



# Estrutura de Dados

Ricardo José Cabeça de Souza

[www.ricardojcsouza.com.br](http://www.ricardojcsouza.com.br)

[ricardo.souza@ifpa.edu.br](mailto:ricardo.souza@ifpa.edu.br)

Parte 3

# CONTROLE DE FLUXO



- **CONTROLE DE FLUXO**

- Linguagem C provê as construções fundamentais de controle de fluxo necessárias para programas bem estruturados

- Agrupamentos de comandos
- Tomadas de decisão (**if-else**)
- Laços com teste de encerramento no início (**while, for**) ou no fim (**do-while**)
- Seleção de um dentre um conjunto de possíveis casos (**switch**)

# CONTROLE DE FLUXO



- **DECISÕES COM if**

– **if** é o comando de decisão básico em C

```
if (expr) {  
    bloco de comandos 1  
    ...  
}
```

```
if ( expr ) {  
    bloco de comandos 1  
    ...  
}  
else {  
    bloco de comandos 2  
    ...  
}
```

# CONTROLE DE FLUXO



- **DECISÕES COM if**

- Se *expr* produzir um valor diferente de 0 (verdadeiro), o ***bloco de comandos 1*** será executado
- A inclusão do **else** requisita a execução do ***bloco de comandos 2*** se a expressão produzir o valor 0 (falso)
- Se dentro de um bloco tivermos apenas um comando a ser executado, as chaves podem ser omitidas

```
if ( expr )  
    comando1;  
else  
    comando2;
```

```
if ( expr )  
{  
    bloco de comandos 1  
    ...  
}  
else  
{  
    bloco de comandos 2  
    ...  
}
```

# CONTROLE DE FLUXO



- **DECISÕES COM if**
  - Exemplo

```
#include <stdio.h>

int main (void)
{
    int a, b;
    printf("Insira dois numeros inteiros:");
    scanf ("%d%d", &a, &b);
    if (a%2 == 0)
        if (b%2 == 0)
            printf("Voce inseriu dois numeros pares!\n");
    return 0;
}
```

# CONTROLE DE FLUXO



- **DECISÕES COM if**

– Cuidado com o aninhamento de comandos **if-else**

Entrada de dados = 5

```
/* temperatura (versao 1 - incorreta) */
#include <stdio.h>

int main (void)
{
    int temp;
    printf("Digite a temperatura: ");
    scanf("%d", &temp);
    if (temp < 30)
        if (temp > 20)
            printf(" Temperatura agradável \n");
    else
        printf(" Temperatura muito quente \n");
    return 0;
}
```

```
/* temperatura (versao 2) */
#include <stdio.h>

int main (void)
{
    int temp;
    printf ( "Digite a temperatura: " );
    scanf ( "%d", &temp );
    if ( temp < 30 )
    {
        if ( temp > 20 )
            printf ( " Temperatura agradável \n" );
    }
    else
        printf ( " Temperatura muito quente \n" );
    return 0;
}
```

```
/* temperatura (versao 3) */
#include <stdio.h>

int main (void)
{
    int temp;
    printf("Digite a temperatura: ");
    scanf("%d", &temp);

    if (temp < 10)
        printf("Temperatura muito fria \n");
    else if (temp < 20)
        printf(" Temperatura fria \n");
    else if (temp < 30)
        printf("Temperatura agradável \n");
    else
        printf("Temperatura muito quente \n");
    return 0;
}
```

# CONTROLE DE FLUXO



- **ESTRUTURAS DE BLOCO**

- C é composta por estruturas de blocos
- Cada chave aberta e fechada em C representa um bloco
- A variável **i**, definida dentro do bloco do **if**, só existe dentro deste bloco

```
...  
if ( n > 0 )  
{  
    int i;  
    ...  
}  
...          /* a variável i não existe neste ponto do programa */
```

# CONTROLE DE FLUXO



- **OPERADOR CONDICIONAL**

- Trata-se de um operador que substitui construções do tipo **if-else**
- Se a **condição** for verdadeira, a **expressão1** é avaliada; caso contrário, avalia-se a **expressão2**

*condição ? expressão1 : expressão2;*

```
/* Condicional */  
#include <stdio.h>  
#include <conio.h>  
int main (void)  
{  
    int temp;  
    printf ( "Digite a temperatura: " );  
    scanf ( "%d", &temp );  
    temp < 30 ? temp=5: temp=10;  
    printf ( " Temp:%d\n",temp );  
    getch();  
    return 0;  
}
```



# CONTROLE DE FLUXO



- **CONSTRUÇÕES COM LAÇOS - while**
  - Procedimentos que devem ser executados em vários passos
  - Enquanto ***expr*** for avaliada em verdadeiro, o bloco de comandos é executado repetidamente

```
while (expr)  
{  
    bloco de comandos  
    ...  
}
```

# CONTROLE DE FLUXO



- **CONSTRUÇÕES COM LAÇOS - while**

- Exemplo

```
/* Fatorial */  
  
#include <stdio.h>  
  
int main (void)  
{  
    int i;  
    int n;  
    int f = 1;  
  
    printf("Digite um número inteiro nao negativo:");  
    scanf("%d", &n);  
  
    /* calcula fatorial */  
    i = 1;  
    while (i <= n)  
    {  
        f *= i;  
        i++;  
    }  
  
    printf(" Fatorial = %d \n", f);  
    return 0;  
}
```

