

Estruturas de Controle em Linguagem C

Introdução à Ciência da Computação

Rosane Minghim

Apoio na confecção: Carlos Elias A. Zampieri

Danilo Medeiros Eler

Rogério Eduardo Garcia

Renato Rodrigues

Blocos de Comandos

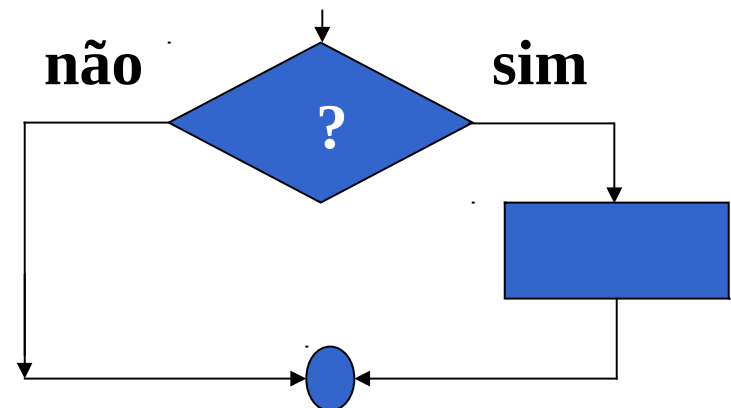
- Os delimitadores `{` e `}` são necessários toda vez que existe mais de um comando num bloco

```
{  
    comandos;  
}
```

Escolha Simples

- Pode-se selecionar a sequência de comandos a ser executada
- Formato:

```
if (condição)  
{  
    Comandos;  
}
```





Exemplo de Escolha Simples

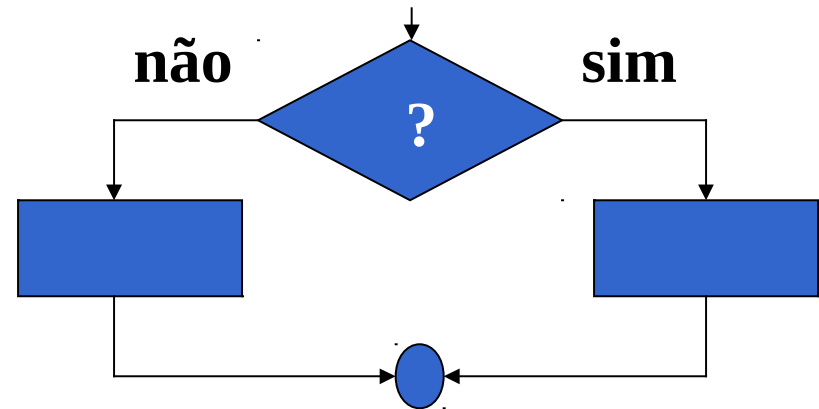
```
if (empregado == diarista)
    salario = salario_base + dias_trabalhados
        * valor_do_dia;
```

```
if (empregado == diarista)
{
    diarista = diarista + 1;
    adicional = dias_trabalhados * valor_do_dia;
    salario = salario_base + adicional;
}
```

Escolha Composta

- Formato:

```
if (condição)  
    comandos;  
else  
    comandos;
```





Exemplo

```
if (empregado == diarista)
{
    salario = salario_base +
        dias_trabalhados * valor_do_dia;
}
else
{
    complemento = fator*complemento_basico;
    salario = salario_base + complemento;
}
```

Regras Básicas

- Usar sempre '{', '}' e ';' mesmo que seja redundante para definir os comandos a serem executados dentro da escolha e fim de comando
- Atenção com os ';' (ponto e vírgula) ao programar
- Sempre abra e feche as chaves ('{' e '}') ao iniciar um bloco de comandos, isto evita esquecer de adicioná-los

```
if (empregado == diarista)
{
    salario = salario_base + dias_trabalhados
                * valor_do_dia;
}
else
{
    adicional = dias_trabalhados * valor_do_dia;
    salario = salario_base + adicional;
}
```

