

Unidade 2
**Elementos de Programação :
 Elementos Básicos**

Engenharia de Petróleo / Química / Produção
Programação I
 Prof. Otacilio José Pereira

Contexto na Disciplina

- 1 – Introdução a Programação
- **2 – Elementos de Programação: Elementos Básicos**
- 3 – Elementos de Programação: Estrutura de Controle
- 4 – Tipos e Estruturas de Dados
- 5 – Modularização

Roteiro das Aulas

- Foco 1 : Exercitar passo a passo e fazer analogia com computador
 - Mostrar exemplo de algoritmo base para raciocínio a seguir
 - Fazer analogia com computador: Entrada e Saída, Processamento e Memória
 - Recursos necessários no algoritmo criado
 - Momentos para a transição do "rascunho" até o programa no computador
- Foco 2 : Elementos básicos
 - Estrutura Geral
 - Entrada e saída
 - Identificador, separador e sequenciamento
 - Memória, variáveis e tipos de dados
 - Atribuição
 - Operadores aritméticos
- Foco 3 : Elementos básicos (continuação)
 - Operações lógicas e relacionais
 - Breve uso de comandos Se e Enquanto
- Foco 4 : Linguagem C
 - Variáveis e tipos de dados
 - Atribuição
 - Operações aritméticas, lógicas, relacionais

Comando SE

- Ainda em Portugol, vamos agora
 - Evoluir a nossa escrita básica de programas em Portugol, que exploraram basicamente sequência de passos simples com fórmulas aritméticas para
 - Programas que podem ter o fluxo das operações modificados por comandos SE e que utilizam tipos e operações lógicas

Aquecimento

- Imagine a seguinte situação
 - Quais os passos você segue para ir para a escola considerando a disponibilidade ou não do carro
 - Para sair de casa para a escola você tem duas alternativas
 - Quando estiver "a pé", seguir um conjunto de passos
 - Quando estiver "de carro", seguir outro conjunto de passos
 - Como poderíamos escrever esta "lógica"
 - Verifica se "Carro está disponível"
 - Se "Carro Disponível"
 - Ligar para a turma
 - Verificar se a turma quer carona
 - Se "Turma quer carona"
 - Pegar a turma
 - Seguir trajeto para escola
 - Senão (isto é, se o "Carro Não Está Disponível")
 - Ir até o ponto
 - Pegar ônibus
 - Saltar próximo do CEUNES
- Como as instruções foram encadeadas, isto é, como o fluxo das instruções está sendo estabelecido?

Comando SE

- Até o momento
 - Nossos programas seguiam uma sequência simples, o fluxo das instruções era seguida uma após a outra
 - Alternativas e decisões
 - Uma maneira de interferir no fluxo é utilizando um comando de condição SE
- se (condição)
então
<bloco quando condição verdadeira>
senão
<bloco quando condição falsa>
fim-se
- A condição é uma expressão lógica que pode gerar um valor Verdadeiro ou Falso, por exemplo:
 - (CarroDisponível = Verdadeiro) ou (Carona = Verdadeiro)
 - (Idade < 18) E (CarteiraMotorista = "Falso")

Operações Relacionais e Lógicas

- Para verificar situações de maior, menor e outros existem os operadores relacionais

$>$ $<$ $=$ \neq \leq (ou \leq) \geq (ou \geq)

Exemplo:

```
inteiro : x;  
logico : emaior;  
x = 10;  
emaior = (x > 5);
```

Obs: a variável emaior neste caso recebeu o valor "Verdadeiro"

- Operadores lógicos são empregados para combinar valores lógicos

- e
- ou
- não

- Exemplo: $(x > 10) \text{ e } (x < 20)$

Exemplos

- Faça um programa que verifique solicite dois números e verifique se M é divisível por N.
- Faça um programa que receba duas idades quaisquer e diga qual a diferença em anos da maior pela menor
- Faça um programa que leia um caracter e verifica se este é vogal ou não

Para aprofundar estudos!

