Aula 16: Repetição (Parte 2)

Introdução a Programação

Túlio Toffolo & Puca Huachi
http://www.toffolo.com.br

BCC201 – 2019/2
Departamento de Computação – UFOP

Aula Anterior

- Laços de repetição
- Comando de repetição `while`
- Exercícios

Aula de Hoje

1. Comando `do-while`
2. Exemplos
3. Uso do laço

Aula de Hoje

1. Comando `do-while`
2. Exemplos
3. Uso do laço

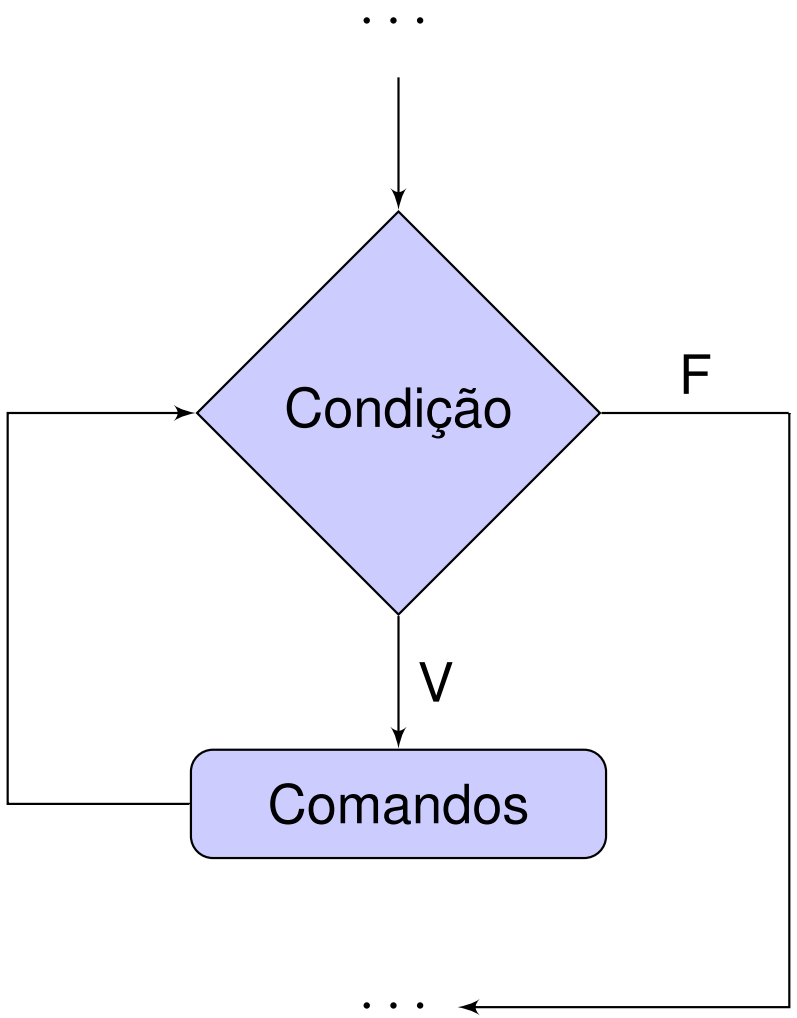
Laços ou Repetições (*loop*)

- Laços são comandos usados sempre que uma ou mais instruções devam ser repetidas enquanto um certa condição estiver sendo satisfeita
- Laços em C++
  - `while`
  - `do - while`
  - `for`

Laços ou Repetições (*loop*)

- Em um laço **controlado logicamente**, os comandos (corpo do laço) são repetidos enquanto uma **expressão lógica for verdadeira**
  - `while`
  - `do - while`
- Em um laço **controlado por contador**, os comandos (corpo do laço) são repetidos um **número predeterminado de vezes**
  - `for`
- Denomina-se **iteração** a repetição de um conjunto de comandos: cada execução do corpo do laço, juntamente com a condição de terminação do laço, é uma iteração

