Universidade Federal de Ouro Preto - UFOP
Departamento de Computação - DECOM
Programação de Computadores I – BCC701

Aula Teórica 07

Material Didático Proposto

Conteúdos da Aula

- **Laço de Repetição while**

- **Para permitir que uma operação seja executada repetidas vezes utiliza-se comandos de repetição;**
- **Uma estrutura deste tipo também é chamada de laço (*loop* em inglês);**
- **No Scilab, são definidos dois comandos de repetição:**
  1. **Laço controlado por contador : for (para)**
  2. **Laço controlado logicamente: while (enquanto)**

Em um laço controlado por contador, ou *laço contado*, os comandos contidos no corpo do laço são repetidos um número predeterminado de vezes.

Sabe-se de antemão o número de vezes que o laço será repetido

No laço controlado logicamente, ou *laço indeterminado*, os comandos no corpo do laço são repetidos enquanto uma expressão lógica for verdadeira.

Não se saber de antemão quantas vezes será repetido o laço

Denomina-se iteração a repetição de um conjunto de comandos:

- **Cada execução do corpo do laço, juntamente com a avaliação da condição de terminação do laço, é uma iteração.**

O comando **for** pode ser definido da seguinte forma:

```
for variável = <início>:<passo>:<fim>
      <conjunto de comandos>
end
```

- **<conjunto de comandos>** é o conjunto de instruções a serem executadas, é denominado **corpo do laço**.
- **variável = <início>:<passo>:<fim>**
  - <variável> recebe <início>.
  - Ao final de cada iteração, o valor da <variável> é incrementada na quantidade <passo>.
  - O laço termina quando o valor na <variável> for maior do que <fim>.
- se <passo> = 1, então ele pode ser omitido **var = <ini>:<fim>**
- **for** e **end** são palavras reservadas da linguagem.

Faça um programa para calcular a soma dos números naturais até um dado valor k dado pelo usuário.

Posteriormente mostrar a soma.

```
k = input("Digite o valor limite: ");
soma = 0;          // limpando variável acumuladora
for num = 0:k   // mesmo que num = 0:1:k
   soma = soma + num;// acumulando em soma
end // fim do laço
printf("Soma dos %g números naturais: %g", k,
soma);
```

Saída:

```
Digite o valor limite: 10
Soma dos 10 números naturais: 55
```

Considere que temos diferentes turmas, cada uma com um dado número de alunos. Devemos fazer um programa para cada turma, visto que cada uma tem o <fim> do laço diferente? Não, basta ler o total de alunos no início.

```
soma = 0;
cont = 0
// entrando com o total de alunos da turma
tot_alu = input("Entre com o total de alunos")
for alu = 1:tot_alu // usando tot_alu como <fim>
     printf("Aluno: %g", alu)
     nota = input("digite a nota: ")
     soma = soma + nota;
     if nota < 6.0 then
          cont = cont + 1;
     end
end
media = soma/tot_alu
printf("A média da turma é %g.", media);
printf("%g alunos obtiveram nota abaixo da...
        ... média", cont);
```

Instrução de Repetição - Sintaxe para o `while`

- O comando **while** é um laço controlado logicamente;
- O laço **while** é definido da seguinte forma:

```
while <expressão lógica o.s. condição>
   <conjunto de comandos>
end
```

- **<conjunto de comandos>** é o conjunto de instruções a serem executadas, é denominado corpo do laço;
- **<expressão lógica>** enquanto a expressão for verdadeira, o <conjunto de comandos> serão executados.

Obs. Deve haver alguma instrução no corpo do while que permite modificar a condição. Caso contrário o laço será %INF!

Instrução de Repetição - Exemplo 1 - `while`

Elabore um programa para gerar e imprimir os números Naturais até um dado número k:

```
k = input("Digite o valor limite");
num = 0; // inicialização fora do laço
while num <= k
    printf("%g    ", num)
    num = num + 1; // incremento dentro do laço
end
```

Saída:

```
Digite o valor limite: 7
0   1   2   3   4   5   6   7
```

Instrução de Repetição - Exemplo 2 - `while`

Elabore um programa para calcular a soma dos números naturais até um dado número k:

```
k = input("Digite o valor limite");
num = 0;
soma = 0;
while num <= k
   soma = soma + num;
   num = num + 1;
end
printf("A soma dos naturais até %g é igual a
%g", k, soma);
```

Saída:

```
Digite o valor limite: 6
A soma dos naturais até 6 é igual a 21
```

Instrução de Repetição - Exemplo 3.1 - `while`

Ler uma sequência de números positivos (>= 0) e calcular a sua média.

Finalize a leitura dos dados quando for digitado um número < 0, que não deve ser considerado.

```
soma = 0;
cont = 0;
num = input("Digite o primeiro número");
while num >= 0
   soma = soma + num;
   cont = cont + 1;
   num = input("Digite outro número ou < 0...
                    para terminar");
end
media = soma/cont;
printf("Média dos números= %g",media);
```

Instrução de Repetição - Exemplo 3.2 - `while`

No exercício anterior, o que acontece se nenhum número positivo for digitado? Como resolver este problema?

```
soma = 0;
cont = 0;
num = input("Digite um número positivo ou 0...
                    para fim");
while num >= 0
   soma = soma + num;
   cont = cont + 1;
   num = input("Digite outro número ou < 0...
                    para terminar");
end
if cont > 0 then
      media = soma/cont;
      printf("Média dos números= %g",media);
else
      printf("Não foi digitado qualquer número...
                     positivo")
end
```

Pode-se codificar o exemplo 3 utilizando o comando for?