Exemplo

```
#include <stdio.h>

int main()
{
    int numero1, numero2, numero3, maior;

    printf("Digite o numero 1: ");
    scanf("%d",&numero1);
    printf("Digite o numero 2: ");
    scanf("%d",&numero2);
    printf("Digite o numero 3: ");
    scanf("%d",&numero3);

    if (numero1>numero2)
    {
        if ((numero2>numero3)|| (numero2==numero3))
        {
            maior = numero1;
        }
        else
        {
            if (numero1>numero3)
            {
                maior = numero1;
            }
            else
            {
                maior = numero3;
            }
        }
    }
}
```

```
else
{
    if ((numero2>numero3)|| (numero2==numero3))
    {
        maior = numero2;
    }
    else
    {
        maior = numero3;
    }
}
printf("Maior: %d\n",maior);
return 0;
}
```


Exemplo - Simplificado

```
#include <stdio.h>

int main()
{
    int numero1, numero2, numero3, maior;

    printf("Digite o numero 1: ");
    scanf("%d",&numero1);
    printf("Digite o numero 2: ");
    scanf("%d",&numero2);
    printf("Digite o numero 3: ");
    scanf("%d",&numero3);

    maior = numero1;
    if (numero2 > maior)
    {
        maior = numero2;
    }
    if (numero3 > maior)
    {
        maior = numero3;
    }

    printf("Maior: %d\n",maior);
    return 0;
}
```



Escolhas Múltiplas

- Permite escolher uma entre várias alternativas
- Formato:

```
switch (expressão)
}
  case constante1:
    comandos;
    break;
  case constante2:
    comandos;
    break;
  ...
  case constanteN:
    comandos;
    break;
  default:
    comandos;
}
```

Exemplo

```
#include <stdio.h>

#define FATOR_CATEGORIA_1 1.0
#define FATOR_CATEGORIA_2 0.8
#define FATOR_CATEGORIA_3 0.6
#define FATOR_GERAL 0.2
#define TAXA_BASICA 20.00

int main()
{
    int categoria;
    float taxa;

    scanf("%d",&categoria);

    taxa = TAXA_BASICA;
```

```
    switch (categoria)
    {
        case 1:
            taxa = taxa - (TAXA_BASICA *
                FATOR_CATEGORIA_1);
            break;
        case 2:
            taxa = taxa - (TAXA_BASICA *
                FATOR_CATEGORIA_2);
            break;
        case 3:
            taxa = taxa - (TAXA_BASICA *
                FATOR_CATEGORIA_3);
            break;
        default:
            taxa = taxa - (TAXA_BASICA *
                FATOR_GERAL);
    }

    printf("%.3f\n", taxa);
}
```



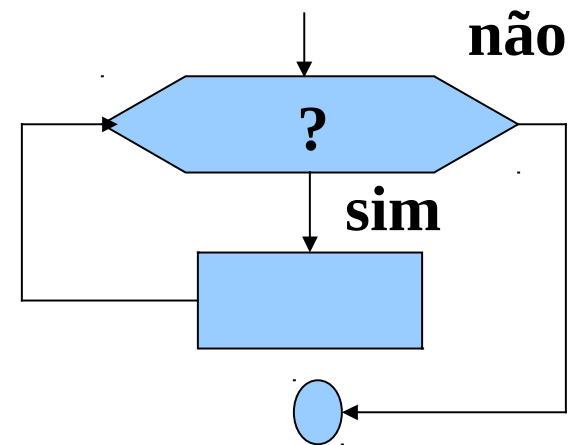
Preocupações:

- 1 - Maior – sempre fazer o algoritmo antes
- 2 - Organização do código
 - nomes apropriados de variáveis e constantes
 - definir, adotar e manter um padrão pra o nome de variáveis
 - inicializações devidas
 - comentários adequados
- 3 - Diálogo com o usuário
 - mensagens adequadas de entrada e saída
 - formatação que facilite a leitura
 - cuidados com maiúsculas e minúsculas, além de outros caracteres ambíguos.
- 4 - Checagem de erros

