Aula 4: Estruturas de Seleção

Adriano Veloso

Algoritmos e Estruturas de Dados I - DCC/UFMG

Estruturação

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

Estruturas de Controle

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.

Estrutura de Seleção

A estrutura de seleção define um bloco de operações de acordo com uma escolha. Existem quatro estruturas de seleção:

1. *if*
2. *else if*
3. *else*
4. *switch*

A Clásula *if*

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

```
...
if(condição) {
    bloco de operações
}
...
```

A Clásula *if*

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

```
...
if(condição) {
    bloco de operações
}
...
```

- condição pode assumir um valor **verdadeiro** ou **falso**

A Clásula *if*

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

```
...
if(condição) {
    bloco de operações
}
...
```

- condição pode assumir um valor **verdadeiro** ou **falso**
- o bloco de operações só executado se a condição for verdadeira.

A Clásula *if*

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

```
...
if(condição) {
    bloco de operações
}
...
```

- condição pode assumir um valor **verdadeiro** ou **falso**
- o bloco de operações só 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

```
...
if(condição) {
    ...
    if(outra condição) {
        bloco de operações
    }
    ...
}
...
```

Condições

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

Condições

Geralmente, uma condição é composta por uma ou mais comparações envolvendo valores. Uma comparação tem a seguinte forma:
expressão operador de comparação expressão

- $x + 4 > 9$
- $(x - y) <= (a + b)$
- ...

Condições

Geralmente, uma condição é composta por uma ou mais comparações envolvendo valores. Uma comparação tem a seguinte forma:
expressão operador de comparação expressão

- $x + 4 > 9$
- $(x - y) <= (a + b)$
- ...

ao processar as comparações em uma condição, o valor resultante sempre será verdadeiro ou falso

Operadores Relacionais e de Igualdade

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 $!=$ $\rightarrow$ "igual" e "diferente"
- $!$ $\rightarrow$ "negação"
- $>$ e $>=$ $\rightarrow$ "maior" e "maior igual"
- $<$ e $<=$ $\rightarrow$ "menor" e "menor igual"
- && e $||$ $\rightarrow$ "e lógico" e "ou lógico"

Exemplos

Avalie as expressões:

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

Exemplos

Avalie as expressões:

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

A Clásula *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*:

```
...
if(condição) {
    bloco de operações
}
```

A Clásula *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*:

```
...
if(condição) {
    bloco de operações
}
else {
    bloco de operações
}
...
```

A Construção *else if*

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

```
...
if(condição) {
    bloco de operações
}
```

A Construção *else if*

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

```
...
if(condição) {
    bloco de operações
}
else if(condição) {
    bloco de operações
}
```

A Construção *else if*

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

```
...
if(condição) {
    bloco de operações
}
else if(condição) {
    bloco de operações
}
else {
    bloco de operações
}
...
```

O Comando *switch*

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

```
switch(expressão integral) {
    case valor_1: bloco de operações
    break;

    ...

    case valor_n: bloco de operações
    break;

}
```

Exemplo

```
...
int numero;
scanf("%d", &numero);
switch(numero * 2) {
    case 6: printf("Seis");
    break;

    case 4: printf("Quatro");
    break;

    case 8:
    case 10: printf("Pode ser tanto oito quanto dez");
    break;
}
```

Contato

adrianov@dcc.ufmg.br