (categoria) {  
    case 'e': case 'E':  
        preco_final = preco * 0.50f;  
        printf("Preco: %f\n", preco_final);  
        break;  
    case 'a': case 'A':  
        preco_final = preco * 0.70f;  
        printf("Preco: %f\n", preco_final);  
        break;  
    case 'n': case 'N':  
        printf("Preco sem desconto: %f\n", preco);  
        break;  
    default:  
        printf("Categoria invalida!\n");  
        break;  
}
```

*EstruturasCondicionais\Cinema01\Cinema01.vcproj*

# Estruturas Condicionais

*Casos de Uso*





# Casos de Uso

## Quando usar cada estrutura?

- **if**
  - Execução condicional de um bloco
  - `if + return`: Para finalizar execução sob determinadas condições (ex: erros)
- **if...else...**
  - Execução condicional de um bloco ou outro
  - Condições mutuamente exclusivas
  - Aceitar um dado ou imprimir mensagem de erro

# Casos de Uso

## Quando usar cada estrutura?

- **if...else if...else**
  - Testar intervalos de valores
  - Várias condições mutuamente exclusivas
  - Condições com prioridade
- **switch (...) ...**
  - Expressão com alternativas discretas
  - Alternativas em grande número
  - Se processamento for igual para para várias alternativas

# Estruturas Condicionais

- Comparativo01
- Letras
- Triangulo
- Sinais
- EfeitoColateral
- SegundoGrau
- TestaSwitch
- Expressao



# Estruturas Condicionais

*Fim do Capítulo*

