

## Artigo EWD 215: A Case against the GO TO Statement:

Dijkstra descobriu que o uso da instrução GO TO tem efeitos desastrosos, ficando convencido que a instrução deveria ser abolida de todas as linguagens de programação de “Alto nível”, isto é, tudo menos o código de máquina. Ele relata que embora a atividade do programador termine quando ele constrói um programa correto, a “criação” do processo correspondente é delegado à máquina. Anos antes, Corrado Böhm e Giuseppe Jacopini provou que qualquer programa poderia ser feito usando laços e desvios condicionais, o chamado teorema da programação estruturada. Outro argumento utilizado por Dijkstra é o fato de que o programador pensa melhor a sequência de comandos estaticamente (a sequência dos comandos textualmente) do que dinamicamente (a sequência que surge quando o programa é executado, mas o código pode estar em locais distintos). Assim, o uso excessivo de desvios poderiam provocar o chamado código espaguete: o fluxo do programa sofre muitas idas e vindas, assim como o macarrão enrolado.

## Artigo Double-checked locking: Clever, but broken:

A linguagem DCL foi projetada para suportar a inicialização preguiçosa, que ocorre quando uma classe adia a inicialização de um objeto de propriedade até que ela seja realmente necessária. O Java trata cada thread como se fosse executado em seu próprio processador com sua própria memória local, cada conversando e sincronizando com uma memória principal compartilhada. O DCL depende de um uso não sincronizado do campo de recurso. Isso parece ser inofensivo, mas não é. A maneira mais eficaz de corrigir o idioma DCL é evitá-lo e a maneira mais simples de evitá-lo é usar a sincronização. O JMM foi a primeira tentativa de definir semântica de memória compartilhada no contexto da especificação de linguagem para uma linguagem de programação de propósito geral. Infelizmente, é bastante complicado, mal compreendido e não implementado consistentemente em JVMs. Não surpreendentemente, a complexidade do JMM resultou em erros de implementação JVM e mal-entendidos generalizados entre os usuários - como a percepção errada de como funciona sincronizado, o que leva a linguagens inseguras como DCL.