

Estrutura de Decisão

Informática – IFSULDEMINAS

Primeiro Semestre de 2013

Roteiro

1 Introdução

2 If - else

3 Switch

Estrutura de Decisão

- Uma grande quantidade de programas podem ser escritos usando estrutura linear.
- Existem situações que testes são necessários durante a execução e dependendo do resultado do teste uma atitude ou outra é tomada – **Estrutura de Decisão ou Estrutura Condicional.**



Estrutura de Decisão

se ... senão

se (condição) então faça instrução 1 senão faça instrução 2

- A **condição** assume valores booleanos, **verdadeiro** ou **falso**.
- Quando a condição for verdadeira, a instrução 1 será executada.
- Quando a condição for falsa, a instrução 2 será executada.

Estrutura de Decisão

se ... senão

se (condição) então faça instrução 1 senão faça instrução 2

- A **condição** assume valores booleanos, **verdadeiro** ou **falso**.
 - Quando a condição for verdadeira, a instrução 1 será executada.
 - Quando a condição for falsa, a instrução 2 será executada.

Estrutura de Decisão

se ... senão

se (condição) então faça instrução 1 senão faça instrução 2

- A **condição** assume valores booleanos, **verdadeiro** ou **falso**.
- Quando a condição for verdadeira, a instrução 1 será executada.
- Quando a condição for falsa, a instrução 2 será executada.

Estrutura de Decisão

se ... senão

se (condição) então faça instrução 1 senão faça instrução 2

- A **condição** assume valores booleanos, **verdadeiro** ou **falso**.
- Quando a condição for verdadeira, a instrução 1 será executada.
- Quando a condição for falsa, a instrução 2 será executada.

Operadores Relacionais

- Os operadores relacionais são usados nas condições das estruturas de decisão.

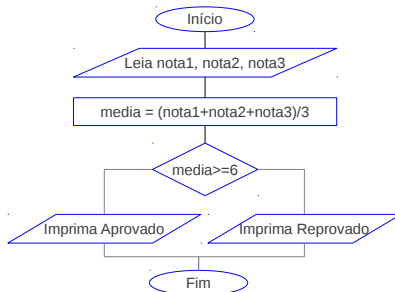
| Operadores | Significado |
|------------|----------------|
| > | Maior |
| < | Menor |
| >= | Maior ou Igual |
| <= | Menor ou Igual |
| == | Igual |
| != | Diferente |

If - else

se ... senão

```
if (condição)
    instrução 1;
[else instrução 2;]
```

- **Exemplo:** dadas três notas de um aluno, determine se o aluno foi ou não aprovado. A média para aprovação é 6.0.



If - else

Aprovado/Reprovado

```
float nota1, nota2, nota3, media;  
scanf("%f%f%f", &nota1, &nota2, &nota3);  
media = (nota1 + nota2 + nota3)/3;  
if (media >= 6)  
    printf("O aluno foi aprovado.");  
else  
    printf("O aluno não foi aprovado.");
```

- Quando temos mais de uma instrução nas cláusulas **if** e **else** devemos usar **{** e **}** para marcar o início e o fim do bloco.

Aprovado/Reprovado

```
if (media >= 6) {  
    printf("O aluno foi aprovado.");  
    printf("Media: %f", media);  
}
```

If - else

Aprovado/Reprovado

```
float nota1, nota2, nota3, media;  
scanf("%f%f%f", &nota1, &nota2, &nota3);  
media = (nota1 + nota2 + nota3)/3;  
if (media >= 6)  
    printf("O aluno foi aprovado.");  
else  
    printf("O aluno não foi aprovado.");
```

- Quando temos mais de uma instrução nas cláusulas **if** e **else** devemos usar { e } para marcar o início e o fim do bloco.

Aprovado/Reprovado

```
if (media >= 6) {  
    printf("O aluno foi aprovado.");  
    printf("Media: %f", media);  
}
```

If - else

Aprovado/Reprovado

```
float nota1, nota2, nota3, media;
scanf("%f%f%f", &nota1, &nota2, &nota3);
media = (nota1 + nota2 + nota3)/3;
if (media >= 6)
    printf("O aluno foi aprovado.");
else
    printf("O aluno não foi aprovado.");
```

- Quando temos mais de uma instrução nas cláusulas **if** e **else** devemos usar { e } para marcar o início e o fim do bloco.

