

## 1. INTRODUÇÃO

Um programa em C++ consiste em um ou mais arquivos. Um arquivo é uma porção de texto contendo código fonte em C++ e comandos do pré-processador. Em outras palavras, ele corresponde a um arquivo fonte em um sistema tradicional. Um arquivo é conceitualmente traduzido em diversas fases, onde a primeira corresponde ao pré-processamento, que realiza a inclusão do arquivo e a substituição de macros, e resulta em uma seqüência de símbolos. A extensão dos nomes dos arquivos fonte em C++, normalmente, é ".cpp". Entretanto, algumas implementações de C++ utilizam ".c", ".C", ".cp" ou ".cxx".

Existem cinco espécies de símbolos em C++: identificadores, palavras-chave, literais, operadores e outros separadores. Brancos, tabulações horizontal e vertical, novas linhas, avanços de formulário e comentários (coletivamente, "espaço em branco"), são ignorados, exceto pelo fato de servirem para separar símbolos. Algum espaço em branco é necessário para separar identificadores, palavras-chave e constantes que de outro modo ficariam adjacentes [2].

Considerando sua estrutura, um programa em linguagem C ou C++ é uma coleção de variáveis, definições e chamadas de função, onde uma função é um conjunto de instruções com um nome e que desempenham uma ou mais ações. Quando o programa começa, ele executa o código de inicialização e chama uma função especial *main()*, onde é colocado o código primário para o programa [1, 4].

Um programa C++ mínimo consiste em:

```
main() { }
```

Este programa define a função *main*, que não possui argumentos e não faz nada. As chaves, { e }, são usadas para expressar agrupamentos em C++; no exemplo anterior, estas indicam o início e o fim do corpo da função (vazia) *main*. Cada programa em C++ deve ter uma função *main*.

Tipicamente, um programa produz alguma saída. O exemplo abaixo mostra um programa que escreve "Hello, World!" na tela do computador.

```
/* Programa Hello, World! */  
#include <iostream>  
void main() // Funcao principal  
{  
    cout << "Hello, World! \n";  
}
```

A linha *#include <iostream>* instrui o compilador a incluir as declarações das facilidades de um fluxo de entrada e saída padrão, encontradas em *<iostream>*. Sem esta declaração, a expressão *cout << "Hello, World! \n"* não faria sentido. O operador *<<* ("colocar em") escreve o seu segundo argumento no primeiro. Neste caso, o *string* "Hello, World! \n" é escrito no fluxo de saída padrão *cout*. Um *string* é uma seqüência de caracteres entre aspas duplas. As aspas servem para o compilador entender o texto delimitado, isto é, para que ele não processe os caracteres circundados como se fossem instruções de programação ou como outros comandos. Em um *string*, o caracter "\" seguido por outro caracter denota um único caracter especial; neste caso "\n" é o caracter de nova linha, então é escrito "Hello, World!" seguido de uma troca de linha [4, 7].

O valor inteiro retornado pela função *main*, se houver algum, é o valor de retorno do programa ao "sistema". Se nada é retornado, "o sistema" irá receber um valor randômico. As facilidades fornecidas pela biblioteca *stream* serão examinadas mais detalhadamente posteriormente [7].

É importante comentar que C e C++ são linguagens de "forma livre". Com poucas exceções, o compilador ignora os retornos de carro e espaços em branco. Sendo assim, deve existir uma maneira de determinar o fim do comando. Em linguagens C e C++, os comandos são delimitados por ponto-e-vírgula.

Os comentários na linguagem C++, como mostra o exemplo, podem ser feitos de duas maneiras: ou eles começam com `/*` e terminam com `*/`, podendo incluir retorno de carro; ou eles utilizam `//` para iniciar o comentário que termina com um retorno de carro. Este último é mais conveniente que `/* */` em comentários de uma linha [1].