

Estrutura de Controle

Projeto de Certificadora de Competência Comum

André Agostinis
Gabriel Victor Tavares
Leila Minello
Luan Almeida

Professor:
André Luis dos
Santos Domingues

Conteúdo

- **Condicionais**
 - Tipos de condicionais
- **Laços de repetição**
- **Exemplos**
- **Referências**

Definição

de condicionais

As estruturas condicionais permitem que programas tomem decisões com base em condições booleanas (verdadeiras ou falsas), executando diferentes blocos de código conforme o resultado. Elas são essenciais para criar programas dinâmicos e adaptáveis.

As estruturas condicionais controlam o fluxo de execução do programa, permitindo que ele responda de forma diferente a várias entradas ou situações. São essenciais para implementar lógica de decisão no desenvolvimento de software.



Principais tipos:

- if
- if ... else
- if ... else if

Tipos de Condicionais

if: executa um bloco de código se a condição for verdadeira.

```
#include <stdio.h>

int main() {
    int idade = 20;

    if (idade >= 18) {
        printf("Você é maior de idade.\n");
    }

    return 0;
}
```

-> se a variável idade for maior ou igual a 18, a mensagem "Você é maior de idade." é exibida.

if .., else: executa um bloco de código se a condição for verdadeira, e outro bloco se a condição for falsa.

```
#include <stdio.h>

int main() {
    int nota = 75;

    if (nota >= 60) {
        printf("Você foi aprovado!\n");
    } else {
        printf("Você foi reprovado!\n");
    }

    return 0;
}
```

-> se a nota é maior ou igual a 60, o estudante é aprovado. Caso contrário, é reprovado.

Tipos de Condicionais

if .., else if: permite testar várias condições, executando diferentes blocos de instrução conforme necessário.

```
#include <stdio.h>

int main() {
    int hora = 14;

    if (hora < 12) {
        printf("Bom dia!\n");
    } else if (hora < 18) {

        printf("Boa tarde!\n");
    } else {
        printf("Boa noite!\n");
    }

    return 0;
}
```

-> dependendo do valor de hora, a mensagem de saudação é exibida.

Laços de Repetição

Os laços de repetição, também conhecidos como loops, permitem executar um bloco de código repetidamente enquanto uma determinada condição é verdadeira.

for: um laço controlado com um número fixo de iterações

```
#include <stdio.h>

int main() {
    for (int i = 0; i < 5; i++) {
        printf("Iteração %d\n", i);
    }

    return 0;
}
```

-> a mensagem "Iteração 0" a "Iteração 4" será exibida cinco vezes

Laços de Repetição

while: executa um bloco de código enquanto a condição especificada for verdadeira.

```
#include <stdio.h>

int main() {
    int i = 0;

    while (i < 5) {
        printf("Valor de i: %d\n", i);
        i++;
    }

    return 0;
}
```

-> o loop é executado enquanto i é menor que 5.

do...while: similar ao while, mas garante que o bloco de código seja executado pelo menos uma vez.

```
#include <stdio.h>

int main() {
    int i = 0;

    do {
        printf("Valor de i: %d\n", i);
        i++;
    } while (i < 5);

    return 0;
}
```

-> o também exibe "Valor de i" de 0 a 4, mas sempre pelo menos uma vez

Exemplo prático

1 - Calculando a soma de números pares

Este exemplo mostra o uso
de um laço for para somar
números pares de 1 a 10.

```
#include <stdio.h>

int main() {
    int soma = 0;

    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) { // Se i é par
            soma += i;
        }
    }

    printf("A soma dos números pares de 1 a 10 é: %d\n", soma);

    return 0;
}
```


Exemplo prático

2 - Verificação de nota

Neste exemplo, usamos uma estrutura de condição if...else para verificar se a nota do aluno é suficiente para ser aprovado.

```
#include <stdio.h>

int main() {
    int nota;

    printf("Digite a nota do aluno: ");
    scanf("%d", &nota);

    if (nota >= 50) {
        printf("Aluno aprovado!\n");
    } else {
        printf("Aluno reprovado!\n");
    }

    return 0;
}
```

Referências

KERNIGHAN, B. W.; RITCHIE, D. M.

The C Programming Language. 2. ed. Englewood
Cliffs: Prentice Hall, 1988.