C++ é uma linguagem de programação compilada multi-paradigma *(seu suporte inclui linguagem imperativa, orientada a objetos e genérica)* e de uso geral. A linguagem é considerada de médio nível, pois combina características de linguagens de alto e baixo níveis.

Desde os anos 1990 é uma das linguagens comerciais mais populares e mais usadas por apresentar grande desempenho.

História e Desenvolvimento

Bjarne Stroustrup, do Bell Labs, foi quem iniciou seu desenvolvimento no ano de 1983 com o objetivo implementar uma versão distribuída do núcleo Unix, que era escrito em C.

Para desenvolver a linguagem, foram acrescentados elementos de outras linguagens de vários níveis, como a Simula 67, que possuía características bastante úteis para o desenvolvimento de software, mas era muito lenta para uso prático, e a linguagem BCPL, que era rápida mas possuía demasiado baixo nível, dificultando sua utilização no desenvolvimento de aplicações. Outras linguagens utilizadas no desenvolvimento foram a ALGOL 68, Ada, CLU e ML.

Em 1983 o nome da linguagem foi alterado de *C with Classes* para C++. Antes implementada usando um pré-processador, a linguagem passou a exigir um compilador próprio, criado pelo próprio Stroustrup.

Novas características foram adicionadas, como funções virtuais, sobrecarga de operadores e funções, referências, constantes, gerenciamento manual de memória, melhorias na verificação de tipo de dado e estilo de comentário de código de uma linha (//). Em 1985 foi lançada a primeira edição do livro *The C++ Programming Language*, contendo referências para a utilização da linguagem, já que ainda não era uma norma oficial. A primeira versão comercial foi lançada em outubro do mesmo ano.

Assim como a linguagem, sua biblioteca padrão também sofreu melhorias ao longo do tempo. Sua primeira adição foi a biblioteca de E/S, e posteriormente a Standard Template Library (STL); ambas se tornaram algumas das principais funcionalidades que distanciaram a linguagem em relação a C.

Depois de anos de trabalho, o comitê ANSI/ISO padronizou o C++ em 1998 (ISO/IEC 14882:1998). Após alguns anos foram reportados defeitos e imprecisões no documento, e uma correção foi lançada em 2003.

Pode-se dizer que C++ foi a única linguagem entre tantas outras que obteve sucesso como uma sucessora à linguagem C, inclusive servindo de inspiração para outras linguagens como Java, a IDL de CORBA e C Sharp.

Operadores

Os operadores em C++ são um conjunto de todos os operadores do C com novas adições à linguagem. Um grupo de novos operadores do C++ são os relativos à conversão de tipo de dado, e consistem em **const\_cast**, **static\_cast**, **dynamic\_cast** e **reinterpret\_cast**. Temos também o novo ‘**typeid**’, que retorna informações sobre o tipo de dado derivado pelo operando. Outro grupo de novos operadores são os relativos à alocação de memória, e consistem em ‘**new**’ e ‘**delete**’. Temos ainda o novo operador de resolução de âmbito, **::** , e que se refere diretamente ao suporte de espaço de nomes e orientação a objeto oferecido pela linguagem.

Pré-processador

O C++ é compilado em três fases: pré-processamento, compilação propriamente dita (tradução para código objeto) e ligação.

Durante a primeira fase, as diretivas de pré-processamento são aplicadas através de transformações léxicas no próprio código fonte, que então alimenta as próximas fases de compilação. Elas são identificadas no código através do caractere **#**. O pré-processamento é utilizado para substituir partes de código, para inutilizar partes de código e para importar módulos externos.

Gabaritos

Os gabaritos são diferentes de macros de programação: enquanto ambas as facilidades podem ser utilizadas para produzir código em tempo de compilação, gabaritos não se restringem a substituições léxicas. Eles possuem conhecimento da semântica e do sistema de tipagem da linguagem, e são utilizados principalmente para polimorfismo estático (ver exemplo em anexo) e programação genérica. Um gabarito também é uma máquina de Turing completa.

Objetos

O C++ introduziu alguns conceitos de orientação a objetos ao C, como exemplificado pelas classes, que apresentam quatro características comumente presentes em linguagens de programação orientadas a objeto: abstração, encapsulamento, herança e polimorfismo. Cada vez que uma classe é instanciada é criado um objeto na memória, que é basicamente um conjunto de atributos e operações reunidos.

Tratamento de exceções

O tratamento de exceção é um mecanismo desenvolvido para lidar com a ocorrência de algumas condições (chamadas exceções) que alteram o funcionamento normal do fluxo de um programa de computador. O C++ suporta tal tratamento, de forma que o estado atual de um programa após uma exceção é alterado automaticamente para outro estado pré-definido para a recuperação do sistema.

Espaço de nomes

O C++ introduziu os espaços de nomes para a organização das bibliotecas, e sua função é agrupar um contexto para identificadores (variáveis, funções, classes, estruturas, entre outros). No contexto de sistemas operativos, o espaço de nomes poderia ser representado por diretórios. Toda a biblioteca padrão está contida no espaço de nomes std (abreviação de *standard*, que em inglês significa *padrão*).

Incompatibilidade com C

É incorreto considerar o C++ como um super conjunto de C, isto é, uma linguagem que implementa o C completamente e que adiciona novas funcionalidades. Grande parte de código C pode ser perfeitamente compilado em C++, mas existem algumas pequenas diferenças sintáticas e semânticas entre as linguagens que tornam alguns trechos de código C válidos em código C++ inválido, ou códigos que exibem comportamentos diferentes em cada linguagem.

Outra questão de portabilidade entre as linguagens é o fato do C++ adicionar várias novas palavras reservadas, como **new** e **class**, que podem ser utilizadas como identificadores (por exemplo nomes de variáveis) em C, gerando incompatibilidade.

Exemplos de código

Exemplo de um programa que imprime na tela "Olá, Mundo!".

#include <iostream>

int main()

{

std::cout << "Olá, Mundo!" << std::endl;

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**“C faz com que dar um tiro no pé seja fácil; C++ torna isso mais difícil, mas quando nós o fazemos, arrebentamos com a perna toda.” — Bjarne Stroustrup, criador da linguagem C++.**