INSTITUTO FEDERAL
Sul-rio-grandense | Campus Gravataí

Estruturas de Controle em C - Sequencia

Prof. Hunder Everto Correa Junior

Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas

Introdução

- Antes de entender as estruturas que permitem controlar a execução de um programa em C é necessário esclarecer alguns pontos !

Case Sensitive

- Vamos começar ressaltando um ponto de suma importância: o C é "Case Sensitive", isto é, *maiúsculas e minúsculas fazem diferença*. Se declararmos uma variável com o nome soma ela será diferente de **Soma**, **SOMA**, **SoMa** ou **sOmA**. Da mesma maneira, os comandos do C **if** e **for**, por exemplo, só podem ser escritos em minúsculas pois senão o compilador não irá interpretá-los como sendo comandos, mas sim como variáveis.

Tipos de Dados

- Na linguagem C, as variáveis podem ser de diversos tipos, cada tipo apropriado para uma situação. Os tipos básicos de dados que C suporta são:
- Caractere- char
- Inteiro- int
- Ponto flutuante- float
- Ponto flutuante de precisão dupla-double
- Sem valor-void

Modificadores

- Além dos tipos básicos de dados, existem alguns modificadores que podem ser utilizados juntamente com os dados básicos. Os modificadores são:
- signed
- unsigned
- short
- long

Tipo	Num de bits	Formato para leitura com scanf	Intervalo	
			Inicio	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
signed char	8	%c	-128	127
int	16	%i	-32.768	32.767
unsigned int	16	%u	0	65.535
signed int	16	%i	-32.768	32.767
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
signed short int	16	%hi	-32.768	32.767
long int	32	%li	-2.147.483.648	2.147.483.647
signed long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
float	32	%f	3,4E-38	3.4E+38
double	64	%lf	1,7E-308	1,7E+308
long double	80	%Lf	3,4E-4932	3,4E+4932

Declaração e Inicialização de Variáveis

- As variáveis no C devem ser declaradas antes de serem usadas. A forma geral da declaração de variáveis é:
- *tipo_da_variável lista_de_variáveis;*
- Ex:
- char ch, letra;
- long count;
- float pi;

- Como o tipo default do C é o **int**, quando vamos declarar variáveis **int** com algum dos modificadores de tipo, basta colocar o nome do modificador de tipo. Assim um **long** basta para declarar um **long int**.

Variáveis Globais

- Há três lugares nos quais podemos declarar variáveis. O primeiro é fora de todas as funções do programa. Estas variáveis são chamadas **variáveis globais** e podem ser usadas a partir de qualquer lugar no programa. Pode-se dizer que, como elas estão fora de todas as funções, todas as funções as veem.

Variáveis Locais

- O segundo lugar no qual se pode declarar variáveis é **no início** de um bloco de código. Estas variáveis são chamadas **locais** e só têm validade dentro do bloco no qual são declaradas, isto é, só a função à qual ela pertence sabe da existência desta variável, dentro do bloco no qual foram declaradas.

Constantes

- Constantes são valores que são mantidos fixos pelo compilador. Já usamos constantes neste curso. São consideradas constantes, por exemplo, os números e caracteres como 45.65 ou 'n', etc...

Constantes Barra Invertida

- O C utiliza, para nos facilitar a tarefa de programar, vários códigos chamados códigos de barra invertida. Estes são caracteres que podem ser usados como qualquer outro. Uma lista com alguns dos códigos de barra invertida é dada a seguir:

Código	Significado
\b	Retrocesso ("back")
\f	Alimentação de formulário ("form feed")
\n	Nova linha ("new line")
\t	Tabulação horizontal ("tab")
\"	Aspas
\'	Apóstrofo
\0	Nulo (0 em decimal)
\\	Barra invertida
\v	Tabulação vertical
\a	Sinal sonoro ("beep")
\N	Constante octal (N é o valor da constante)
\xN	Constante hexadecimal (N é o valor da constante)

Exemplo

```
#include <stdio.h>
int main (void)
{
        printf("teste \tde \ncaracteres");
}
```

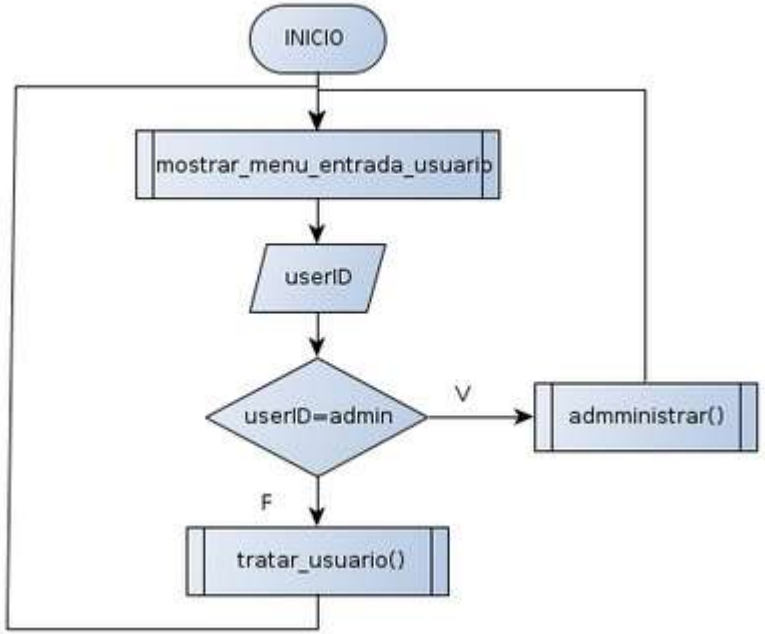
O que \t e \n fazem no programa ?

