

Aula 4: Estruturas Condicionais

Professor(a): João Eduardo Montandon (103)

Virgínia Fernandes Mota (106)

jemaf.github.io

<http://www.dcc.ufmg.br/~virginiaferm>

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA



- À medida em que os programas ficam mais complexos, devemos fazer todo possível para mantê-lo simples. Uma simplificação é uma técnica chamada **programação estruturada**.
 - O programa é quebrado em estruturas
 - Redirecionamento em duas direções
 - Condições
 - Operadores lógicos

- As estruturas de controle devem ser simples. A complexidade do programa está em como essas estruturas de controle são combinadas. As três estruturas de controle básicas são:
 - Sequência → uma operação é executada após a outra. é a estrutura de controle que já vimos até agora.
 - Seleção → Uma escolha entre um conjunto de operações. Veremos ainda hoje.
 - Iteração → Repetição de um conjunto de operações. Veremos em outra aula.

O paradigma de programação estruturada define que todos os programas podem ser reduzidos a apenas essas três estruturas. Existem outros paradigmas de programação, como a programação orientada a objetos, porém estes outros não serão abordados neste curso.

- A estrutura de seleção define um bloco de operações de acordo com uma escolha. Existem quatro estruturas de seleção:
 - *if*
 - *else if*
 - *else*
 - *switch*

A estrutura de seleção *if* tem a seguinte sintaxe:

```
1 ...  
2 if (condição){  
3     bloco de operações  
4 }  
5 ...
```

- **condição** pode assumir um valor verdadeiro ou falso.
- o bloco de operações só é executado se a condição for verdadeira.
- um bloco de operações pode conter outros blocos de operações, ou seja, pode haver estruturas de seleção **aninhadas**.

Estruturas Aninhadas

```
1 ...  
2 if (condição){  
3     ...  
4     if (outra condição){  
5         bloco de operações  
6     }  
7     ...  
8 }  
9 ...
```

- Uma condição é uma expressão lógica que é avaliada como sendo verdadeira ou falsa.
 - Em C, uma condição falsa recebe o valor 0
 - Uma condição verdadeira recebe um valor diferente de 0
- Geralmente, uma condição é composta por uma ou mais comparações envolvendo valores.
 - expressão *operador de comparação* expressão
 - $x + 4 > 9$
 - $(x - y) \leq (a + b)$
- Ao processar as comparações em uma condição, o valor resultante sempre será verdadeiro ou falso.

- Os operadores se diferem não somente na funcionalidade de cada um, mas também na precedência: os operadores relacionais tem precedência sobre os operadores de igualdade.

Os operadores são:

- `==` e `!=` → "igual" e "diferente"
- `!` → "negação"
- `>` e `>=` → "maior" e "maior igual"
- `<` e `<=` → "menor" e "menor igual"
- `&&` e `||` → "e lógico" e "ou lógico"

Avalie as expressões:

- $(9 \geq 7 + 3) \ \&\& \ (7 \% 3 == 1)$
- $(6 > 2) \ || \ 25/5 == 4$
- $7 \leq 9 \ || \ 6 > 5 \ \&\& \ 7 == 2$
- $(7 \leq 9 \ || \ 6 > 5) \ \&\& \ 7 == 2$
- $25 > 3 \ \&\& \ !(6 == 4)$

Avalie as expressões:

- $(9 \geq 7 + 3) \&\& (7 \% 3 == 1)$ **F**
- $(6 > 2) \parallel 25/5 == 4$ **V**
- $7 \leq 9 \parallel 6 > 5 \&\& 7 == 2$ **V**
- $(7 \leq 9 \parallel 6 > 5) \&\& 7 == 2$ **F**
- $25 > 3 \&\& !(6 == 4)$ **V**

A cláusula *else*

- Uma versão mais completa da cláusula *if* contém blocos de operações tanto na parte verdadeira quanto na falsa.
- O bloco de operações associado a condição falsa começa com a cláusula *else*:

```
1  ...  
2  if (condição){  
3      bloco de operações  
4  }  
5  else {  
6      bloco de operações  
7  }  
8  ...
```

