

FAETERJ - Paracambi

Análise e Desenvolvimento de Sistemas

Programação Estruturada – PRG - I

Prof. Carlos Eduardo Costa Vieira

Resumo da Apresentação

- Estrutura Condicional
 - Condicional Simples;
 - Exemplo;
 - Condicional Composto;
 - Exemplo;
 - Estrutura Case;
 - Exemplo;
- Bibliografia.

Estrutura Condicional

- Em uma estrutura condicional, os comandos são executados de cima para baixo, uma única vez;
- Porém, dependendo da entrada de dados e das condições, nem todos os comandos serão executados.

Estrutura Condicional

■ Condicional Simples

- A seleção de uma ação é feita a partir da especificação de uma alternativa dirigida por uma condição. Quando a condição é verdadeira, executa-se os comandos do `if`; caso contrário, vai para o final da estrutura;

■ Sintaxe

```
if (condição) {  
    <comando(s)>;  
}
```

Exemplo (Aula2.1.c)

- Elaborar um algoritmo para calcular se um determinado número é par ou ímpar.

```
#include <stdio.h>
int main() {
    int num;
    printf("Entre com um número: ");
    scanf("%d", &num);
    if (num%2 == 0) {
        printf("O número %d é par !!!\n", num);
    }
    if (num%2 != 0) {
        printf("O número %d é ímpar !!!\n", num);
    }
    return 0;
}
```

Estrutura Condicional

■ Condicional Composto

- A seleção de uma ação é feita a partir da especificação de duas alternativas dirigida por uma condição. Se a condição for verdadeira, o(s) comando(s)_1 do `if` será(ão) executado(s); caso contrário, o(s) comando(s)_2 do `else` será(ão) executado(s);

■ Sintaxe

```
if (condição) {  
    comando(s)_1;  
}  
else {  
    comando(s)_2;  
}
```

Exemplo (Aula2.2.c)

- Elaborar um algoritmo para calcular se um determinado número é par ou ímpar.

```
#include <stdio.h>
int main() {
    int num;
    printf("Entre com um número: ");
    scanf("%d", &num);
    if (num%2 == 0) {
        printf("O número %d é par !!!\n", num);
    }
    else{
        printf("O número %d é ímpar !!!\n", num);
    }
    return 0;
}
```

Estrutura Condicional

■ Estrutura Case

- O comando `switch(variável)` avalia o valor de uma variável para decidir qual case será executado. A variável deverá ser obrigatoriamente do tipo `char`, `int` ou `long`. O comando `break` deverá ser utilizado para impedir a execução dos comandos cases subsequentes. Quando o valor da variável não coincidir com aqueles especificados nos cases, será executado então o `default`.

■ Sintaxe

```
switch (variável) {  
    case valor1: <comando(s)_1>;  
        break;  
    case valor2: <comando(s)_2>;  
        break;  
    ...  
    default: <comando(s)_n>;  
}
```


Exemplo (Aula2.3.c)

```
#include <stdio.h>
```

```
int main() {  
    int epoca;  
    printf("Entre com a época: ");  
    scanf("%d", &epoca);  
    switch(epoca) {  
        case 1: printf("Verão.\n");  
        break;  
        case 2: printf("Outono.\n");  
        break;  
        case 3: printf("Inverno.\n");  
        break;  
        case 4: printf("Primavera.\n");  
        break;  
        default: printf("Época Inválida.\n");  
    }  
    return 0;  
}
```

Bibliografia

- ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. **Fundamentos da Programação de Computadores: Algoritmos: Pascal, C/C++ e Java.** 3. ed. São Paulo: Pearson Prentice Hall, 2012.
- DEITEL, P. J; DEITEL, H. M. **C: Como Programar.** 6. ed. São Paulo: Pearson Prentice Hall, 2011.
- OLIVEIRA, Ulysses. **Programando em C: Fundamentos.** 22 ed. Rio de Janeiro: Ciência Moderna, 2008.