- Leituras recomendadas
 - GUIMARÃES, A. M.; LAGES, N. A. C. Algoritmos Estruturados. LTC, Rio de Janeiro, 1985.
 - Capítulo 2 – Portugol
 - FARRER, Harry et al. Algoritmos Estruturados. 2 ed. Ed Guanabara, Rio de Janeiro, 1989.
 - Capítulo 1 – Até tópico 1.8

- Exercícios

Resolver exercícios da lista 2.2

Contexto

- Na unidade anterior, tratamos a escrita de algoritmos através do que se pode chamar "Relato do Observador", conforme exposto no livro de Lages e Guimarães
- O trabalho agora é organizar ainda mais a escrita do nosso programa para que este fique no ponto para passarmos para um computador
- Observe figura que mostra a transição do problema, passando pelo rascunhos e seguindo até chegar ao programa em uma linguagem de programação, no caso nossa linguagem em C

– Problema =>

Idéias / Rascunho =>

Relato do Observador =>

Portugol => Programa em C

Conceitos de Compilação

- Até então, nossos programas eram escritos em pseudo-código, com instruções já mais bem definidas porém ainda com compreensão restrita a nós.
- O nosso trabalho agora é escrever o programa de forma que este possa ser executado por um computador, para isso utilizando a Linguagem de Programação C
- Para isso é necessário compreender alguns aspectos, dentre eles:
 - Linguagem de programação
 - Compilação
 - Programa fonte
 - Programa executável

Processo

- Esquema
 - Organização das Idéias
 - Relato
 - Portugol
 - Linguagem C
- Ponto de partida:
 - Algoritmo implementado em pseudo-código
- Passos
 - Algoritmo reescrito em uma linguagem de programação
 - O compilador gera o código executável
 - O programa executável pode ser executado em um computador
 - Estas atividades são realizadas em um ambiente de programação
- Ao final
 - Programa executável
- Pegar figuras

Pontos básicos para aprender uma linguagem

- Em geral para iniciar o estudo de uma linguagem, uma sugestão de tópicos iniciais a estudar são:
 - Estrutura básica do programa
 - Blocos e separadores
 - Bibliotecas
 - Tipos de dados e Variáveis
 - Entrada e Saída
 - Atribuição e operações aritméticas
 - Comando de decisão “se”
 - Operações relacionais e lógicas
 - Comentários

Estrutura básica do programa

- Bibliotecas
 - Alguns recursos utilizados nos programas são implementadas em bibliotecas que precisam ser “importadas” para dentro de nosso programa
 - Exemplo

```
#include <stdio.h>
#include <stdlib.h>
...
```
- Comentar função Main
 - O programa em C tem o seu início definido pela função main()
 - Veremos mais sobre funções em capítulos mais a frente

Estrutura básica do programa

- “Esqueleto” do programa em C

```
#include <stdio.h>
#include <stdlib.h>
main ()
{
    \\ declaração de variáveis

    \\ instruções em C
}
```
- Blocos e separadores
 - Para abrir e fechar blocos em c são utilizadas as chaves { }
 - Para separar os comandos entre si, é utilizado o “ ; ”

Tipos de Dados

- Tipos básicos
 - `int` - Números inteiros
 - `float` - Números reais
 - `char` - Para apenas um caractere
- Enquanto no Portugol o caractere envolve um ou mais caracteres, o tipo "char" no C representa apenas um caractere.