Podemos tentar algo mais complexo ?

```
#include <stdio.h>
    int main ()
    {
        int Dias;                       /* Declaracao de Variaveis */
        float Anos;
        printf ("Entre com o número de dias: "); /* Entrada de Dados */
        scanf ("%d",&Dias);
        Anos=Dias/365.25;               /* Conversao Dias->Anos */
        printf ("\n\n%d dias equivalem a %f anos.\n",Dias,Anos);
                return(0);
    }
```

Estruturas de Controles

- Assim como em algoritmos, as estruturas de controle do C servem para controlar o fluxo de execução de um programa. Com elas é possível por exemplo, testar opções ou fazer repetições dentre de um programa.



Estrutura Sequencial

- Obedece a uma sequencia comum de comandos sequenciais.

- INICIO
- DESENVOLVIMENTO
- FIM

Exemplo 1 - Sequencia

```
#include <stdio.h>
#include <stdlib.h>
/* Um Primeiro Programa */
int main ()
{
        printf ("Ola Mundo!\n");
        system("pause");
        return(0);
}
```

Entrada e Saída de Dados

Entrada de Dados

- Função **scanf**

```
scanf ("formatos", &var1, &var2,...)
```

Exemplos:

```
int i, j;
float x;
char c;
scanf("%d", &i);
scanf("%d %f", &j, &x);
scanf("%c", &c);
scanf("%s", &nome);
```

%d	inteiro decimal
%f	float
%lf	double
%c	char
%s	string

Entrada de Dados (Exemplo 2)

Algoritmo

ler n1

ler n2

ler n3

ler n1, n2, n3

Na Linguagem C...

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    float  n1, n2, n3, media;
    scanf ("%f",&n1);
    scanf ("%f",&n2);
    scanf ("%f",&n3);
    scanf ("%f %f %f",&n1, &n2, &n3);
    system("PAUSE");
}
```

OBS: não deixe espaço antes do fecha "

Entrada de Dados (exemplo 3)

Algoritmo

ler n1, n2, n3

Media←(n1+n2+n3)/3

Na Linguagem C...

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    float  n1, n2, n3, media;
    scanf ("%f %f %f",&n1, &n2, &n3);
    media=(n1+n2+n3)/3;
    system("PAUSE");
}
```

Problema de Buffer em dados

Char

- Em algumas situações na entrada de dados o C pode não entender que você já deu um [Enter].
- Solução: usar “ %i” (com um espaço entre as aspas e o %)
- Ex:
- scanf (“ %c, &n3); ou
- scanf (“%[^\n]”, &n3);

Formato em Português

- #include <locale.h>

- setlocale(LC_ALL, "Portuguese");

Exemplo:

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

int main()
{

setlocale(LC_ALL, "");

}
```

Saída de Dados

- Função **printf**

```
printf ("formatos", var1, var2,...)
```

Exemplos:

```
int i, j;
float x;
char c;
printf("%d", i);
printf("%d %f", j, x);
printf("%c", c);
```

%d	inteiro
%f	float
%lf	double
%c	char
%s	string

Saída de Dados (Exemplo 4)

```
 #include <stdio.h>
 #include <stdlib.h>
#include <locale.h>
 int main()
 {
    setlocale(LC_ALL, "Portuguese");
    int i, j;
    float x;
    i = 1;
    j = 2;
    x = 3;
    printf("%d", i);
    printf(" %d %f", j, x);
    system("PAUSE");
 }
```

Saída de Dados (Exemplo 5)

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    float n1, n2, n3, media;
   // entrada de dados
    scanf ("%f %f %f",&n1, &n2, &n3);
  // operacao
    media=(n1+n2+n3)/3;
 // saida de dados
    printf("%f", n1);
    printf("%f",  n2);
    printf("%f", n3);
    printf("%f", media);

    system("PAUSE");


}
```

Saída de Dados (Exemplo 6)

```
#include <stdio.h>
#include <stdlib.h>
main()
{
     // definicao de variaveis
     int i, j;
    float  x;


    //entrada de dados
    scanf("%d", &i);
    scanf("%d %f", &j, &x);


    // exibicao de dados
    printf("I= %d\n", i);
    printf("J= %d\nX= %f\n", j, x);


   system("PAUSE");
}
```

Entrada e Saída

Exemplo 7

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    float n1, n2, n3, media;
     scanf ("%f %f %f",&n1, &n2, &n3);
     media=(n1+n2+n3)/3;
     printf ("%f",media);


    system("PAUSE");
}
```

Exemplo 8

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    float  n1, n2, n3, media;
     printf("Digite 3 notas: ");
     scanf ("%f %f %f",&n1, &n2, &n3);
     media=(n1+n2+n3)/3;
     printf ("Media: %.2f\n",media);

     system("PAUSE");
}
```

Faça os Exercícios...

Praticar, importante é !