Exemplo

Faça um programa para ler dois inteiros e determinar qual é o maior entre eles.

```
1 #include <stdio.h>
2 int main() {
3     int A, B, maior;
4     printf("Digite 2 valores inteiros \n");
5     scanf("%d %d", &A, &B);
6     if (A > B) {
7         maior = A;
8     }
9     else{
10        maior = B;
11    }
12    printf("\n O maior é %d", maior);
13    return 0;
14 }
```

- Quando queremos associar diferentes blocos de operações a diferentes condições, usamos a construção *else if*.

```
1 ...  
2 if (condição){  
3     bloco de operações  
4 }  
5 else if (condição){  
6     bloco de operações  
7 }  
8 else {  
9     bloco de operações  
10 }  
11 ...
```

Exemplo

Faça um programa para ler dois inteiros e determinar qual é o maior entre eles ou se eles são iguais.

```
1 #include <stdio.h>
2 int main() {
3     int A, B;
4     printf("Digite 2 valores inteiros \n");
5     scanf("%d %d", &A, &B);
6     if (A > B) {
7         printf("\n O maior é %d", A);
8     }
9     else if (A < B) {
10        printf("\n O maior é %d", B);
11    }
12    else {
13        printf("\n Os dois são iguais");
14    }
15    return 0;
16 }
```

O comando *switch*

- A linguagem C inclui uma alternativa **multi-escolha** para a cláusula *if*.
- O comando *switch* tem a seguinte sintaxe:

```
1 switch(expressão integral) {  
2     case valor1: bloco de operações  
3                 break;  
4     ...  
5     case valorn: bloco de operações  
6                 break;  
7 }
```

Exemplo

```
1 #include <stdio.h>
2 int main() {
3     int epoca;
4     printf("Digite o numero do trimestre: ")
5     scanf("%d", &epoca);
6     switch(epoca) {
7         case 1: printf("Verão");
8                 break;
9         case 2: printf("Outuno");
10                break;
11        case 3: printf("Inverno");
12                break;
13        case 4: printf("Primavera");
14                break;
15        default: printf("Trimestre inválido");
16                 break;
17    }
18    return 0;
19 }
```


Construir um algoritmo para ler os coeficientes A, B e C de uma equação do segundo grau $Ax^2 + Bx + C = 0$ e se:

Delta for negativo: imprima a mensagem "Não há solução real."

Delta for maior ou igual a zero: calcular e imprimir as raízes da equação.

Exercício - Solução

```
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     float a, b, c, delta, x1, x2;
5     printf("Leia os coeficientes A, B e C da equação de 2o grau:")
6     scanf("%f %f %f", &a, &b, &c);
7     delta = pow(b,2) - 4*a*c;
8     if (delta >= 0){
9         x1 = (-b + sqrt(delta)) / (2*a);
10        x2 = (-b - sqrt(delta)) / (2*a);
11        printf("As raízes são: %f %f", x1, x2);
12    }
13    else {
14        printf("Não há solução real");
15    }
16    return 0;
17 }
```

1. Ler um número inteiro e informar se ele é par ou ímpar.
2. Faça um programa para ler dois inteiros e determinar qual é o maior entre eles ou se eles são iguais.
3. Desenvolver um algoritmo para ler o número do dia da semana e imprimir o seu respectivo nome por extenso. Considerar o número 1 como domingo, 2 para segunda etc. Caso o dia da semana não exista (menor do que 1 ou maior do que 7), exibir a mensagem "Dia da semana inválido".
4. Fazer um algoritmo para ler dois números e um dos símbolos das operações: +, -, * e /. Imprimir o resultado da operação efetuada sobre os números lidos.
5. Faça um programa que leia 3 comprimentos (x,y e z) e responda se eles formam um triângulo, ou seja, se $x < y + z$ e $y < x + z$ e $z < x + y$.
6. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:
 - para homens: $(72.7 * h) - 58$
 - para mulheres: $(62.1 * h) - 44.7$Informe também se a pessoa está acima ou abaixo deste peso.

Na próxima aula...

Estruturas de Repetição