No portugol:

`caractere` palavra; \\ Pode ser 'a' ou 'abacaxi'

Na linguagem C:

`char c;` \\ Pode ser apenas um símbolo

Variáveis

- Sintaxe
<tipo de dados> <nome da variável>
- Exemplo
 - `int i;`
 - `float f = 3.45;`
 - `char c;`

Entrada e Saída

- Para a saída, isto é, para mostrar em informações em tela o comando é o `printf`:

```
printf("Este é um texto a ser mostrado em tela");
```
- Para mostrar informações de variáveis, a sintaxe é a seguinte

```
#include <stdio.h>      \\ A biblioteca stdio (Standard Input Output) deve ser importada
```

```
int i = 10;  
float f = 3.70;  
char c = 'a';
```

```
printf("Variável inteira i = %d", i);      \\ imprimirá: Variável inteira i = 10  
printf("Variável float f = %f", f);      \\ imprimirá: Variável float f = 3.70000  
printf("Variável char c = %c", c);      \\ imprimirá: Variável char c = a
```

```
...
```
- Mais a frente veremos mais detalhe sobre entrada e saída, por hora vamos apenas apresentar o básico para a tradução dos programas escritos em Portugol para linguagem C

Entrada e Saída

- Para a entrada de dados, isto é, para leitura de teclado o comando é o `scanf`:
...
`int i;`
`scanf("%d", &i);`
...
- Tipos de dados
 - Para cada tipo de dados da variável a ser lida, deve ser usado um "coringa" usando o %, conforme exemplos abaixo:

<code>%d</code>	para inteiros	<code>scanf("%d", &NumAlunos);</code>
<code>%f</code>	para float	<code>scanf("%f", &PrecoProduto);</code>
<code>%c</code>	para caractere	<code>scanf("%c", &TipoConsumidor);</code>
- Pontos de atenção
 - A variável a ser lida deve estar precedida de "&" indicando o endereço da variável

• Outros comandos podem ser usados para leitura de dados, trataremos também adiante

Atribuições e Operações Aritméticas

- O trato de atribuições e operações aritméticas são equivalentes às empregadas no Portugol.
- Exemplo:
...
`float f, a, b;`
`a = 10 * 3;`
`b = 20 - 2;`
`f = (a * 3) + (45 / b);`
...
- Operadores aritméticos
`+ - * /`

Para divisão inteira:
`%` : para o resto da divisão
`/` : para o quociente da divisão

Comando de Decisão SE

- O comando "SE" escrito no Portugol é tratado com a seguinte sintaxe

```
if (condição)
{ ... }
else
{ ... }
```

- Exemplo

```
...
int numero;

printf("Digite um número");
scanf("%d", &numero);

if ( (numero % 2) == 0 )
{
    printf("O numero %d é par.", numero);
}
else
{
    printf("O numero %d é ímpar.", numero);
}
...
```

Operadores relacionais e lógicos

- Para escrever as condições no comando "if" é necessário empregar operadores relacionais e operadores lógicos
- Operadores relacionais
> < >= <= ==
 - **Atenção que para verificar igualdade o operador é "=="**
- Operadores lógicos
 - Servem para combinar sentenças lógicas
 - Operador E : && (x>10) && (x<20)
 - Operador OU : || (x>10) || (x<20)
 - Operador NOT : ! !(x == 10) \ x não é igual a 10

Comentários

- Para escrever comentários de linha "`//`"
`int i; // este é um comentário de linha`
- Para escrever comentários de linha "`/* ... */`"
`/* Programa: CalculaNota
Autor: Fulano de tal
*/`

Programa completo

```
/* Programa : NumeroPar
Descrição : Verificar se numero par
Autor : Otacilio
Data : 26/08/2009 */

#include <stdio.h> // Bibliotecas
#include <stdlib.h>

main()
{
    int numero;
    // Leitura do número
    printf("Digite um número : ");
    scanf("%d", &numero);

    // Verifica se é divisível por 2 (regra para número par)
    if ( (numero % 2) == 0 )
    {
        printf("O número %d eh par. \n", numero);
    }
    else
    {
        printf("O número %d eh impar. \n", numero);
    }
    system("PAUSE"); // Para antes de encerrar o programa
}
```



Revisão

- Escrita de programa com comando de decisão "SE"
- Início com a programação em C
- Programa básico em C
