

Diferenças entre C e Java

Apesar da sintaxe das Linguagens Java e C serem similares, elas são linguagens muito diferentes. A seguinte tabela mostra as principais diferenças:

Java	C
Orientada a objeto	Procedural
Interpretada	Compilada
Gerência de memória	Sem gerencia de memória
Referências	Ponteiros
Exceções	Códigos de erros

Orientado a Objetos vs. Procedimento

Uma das maiores diferenças entre Java e C é o uso de um paradigma de programação diferente. Java é uma linguagem orientada a objeto. Um programa Java consiste em uma coleção de objetos. Estes objetos contêm os dados usados no programa e possuem métodos para executar operações nesses dados.

A linguagem C é procedural. Para ser considerada **procedural**, uma **linguagem** de programação deve suportar o conceito de procedimentos (ou funções), e possuir uma sintaxe para defini-los. Idealmente, ela deve suportar a especificação de tipos de argumentos, variáveis locais, chamadas recursivas e o uso de procedimentos em módulos distintos de um programa.

Um programa C consiste em uma coleção de procedimentos (ou funções). Os dados utilizados pelo programa podem ser colocados em local variáveis (dentro de uma função) ou variáveis globais (fora das funções). Não há noção de objetos em C. Assim como em Java, há uma função principal especial, que é usado para iniciar o programa.

Interpretado versus Compilado

Java é uma linguagem interpretada. O código original Java é transformado em bytecode, que é carregado por um programa “intérprete”. Este programa então 'executa' cada uma das instruções de bytecode, traduzindo-os em código da máquina. Programas C são compilados. Em vez de ser traduzido para algum formato intermediário (como bytecode), é traduzido diretamente em código de máquina. Este código de máquina é executado diretamente pelo processador.

Gerenciamento de memória

Em Java, o gerenciamento de memória é feito automaticamente pelo intérprete. Novos objetos podem ser criados usando um comando *new*. Quando os objetos não são mais usados (ou seja, não existem mais referências apontando para eles) eles são removidos pelo “coletor de lixo”. Em C, o programador precisa fazer seu próprio gerenciamento de memória. Usando chamadas do sistema através das funções *malloc* e *free*, blocos de memória podem ser alocados e liberados.

Referências vs. Ponteiros

Ponteiros são variáveis que armazenam endereços de memória e permitem referenciá-los. Referências dizem respeito a objetos ou variáveis específicos, abstraindo-se totalmente o lugar e forma de armazenamento.

Como ponteiros armazenam explicitamente (e te permitem manipular) endereços, a linguagem C possui algumas propriedades como aritmética de ponteiros, aliasing (tratar o mesmo endereço de memória como tipos de dados diferentes) e infelizmente, acesso a endereços inválidos.

A linguagem Java abstrai a alocação de memória e abdica do conceito de ponteiro e usa conceitos mais abstratos, como referências ao objeto.

Uma referência em Java é uma variável especial que referencia (aponta para) um objeto. Somente objetos podem ser referenciados. Por exemplo, não é possível ter uma referência a uma variável do tipo *int*.

Exceções vs. Códigos de erro

Sempre que ocorre um erro em Java, uma exceção é gerada. C não tem exceções. Uma função retorna algum código de erro (quando se espera um erro) ou seu programa trava (geralmente muito tempo após a ocorrência do erro).