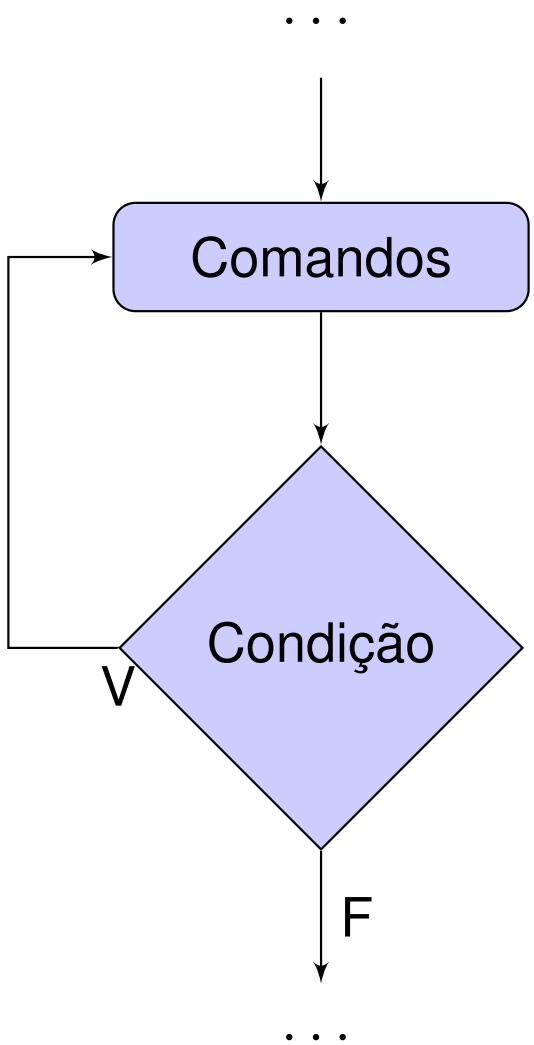
Laço `while`



Laço `while`

```
1  while (<expressão>)
2  {
3      <comando_1>;
4      ...
5      <comando_n>;
6  }
```

Laço `do-while`



Comando `do-while`

```
1  do {
2      <comando_1>;
3      ...
4      <comando_n>;
5  } while (<expressão>);
```

No comando `do-while`, ao contrário do comando `while`, o teste do laço/loop está no final, isso significa que os `<comandos>` serão executados **no mínimo uma vez**.

Aula de Hoje

Exemplo 1

Faça um programa que leia as notas da primeira prova de BCC201 e calcule e imprima a média das notas.

Considere que o número de alunos é desconhecido. Utilize como critério de parada do programa uma nota negativa.

Exemplo 1 (usando `while`)

```c
#include <stdio.h>

int main()
{
    int alunos = 0;
    double nota, soma = 0;

    printf("Digite um nro negativo para sair ou a nota do aluno: ");
    scanf("%lf", &nota);

    while (nota >= 0) {
        soma += nota;
        alunos++;
        printf("Digite um nro negativo para sair ou a nota do aluno: ");
        scanf("%lf", &nota);
    }
    double media = soma / alunos;
    printf("A média das notas é: %.2lf\n", soma);

    return 0;
}
```

Exemplo 1 (usando `do-while`)

```c
#include <stdio.h>

int main()
{
    int alunos = 0;
    double nota, soma = 0;

    do {
        printf("Digite um nro negativo para sair ou a nota do aluno: ");
        scanf("%lf", &nota);
        if (nota >= 0) {
            soma += nota;
            alunos++;
        }
    } while(nota >= 0);

    double media = soma / alunos;
    printf("A média das notas é: %.2lf\n", soma);

    return 0;
}
```

Exemplo 2: Jogo de Adivinhar

Faça um jogo onde o usuário tem que acertar um número selecionada pelo programa. O jogo termina quando o número digitado estiver correto. Imprimir o número de tentativas até o acerto.

Exemplo 2: Jogo de Adivinhar

```c
// Jogo adivinha usando do-while
#include <stdio.h>

int main()
{
    int resp, adivinha = 10;
    int tentativas = 0;

    do {
        printf("Digite um nro: ");
        scanf("%d", &resp);
        if (resp != adivinha)
            printf("%d é incorreto, tente novamente!\n\n", resp);
        tentativas++;
    } while (resp != adivinha);

    printf("%d  é o nro CORRETO!\n", resp);
    printf("Número de tentativas: %d\n", tentativas);
    printf("FIM DO JOGO...\n");
    return 0;
}
```

Exemplo 2: Jogo de Adivinhar

```c
// Jogo adivinha usando while
#include <stdio.h>

int main()
{
    int resp, adivinha = 10;
    int tentativas = 1;

    printf("Digite um nro: ");
    scanf("%d", &resp);

    while (resp != adivinha); {
        printf("Digite um nro: ");
        scanf("%d", &resp);
        if (resp != adivinha)
            printf("%d é incorreto, tente novamente!\n\n", resp);
        tentativas++;
    }
    printf("%d  é o nro CORRETO!\n", resp);
    printf("Número de tentativas: %d\n", tentativas);
    printf("FIM DO JOGO...\n");
    return 0;
}
```

Exemplo 3: Jogo de Adivinhar

E se o usuário quiser jogar mais de uma vez?

```c
#include <stdio.h>
#include <stdlib.h> // para incluir a função rand

int main()
{
    int tentativas, adivinha, resp, novamente;

    do {
        tentativas = 0;
        adivinha = rand() % 10;

        do {
            printf("Digite um nro: ");
            scanf("%d", &resp);
            if (resp != adivinha)
                printf("%d é incorreto, tente novamente!\n\n", resp);
            tentativas++;
        } while (resp != adivinha);

        printf("%d  é o nro CORRETO!\n", resp);
        printf("Número de tentativas: %d\n", tentativas);
        printf("Deseja jogar novamente? (0=NAO, 1=SIM)\n");
        scanf("%d", &novamente);
    } while (novamente == 1);

    return 0;
}
```

Aula de Hoje

1. Comando `do-while`
2. Exemplos
3. Uso do laço

Validação de entrada

Exemplo:

1. O usuário deve digitar um valor inteiro, positivo e par.
2. O programa continua após a digitação de um valor que atende essas restrições.
3. Observe que o usuário pode errar na digitação do valor..