# CONTROLE DE FLUXO



- **CONSTRUÇÕES COM LAÇOS – do-while**

```
do  
{  
    bloco de comandos  
} while (expr_booleana);
```

# CONTROLE DE FLUXO



## • CONSTRUÇÕES COM LAÇOS – do-while

### – Exemplo

```
/* Fatorial (versao 3) */  
  
#include <stdio.h>  
  
int main (void)  
{  
    int i;  
    int n;  
    int f = 1;  
  
    /* requisita valor do usuário */  
    do  
    {  
        printf("Digite um valor inteiro nao negativo:");  
        scanf ("%d", &n);  
    } while (n<0);  
  
    /* calcula fatorial */  
    for (i = 1; i <= n; i++)  
        f *= i;  
  
    printf(" Fatorial = %d\n", f);  
    return 0;  
}
```

# CONTROLE DE FLUXO



- **CONSTRUÇÕES COM LAÇOS - for**

```
for (expr_inicial; expr_booleana; expr_de_incremento)  
{  
    bloco de comandos  
    ...  
}
```

# CONTROLE DE FLUXO



- **CONSTRUÇÕES COM LAÇOS - for**

- Exemplo

```
/* Fatorial (versao 2) */  
  
#include <stdio.h>  
  
int main (void)  
{  
    int i;  
    int n;  
    int f = 1;  
  
    printf("Digite um número inteiro nao negativo:");  
    scanf("%d", &n);  
  
    /* calcula fatorial */  
    for (i = 1; i <= n; i++)  
    {  
        f *= i;  
    }  
    printf(" Fatorial = %d \n", f);  
    return 0;  
}
```

# CONTROLE DE FLUXO



- **INTERRUPÇÕES COM break**

– Quando utilizado dentro de um laço, interrompe e termina a execução do mesmo

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

```
int main (void)
{
    int i;
    for (i = 0; i < 10; i++)
    {
        if (i == 5)
            break;
        printf("%d  ", i);
    }
    printf("fim\n");
    return 0;
}
```

Saída:

0 1 2 3 4 fim

# CONTROLE DE FLUXO



- **INTERRUPÇÕES COM continue**

- Interrompe a execução de um laço passando para a próxima iteração

```
#include <stdio.h>

int main (void)
{
    int i;
    for (i = 0; i < 10; i++ )
    {
        if (i == 5) continue;
        printf("%d  ", i);
    }
    printf("fim\n");
    return 0;
}
```

Saída:

0 1 2 3 4 6 7 8 9 fim



# CONTROLE DE FLUXO



- **SELEÇÃO**

- Comando (**switch**) para selecionar um dentre um conjunto de possíveis casos

```
switch ( expr )
{
    case op1:
        ...          /* comandos executados se expr == op1 */
        break;
    case op2:
        ...          /* comandos executados se expr == op2 */
        break;
    case op3:
        ...          /* comandos executados se expr == op3 */
        break;
    default:
        ...          /* executados se expr for diferente de todos */
        break;
}
```

# CONTROLE DE FLUXO



- **SELEÇÃO**

- **opi** deve ser um número inteiro ou uma constante caractere
- Se **expr** resultar no valor **op**, os comandos que se seguem ao caso **op** são executados, até que se encontre um **break**
- O caso **default** representa nenhum dos outros

# CONTROLE DE FLUXO



- **SELEÇÃO**  
— Exemplo

```
/* calculadora de quatro operações */  
  
#include <stdio.h>  
  
int main (void)  
{  
    float num1, num2;  
    char op;  
  
    printf("Digite: numero op numero\n");  
    scanf ("%f %c %f", &num1, &op, &num2);  
    switch (op)  
    {  
        case '+':  
            printf(" = %f\n", num1+num2);  
            break;  
        case '-':  
            printf(" = %f\n", num1-num2);  
            break;  
        case '*':  
            printf(" = %f\n", num1*num2);  
            break;  
        case '/':  
            printf(" = %f\n", num1/num2);  
            break;  
        default:  
            printf("Operador invalido!\n");  
            break;  
    }  
    return 0;  
}
```

# Estrutura de Dados



- **REFERÊNCIAS**
- Tenenbaum, Aaron M. Langsam, Yedidyah, Augenstein, Moshe J. **Estruturas de dados usando C**. São Paulo : MAKRON *Books*, 1995.
- Veloso, Paulo. et. al. **Estrutura de dados**. Rio de Janeiro: Campus, 1986.
- Moraes, Celso Roberto. **Estrutura de dados e algoritmos**. 2. ed. São Paulo: Futura, 2003.
- Celes, W. Rangel, J. L. **Curso de Estrutura de Dados**. PUC-Rio, 2002.
- W. Celes, R. Cerqueira, J.L. Rangel. **Introdução a Estruturas de Dados - com técnicas de programação em C**. Rio de Janeiro: Campus, 2004.