Primeiro Programa

Este capítulo apresenta os principais conceitos encontrados na estrutura de um programa de computador escrito na linguagem C.
O programa utilizado como exemplo neste capitulo serve como ponto de partida para os demais exemplos e exercícios ao longo do curso.

O programa Bom Dia

Este é um programa bem simples, cujo propósito limita-se a imprimir a frase "O primeiro programa lhe deseja um bom dia!". Este programa é tão simples que nem sequer possui uma interface gráfica convencional, com a qual estamos acostumados no Windows. Ele abre uma janela própria e, portanto, executa dentro de uma janela do *prompt* do MS-DOS.

```
// PrimeirosPassos.cpp: Nosso primeiro programa em C
#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[]) {
   printf("O primeiro programa lhe deseja um bom dia!");   return 0;
}
```

Estrutura do Código Fonte

No capítulo anterior, aprendemos que utilizamos a linguagem de programação (neste caso, a linguagem C) para descrever como realizar as operações definidas por um algoritmo. No entanto, para se obter um programa funcional, além da descrição do algoritmo em si, precisamos indicar outras informações. Normalmente, encontramos os seguintes itens em um programa típico. Cada um será descrito detalhadamente mais adiante:

- Comentários
- Diretivas de compilador
- Definição procedimentos e funções
- Instruções
- Pontuação

Comentários

Comentário é um texto que será ignorado pelo compilador, mas contém informação útil para o programador. Neste exemplo, utilizamos uma linha de comentários para descrever do que trata nosso programa.
O primeiro tipo de comentário é delimitado pelos pares de símbolos "/*" e "*/", e pode ocupar uma ou mais linhas no código fonte. Já o segundo tipo de comentário, começa com o par de símbolos // e estende-se até o final da linha. Mas tarde, você poderá consultar o capítulo que trata exclusivamente sobre os comentários em C e C++.

Diretivas de compilador

No início do código fonte, antes de qualquer instrução do algoritmo, costuma-se adicionar diretivas de compilador. Seu uso mais recorrente é indicar outros arquivos que devem ser incluídos antes de se iniciar o processo de compilação. Estes arquivos contêm declarações das instruções mais comuns, como por exemplo, leitura e escrita de dados em arquivos.
Outro uso das diretivas é alterar o comportamento do compilador para um determinado trecho do código fonte.
O estudo posterior do Capítulo Bibliotecas esclarecerá a importância e o uso correto das diretivas de compilador. Por ora, para entender os exemplos e compilar os exercícios, os programas devem apresentar as seguintes diretivas:

```
#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
```

Caso seja necessário realizar operações matemáticas avançadas, deve-se incluir a seguinte diretiva:

```
#include <math.h>
```

Procedimentos e Funções

Um procedimento ou uma função é um conjunto de instruções que realizam alguma tarefa.
Desconsiderando os comentários e as diretivas de compilador, o código fonte constitui-se numa seqüência de definições de procedimentos e funções.
Em um programa escrito em C, ou C++, existe uma função especial, chamada main. Esta função contém as instruções por onde se inicia a execução do programa.
O capítulo Procedimentos e Funções, a ser estudado mais para o final do curso, aborda este assunto em detalhes.

Instruções e Pontuação

Cada procedimento ou função é composto por um conjunto de instruções que são executadas seqüencialmente. Cada instrução é separada um da outra por sinais de pontuação. A pontuação mais comum é o ponto e vírgula (;), mas existem outros símbolos que servem de pontuação: ! % ^ & * ( ) – + = { } | ~ [ ] \ ' : " < > ? , . / #.

Práticas de programação

Um bom programador, além de preocupar-se com um código fonte, descrevendo um algoritmo correto e eficiente, também considera a importância da aparência visual do código fonte.
As práticas e estilos aqui recomendados facilitam o entendimento do código fonte.

Linhas em branco. São ignoradas pelo compilador. Utilize para separar blocos de instruções.
Espaços e tabulações. São ignorados pelo compilador. Podem ser úteis para formatar o código fonte. Recomenda-se escrever todas as instruções começando na mesma coluna. Utilize espaços em branco ou tabulações para recuar as

instruções um pouco para a direita, diferenciando-as do cabeçalho do procedimento ou da função. Note que a maioria dos editores de C e C++ fazem isso automaticamente para você.

Uma instrução por linha. Através do emprego correto de pontuação, o compilador é capaz de reconhecer as instruções. Mesmo assim, recomenda-se escrever apenas uma instrução por linha, terminando-a com ponto e vírgula.

No decorrer deste curso, serão apresentadas outras boas práticas de escrita de código fonte.