Deve ser utilizado o `if`, `while` ou `do-while`?

Validando entrada

```
1   ...
2   printf("\nDigite um valor: ");
3   scanf("%d", &x);
4
5   while (x <= 0 || x % 2 != 0)
6   {
7       printf("\nERRO: Valor Inválido !");
8       printf("\nDigite um valor: ");
9       scanf("%d", &x);
10  }
11
12  // o programa prossegue com um número que
13  // atende a restrição
```

Observação:

- Não se pode prever quantas vezes o usuário entrará com um valor incorreto;
- O comando `if` não nos atende neste caso...

Validando entrada

```
1  ...
2  do {
3      printf("\nDigite um valor: ");
4      scanf("%d", &x);
5
6      if (x <= 0 || x % 2 != 0)
7          printf("\nERRO: Valor Inválido !");
8  } while (x <= 0 || x % 2 != 0);
9
10 // o programa prossegue com um número que
11 // atende a restrição
```

Observação:

- Não se pode prever quantas vezes o usuário entrará com um valor incorreto;
- O comando `if` não nos atende neste caso...

Repetição do programa

O comando `do-while` é útil para implementar repetição do programa:

```
1  char op;
2  do  {
3
4      /* o código do programa entra aqui... */
5
6      printf("\nRepetir a execução (s/n)?");
7      scanf("%d", &op);
8  } while (op == 's' || op == 'S') // o programa aceita S maiúsculo tb
```

Em geral, o comando `do-while` é usado quando os `<comandos>` devem ser executados **no mínimo uma vez**.

Exemplo 4: Algoritmo de Euclides

Calcula o MDC (máximo divisor comum) entre dois inteiros.
Efetuar várias divisões até chegar a uma divisão exata. O divisor desta divisão é o MDC. Ex.: cálculo do mdc(48,30). Regra prática:

1º) dividimos o número maior pelo número menor:
48 / **30** = 1 (com resto **18**)

2º) dividimos o divisor 30, que é divisor da divisão anterior, por 18, que é o resto da divisão, e assim sucessivamente:
30 / **18** = 1 (com resto **12**)
18 / **12** = 1 (com resto **6**)
12 / **6** = 2 (com resto **zero** - divisão exata)

3º) O divisor da divisão exata é 6.

Então **mdc(48, 30) = 6**.

Exemplo 4: Algoritmo de Euclides

Deve ser utilizado `do-while` ou `while` para calcular o MDC?

Exemplo 4: Algoritmo de Euclides (I)

```c
/*
* Calcula o MDC (máximo divisor comum) entre dois inteiros.
*/
int mdc(int x, int y)
{
    // criando variável para armazenar o resto
    int resto;

    // garantindo que x >= y
    if (x < y)
        troca(&x, &y);   // função que troca os valores de x e y

    resto = x % y;       // calcula o resto
    while (resto != 0) {
        x = y;           // x é atualizado
        y = resto;       // y armazena o MDC
        resto = x % y;   // recalcula o resto
    }

    return y;
}
```

Exemplo 4: Algoritmo de Euclides (II)

```c
/*
* Calcula o MDC (máximo divisor comum) entre dois inteiros.
*/
int mdc(int x, int y)
{
    // criando variável para armazenar o resto
    int resto;

    // garantindo que x >= y
    if (x < y)
        troca(&x, &y);   // função que troca os valores de x e y

    do {
        resto = x % y;   // calcula o resto
        x = y;           // x armazena o MDC
        y = resto;       // atualiza o valor de y para o resto
    } while (resto != 0);

    return x;
}
```

Exemplo 4: Algoritmo de Euclides

Usando `while`

```c
int mdc(int x, int y)
{
    if (x < y)
        troca(&x, &y);

    int resto = x % y;

    while (resto != 0) {
        x = y;
        y = resto;
        resto = x % y;
    }

    return y;
}
```

Usando `do-while`

```c
int mdc(int x, int y)
{
    if (x < y)
        troca(&x, &y);

    int resto;

    do {
        resto = x % y;
        x = y;
        y = resto;
    } while (resto != 0);

    return x;
}
```

Perguntas?