

SSC0101 - ICC1 – Teórica

Introdução à Ciência da Computação I

Estrutura de Programas

Repetição – Parte III

Prof. Vanderlei Bonato: vbonato@icmc.usp.br

Prof. Claudio Fabiano Motta Toledo: claudio@icmc.usp.br

Sumário

- Estrutura de Repetição
- Estrutura de Repetição em Algoritmos
 - PARA
 - ENQUANTO
 - REPITA
- Estrutura de Repetição em Linguagem C
 - for
 - while
 - do-while
- Comando de controle de laço
 - break
 - continue
 - goto

Comando do-while

```
do  
{  
    instrução;  
}while(expressão de teste);
```

- Permite executar o bloco mesmo se o teste for falso no início
- Pouco utilizado

Comando do-while

Exemplo 1:

```
//Soma uma série de valores inteiros até receber valor 0
i=0; sum=0;
do{
    sum += i;
    scanf("%d", &i);
}while (i>0)
```

Comando do-while

Exemplo 2:

```
//Recebe apenas inteiros positivos
do{
    printf("Entre com valor inteiro positivo: ");
    scanf("%d", &n);
    if(error = (n<=0))
        printf("\nERROR: Digite novamente!\n\n");
}while (error);
```

Comando do-while

Exemplo 3: Comparando while e do-while

```
//Pedindo pra sair em um while
printf("\nPede pra sair!!! (digite S):");
sair = toupper(getch());
while(sair!='S'){
    printf("\nPede pra sair!!! (digite S):");
    sair = toupper(getch());
}
printf("\nSaiu!!!");
```

Comando do-while

Exemplo 4: Comparando while e do-while (continuação)

```
//Pedindo pra sair com do-while
```

```
do{  
    printf("\nPede pra sair!!! (digite S):");  
    sair = toupper(getch());  
}while(sair!='S');  
printf("\nSaiu!!!");
```

Comando break

- Pode ser utilizado no corpo de qualquer estrutura de laço.
- Causa a saída imediata do laço desviando o programa para a próxima instrução após o laço atual.
- Se estiver em laços aninhados, o break afetará somente o laço que o contém e seus laços internos.
- No caso de laços como **for**, **while** e **do-while**, o laço é interrompido e os comandos do programa são retomados a partir da primeira linha fora do laço.
- No caso do **switch**, a execução da sequência de comandos é interrompida.

Comando break

Exemplo:

```
sum=0;
while(1){
scanf("lf",&x);
if(x<0.0)
    break;
sum += sqrt(x);
}
printf("sum= %f\n", sum);
```

Comando continue

- Força a próxima iteração do laço, não executando o código que vem a seguir
- Ocorre apenas nos comandos “for”, “while” e “do-while”.
- Esse comando deve ser evitado, pois dificulta a leitura e manutenção do código

Comando continue

- No caso do “while” e “do-while” a execução é desviada para o teste condicional e depois segue para o corpo do laço.
 - ✓ Interrompe a sequência de execuções dentro do laço,
 - ✓ Verifica a condição.
- No caso do “for”, o desvio é feito para o incremento, seguido pelo teste condicional e corpo do laço.
 - ✓ Interrompe a sequência de execuções dentro do laço,
 - ✓ Incrementa ou decrementa o contador,
 - ✓ Verifica a condição.

Comando continue

Exemplo:

```
int main(){
    int valor;
    for(valor=1; valor<=10; valor++){
        if(valor%2)
            continue;
        printf("%d ", valor);
    }
    return 0;
}
```

Saída

2 4 6 8 10

Comando continue

Exemplo:

```
int main(){
    int valor, cont = 0;
    do{
        cont++;
        printf("\n\nAdvinhe o valor de um número entre 1 e 10:");
        scanf(" %d", &valor);
        if(valor<1||valor>10)
            continue;
    }while(valor!=5);
    printf("\nValor era: %d", valor);
    printf("\nVoce precisou de %d tentativas para descobrir", cont);
    return 0;
}
```

Comando goto

- Comando considerado dispensável para a programação estruturada
- Alguns programadores defendem o seu uso e outros rejeitam
- O fato é que com as três estruturas de laço vistas anteriormente (for, while, do-while), o “goto” não é necessário
- Suportado em algumas linguagens para manter a compatibilidade com o BASIC e FORTRAN

Comando goto

```
int main()
{
    goto p2;
    printf("passou por aqui - p1 \n");

p2: printf("passou por aqui - p2 \n");

    system("pause");
}
```

Comando goto

- Comandos rotulados identificam determinado comando ou bloco de comandos.
- Exemplos de rótulos:
 - bye: exit(1);
 - L44: a=b+c;
 - bug1: bug2: bug3: printf("bug found\n");
- Identificadores devem ser utilizados como rótulos.
 - 333: a=b+c /* 333 não é um identificador */

Comando goto

- `goto <rótulo>`

- Exemplo:

```
goto erro;
```

```
....
```

```
error:{
```

```
printf("Um erro ocorreu!\n");
```

```
exit(1);
```

```
}
```

Comando goto

- Exemplo:

```
while (scanf("%lf",&x)==1){
  if(x<0.0)
    goto negative_alert;
  printf("%f  %f\n", sqrt(x), sqrt(2*x));
}
negative_alert: printf("Valor negativo encontrado!\n");
```

Referências

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

Kelley, A.; Pohl, I., *A Book on C: programming in C*. 4ª Edição. Massachusetts: Pearson, 2010, 726p.

Kernighan, B.W.; Ritchie, D.M. C, *A Linguagem de Programação: padrão ANSI*. 2ª Edição. Rio de Janeiro: Campus, 1989, 290p.

FIM Aula 8