Instrução de Repetição - `for` ou `while` ?

- Quando usar o **`for`** ou o **`while`**?
- No exemplo 2 (k 1º naturais) o uso do **`for`** é mais adequado.
- Mas, existem situações em que o comando **`while`** é mais adequado, ou, em que <u>não é possível</u> utilizar o comando **`for`**:
  - a) o número de repetições do laço é desconhecido como no exercício 3
  - b) são necessários testes lógicos que não usam somente o operador <=, são usados os demais operadores relacionais e lógicos

Validando dados de entrada

1. Realizar a leitura inicial
2. Repetir a leitura do dado enquanto este não estiver de acordo com as condições desejadas
3. Prosseguir o programa, processando o dado válido

Validação de Dados de Entrada, no caso ≠ de Zero

```
a = input("Entre com o valor de a não nulo: ");
while (a == 0)
      printf("a não pode ser nulo!");
      a = input("Entre com novo valor de a : ");
end
```

Observações:

- **Não se pode prever quantas vezes o usuário entrará com um valor incorreto (nulo);**
- **Não é possível utilizar o comando `for` nem o comando `if` neste caso.**

Validação de Dados de Entrada, no caso maior que Zero

```
a = input("Entre com o valor > 0"); // leitura inicial
while (a <= 0)
  printf("a deve ser > 0");
  a = input("Entre com novo: ");    // próxima leitura
end
```

Observações:

- **Não se pode prever quantas vezes o usuário entrará com um valor incorreto (nulo);**
- **Não é possível utilizar o comando `for` nem o comando `if` neste caso.**

Validação de Dados de Entrada, para um número inteiro

```
a = input("Entre com um valor inteiro");
while (a <> int(a))
      printf("Erro, valor fracionário!");
      a = input("Entre com um valor inteiro");
end
```

Observações:

- **Não se pode prever quantas vezes o usuário entrará com um valor incorreto (nulo);**
- **Não é possível utilizar o comando `for` nem o comando `if` neste caso.**

Observações:

a) use o for sempre que possível, ele será mais seguro e eficiente;

b) cuidado ao utilizar o while, pois será possível que o laço nunca termine (laço infinito), veja 2 exemplos:

```
x = 0;
while x <= 10
  printf("x = %g\n", x)
end
```

O valor de x nunca será alterado. Logo, teremos um laço infinito.

```
x = 3;
while x <= 10
  printf("x = %g\n", x)
  x = x - 1;
end
```

O valor de x é iniciado com 3, sendo depois decrementado em 1.
O valor de x sempre será <= 10.
O programa entra em um laço infinito.

Instrução de Repetição – Exemplo 6

- **Em algumas situações desejamos repetir um programa que acabamos de executar.**
- **Então vamos até o Scinotes e executamos novamente o programa.**
- **É possível executar quantas vezes quisermos um determinado programa, permanecendo no console do Scilab.**
- **Basta acrescentarmos ao código do nosso programa os códigos especificados no exemplo a seguir.**

Instrução de Repetição – Exemplo 6

```
Repetir = "s"; // supõe que o usuário
               // executará o programa pelo
               //  menos uma vez
while repetir == "s"

  // Início do seu programa
  // Comandos do seu programa
  // Fim do seu programa

  // Decisão sobre a repetição do programa
  repetir = input("Repetir? (s/n)", "string");
end
printf ("Término do programa.\n");
```

Exercício 1.

Escreva um programa para ajudar uma pessoa a acumular em uma “conta investimento” o montante igual ou superior a R$ 10.000,00. As regras desta conta são:

- Ela faz um depósito inicial
- A cada meses o montante é atualizado em 1,5% e ela fazer um depósito adicional de qualquer valor
- A aplicação termina quando o montante for igual ou superior à meta de R$ 10.000,00
- Ao final, informar o montante acumulado e o número de meses que decorreram desde o início da aplicação

Resolução:

```
saldo = input("Depósito inicial")
cont = 0
while saldo <= 10000
      saldo = saldo *1,015
      dep = input("depósito adicional")
      saldo = saldo + dep
      cont = cont + 1
end
printf("Saldo acumulado: %.2f\nPrazo decorrido: %g",
saldo, cont)
```

Exercício 1.1

Escreva um programa para ajudar uma pessoa a acumular em uma “conta investimento” o montante igual ou superior a uma **dada meta**. As regras desta conta são:

- Ela faz um depósito inicial
- A cada meses o montante é atualizado em 1,5% e ela fazer um depósito adicional de qualquer valor
- A aplicação termina quando o montante for igual ou superior à meta dada
- Ao final, informar o montante acumulado e o número de meses que decorreram desde o início da aplicação

Verificar se a meta é válida, o.s., se a meta é um valor > 0. Se não for, pedir que digite novamente até que esta seja válida.