Aprovado/Reprovado

```
if (media >= 6) {
    printf("O aluno foi aprovado.");
    printf("Media: %f", media);
}
```

If - else - if encaixados

- Há situações que precisamos testar várias alternativas.
- Podemos usar vários **ifs**.
- Ou ... um encadeamento de **if - else - if**.
- **Exemplo:** dado um inteiro *día*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;  
  
scanf("%d", &dia);  
  
if (dia == 1) {  
    printf("Domingo.");  
} else if (dia == 2) {  
    printf("Segunda.");  
} else if (dia == 3) {  
    ...  
}
```

If - else - if encaixados

- Há situações que precisamos testar várias alternativas.
- Podemos usar vários **ifs**.
 - Ou ... um encadeamento de **if - else - if**.
 - **Exemplo:** dado um inteiro *día*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;  
  
scanf("%d", &dia);  
  
if (dia == 1) {  
    printf("Domingo.");  
} else if (dia == 2) {  
    printf("Segunda.");  
} else if (dia == 3) {  
    ...  
}
```

If - else - if encaixados

- Há situações que precisamos testar várias alternativas.
- Podemos usar vários **ifs**.
- Ou ... um encadeamento de **if - else - if**.
 - **Exemplo:** dado um inteiro *día*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;  
  
scanf("%d", &dia);  
  
if (dia == 1) {  
    printf("Domingo.");  
} else if (dia == 2) {  
    printf("Segunda.");  
} else if (dia == 3) {  
    ...  
}
```

If - else - if encaixados

- Há situações que precisamos testar várias alternativas.
- Podemos usar vários **ifs**.
- Ou ... um encadeamento de **if - else - if**.
- **Exemplo:** dado um inteiro *dia*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;  
  
scanf("%d", &dia);  
  
if (dia == 1) {  
    printf("Domingo.");  
} else if (dia == 2) {  
    printf("Segunda.");  
} else if (dia == 3) {  
    ...  
}
```


If - else - if encaixados

- Há situações que precisamos testar várias alternativas.
- Podemos usar vários **ifs**.
- Ou ... um encadeamento de **if - else - if**.
- **Exemplo:** dado um inteiro *dia*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;

scanf("%d", &dia);

if (dia == 1) {
    printf("Domingo.");
} else if (dia == 2) {
    printf("Segunda.");
} else if (dia == 3) {
    ...
}
```

Operadores Lógicos

| Operadores | Significado |
|------------|-------------|
| && | e |
| | ou |
| ! | negação |

- Os operadores lógicos são usados nas condições das estruturas de decisão.
- A expressão **(condição1) && (condição2)** é avaliada como verdadeira quando ambas as condições são verdadeiras.
- A expressão **(condição1) || (condição2)** é avaliada como verdadeira quando pelo menos uma das condição é verdadeira.
- A expressão **!(condição)** é a negação da condição, é avaliada como verdadeira quando a condição é falsa.

Operadores Lógicos

| Operadores | Significado |
|------------|-------------|
| && | e |
| | ou |
| ! | negação |

- Os operadores lógicos são usados nas condições das estruturas de decisão.
- A expressão **(condição1) && (condição2)** é avaliada como verdadeira quando ambas as condições são verdadeiras.
- A expressão **(condição1) || (condição2)** é avaliada como verdadeira quando pelo menos uma das condição é verdadeira.
- A expressão **!(condição)** é a negação da condição, é avaliada como verdadeira quando a condição é falsa.

Operadores Lógicos

| Operadores | Significado |
|------------|-------------|
| && | e |
| | ou |
| ! | negação |

- Os operadores lógicos são usados nas condições das estruturas de decisão.
- A expressão **(condição1) && (condição2)** é avaliada como verdadeira quando ambas as condições são verdadeiras.
- A expressão **(condição1) || (condição2)** é avaliada como verdadeira quando pelo menos uma das condição é verdadeira.
- A expressão **!(condição)** é a negação da condição, é avaliada como verdadeira quando a condição é falsa.

Operadores Lógicos

| Operadores | Significado |
|------------|-------------|
| && | e |
| | ou |
| ! | negação |

- Os operadores lógicos são usados nas condições das estruturas de decisão.
- A expressão **(condição1) && (condição2)** é avaliada como verdadeira quando ambas as condições são verdadeiras.
- A expressão **(condição1) || (condição2)** é avaliada como verdadeira quando pelo menos uma das condição é verdadeira.
- A expressão **!(condição)** é a negação da condição, é avaliada como verdadeira quando a condição é falsa.

Operadores Lógicos

| Operadores | Significado |
|------------|-------------|
| && | e |
| | ou |
| ! | negação |

- Os operadores lógicos são usados nas condições das estruturas de decisão.
- A expressão **(condição1) && (condição2)** é avaliada como verdadeira quando ambas as condições são verdadeiras.
- A expressão **(condição1) || (condição2)** é avaliada como verdadeira quando pelo menos uma das condição é verdadeira.
- A expressão **!(condição)** é a negação da condição, é avaliada como verdadeira quando a condição é falsa.

Combinando Condicionais

Às vezes podemos simplificar

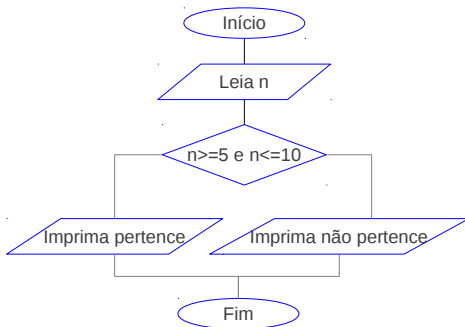
```
if (condição1) {  
    if (condição2) {  
        if (condição3) {  
            ...  
        }  
    }  
}
```

por