Repetição por Condição

- Uma das formas de repetir um conjunto de comandos de um algoritmo é subordiná-los a um comando de repetição usando uma estrutura da forma:

```
while (condição)
{
    bloco de comandos;
}
```



Exemplo

Algoritmo calcula_senos

variável

n, i: inteiro

leia(n)

i ← 0

enquanto i ≤ n faça

 escreva(seno(i))

 i ← i + 1

fim enquanto

fim

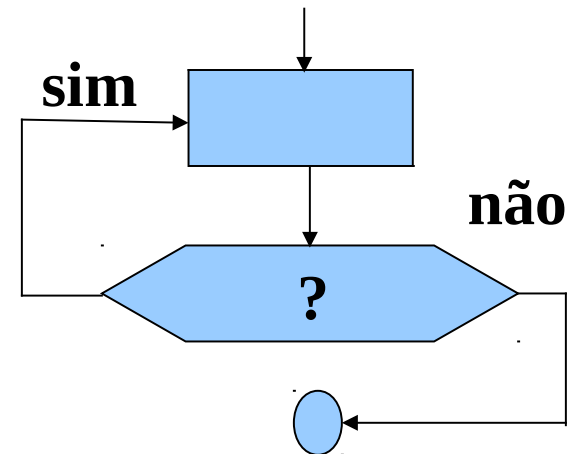
```
#include <stdio.h>
#include <math.h>

int main()
{
    int n,i;
    float angulo;
    scanf("%d",&n);
    i=0;
    while (i <= n)
    {
        //Converte angulo i para
        //radianos
        angulo = i * M_PI / 180;
        printf("seno de %.1f graus=
                %.2f\n",angulo,sin(angulo));
        i = i + 1;
    }
    return 0;
}
```

Repetição por Condição 2

- Outra forma de repetir um conjunto de comandos de um algoritmo, muito parecida a estrutura enquanto, é utilizando o seguinte comando de repetição:

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



Observe que nesta estrutura primeiro é executado o bloco de comandos e somente após é feito o teste condicional!

Exemplo

Algoritmo calcula_senos

variável

n, i: inteiro

leia(n)

i ← 0

repita

escreva(seno(i))

 i ← i + 1

até (i ≤ n)

fim

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

int main()
{
    int n,i;
    float angulo;
    scanf("%d",&n);
    i=0;
    do
    {
        //Converte angulo i para
        //radianos
        angulo = i * M_PI / 180;
        printf("seno de %.1f graus=
            %.2f\n",angulo,sin(angulo));
        i = i + 1;
    } while (i <= n);
    return 0;
}
```




Repetição por Contagem

- Na iteração baseada em contagem, sabe-se **antecipadamente** quantas vezes um conjunto de comandos vai ser repetido.
- Formato:

```
int var;  
for (var=valor_inicial;var<valor_final;var+=valor_do_passo)  
{  
    bloco de comandos;  
}
```



Exemplo 1

```
#include <stdio.h>

int main()
{
    int n,i;
    printf("Digite o final da contagem: ");
    scanf("%d",&n);

    for (i=1;i<=n;i++)
    {
        printf("%d\n",i);
    }
    return 0;
}
```



Exemplo 2

```
#include <stdio.h>

int main()
{
    int i,n;

    printf("Conta decrescente\n");
    printf("Digite o valor do inicio da contagem: ");
    scanf("%d",&n);

    for (i=n;i>=1;i--)
    {
        printf("%d\n",i);
    }

    return 0;
}
```



Exemplo 3

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

```
int main()
```

```
{
```

```
    int i,n,meio;
```

```
    printf("Digite o final da contagem: ");
```

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

```
    meio = n/2;
```

```
    for (i=1;i<=meio;i++)
```

```
    {
```

```
        printf("%d\n",i*2-1);
```

```
    }
```

```
    return 0;
```

```
}
```



Mais exemplos

- Nos códigos-exemplo que acompanham esses slides
- Tarefa:
 - Estudar, reproduzir e modificar os códigos exemplo que acompanham esses slides.
- **FIM DOS SLIDES**