```
if (condição1 && condição2 && condição3) {  
    ...  
}
```

Combinando Condicionais

- **Exemplo:** dado um número n , verifique se ele pertence ao intervalo $5 \leq n \leq 10$.



Combinando Condicionais

$5 \leq n \leq 10$?

```
int n;
```

```
scanf("%d", &n);
```

```
if (n >= 5 && n <= 10)
```

```
    printf("%d está no intervalo.", n);
```

```
else
```

```
    printf("%d não está no intervalo.", n);
```

Switch

- Muitas vezes temos que fazer testes sobre uma **mesma variável** e a partir daí tomar decisões.
- O comando switch permite múltiplas decisões sobre a mesma variável.
- Pode ser aplicado à variáveis dos tipos **int** e **char**.

switch

```
switch (variável) {  
    case valor1:  
        comandos  
        break;  
    case valor2:  
        comandos  
        break;  
    default:  
        comandos  
}
```

Switch

- Muitas vezes temos que fazer testes sobre uma **mesma variável** e a partir daí tomar decisões.
- O comando switch permite múltiplas decisões sobre a mesma variável.
- Pode ser aplicado à variáveis dos tipos **int** e **char**.

switch

```
switch (variável) {  
    case valor1:  
        comandos  
        break;  
    case valor2:  
        comandos  
        break;  
    default:  
        comandos  
}
```

Switch

- Muitas vezes temos que fazer testes sobre uma **mesma variável** e a partir daí tomar decisões.
- O comando switch permite múltiplas decisões sobre a mesma variável.
- Pode ser aplicado à variáveis dos tipos **int** e **char**.

switch

```
switch (variável) {  
    case valor1:  
        comandos  
        break;  
    case valor2:  
        comandos  
        break;  
    default:  
        comandos  
}
```

Switch

- Muitas vezes temos que fazer testes sobre uma **mesma variável** e a partir daí tomar decisões.
- O comando switch permite múltiplas decisões sobre a mesma variável.
- Pode ser aplicado à variáveis dos tipos **int** e **char**.

switch

```
switch (variável) {  
    case valor1:  
        comandos  
        break;  
    case valor2:  
        comandos  
        break;  
    default:  
        comandos  
}
```

Switch

- **Exemplo:** dado um inteiro *dia*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;

scanf("%d", &dia);

switch(dia) {
    case 1:
        print(Domingo!);
        break;
    case 2:
        print(Segunda!);
        break;
    ...
    default:
        printf("Dia inválido!");
}
```

Switch

- **Exemplo:** dado um inteiro *dia*, determine o correspondente dia da semana.

Dia da Semana

```
int dia;

scanf("%d", &dia);

switch(dia) {
    case 1:
        print(Domingo!);
        break;
    case 2:
        print(Segunda!);
        break;
    ...
    default:
        printf("Dia inválido!");
}
```

Switch

- O comando começa a ser executado a partir do ponto em que identifica o valor da variável.
- Continua executando até que o comando **break** ou o final do comando é encontrado.
- A condição **default** pode ser utilizada quando nenhuma das opções anteriores é satisfeita.

Switch

- O comando começa a ser executado a partir do ponto em que identifica o valor da variável.
- Continua executando até que o comando **break** ou o final do comando é encontrado.
- A condição **default** pode ser utilizada quando nenhuma das opções anteriores é satisfeita.

Switch

- O comando começa a ser executado a partir do ponto em que identifica o valor da variável.
- Continua executando até que o comando **break** ou o final do comando é encontrado.
- A condição **default** pode ser utilizada quando nenhuma das opções anteriores é satisfeita.