

RECURSIVIDADE

Samuel Monteiro Ferreira

0019208

ENGENHARIA DE SOFTWARE – 6 SEMESTRE

GAMA – DF



(61) 3035-3900



SIGA Área Especial para Indústria nº 02
Setor Leste - Gama - DF
CEP: 72445-020



O que é recursividade na programação?

A recursão pode ser considerada um processo de repetição de uma rotina. De uma maneira mais simples a recursividade pode ser definida como uma rotina (procedimento ou método) que chama a si mesmo, de forma direta ou indireta.

Dessa forma é preciso tomar cuidado com os loops, porque como é um método que chama a si mesmo inúmeras vezes se implementado da forma incorreta, pode ocasionar um erro de loop infinito e acabar ocasionando um travamento no sistema.

E para que não aconteça esse tipo de problema, é necessário definir uma condição de terminação de forma correta. E até mesmo uma condição de terminação pode causar loop.

Existe a recursão direta e recursão indireta.

RECURSÃO DIRETA: uma rotina que é formada por um conjunto de comandos e uma chamada a ela mesma.

RECURSÃO INDIRETA: uma rotina que contém uma chamada a outra rotina que, por sua vez, tem uma chamada a outra rotina e assim sucessivamente.

Um método recursivo é chamado pra resolver um problema no qual ele sabe resolver somente a parte mais simples, o caso mais trivial. A solução pra um problema recursivo normalmente pode ser dividido em duas partes: Uma solução que é trivial e outra que é mais geral.

A recursão aplica uma técnica de divisão e conquista, funciona mais ou menos assim: Se o problema que é pra ser resolvido é muito grande, o problema é dividido em dois, se ainda continuar grande, é dividido em mais dois e assim sucessivamente, até chegar em algo mais simples.

A recursão sabe como resolver parte de um problema, mas outra parte não, então ela vai dividindo o problema em duas partes.



No código acima, qual é o caso base e por que ele é necessário?

O caso base no código da atividade, é uma condição que faz com que o método pare de chamar assim mesmo. No código o caso base é o “ENCERRAMENTO DA FILA DE ATENDIMENTO DE CLIENTES”, ele é necessário porque ele evita com que o método entre num loop infinito e gere o erro de stackOverflow.

O que aconteceria se não houvesse um caso base no programa?

Como já dito anteriormente, se não houvesse um caso base, o método iria chamar a si mesmo infinitamente, causando um erro.

Dê um exemplo do mercado de trabalho (diferente do da fila) onde a recursividade pode ser utilizada.

Um exemplo da recursividade no mercado de trabalho seria numa empresa com grande volume de dados onde seria necessário realizar uma navegação nos dados, como uma estrutura de produtos em um sistema.

Explique a diferença entre recursividade e laços de repetição (for/while).

A recursão e os laços de repetição como for, while, do while e for Each, são estruturas de controle que permitem a execução repetida de um certo bloco de código, mas com as abordagens diferentes.

A recursão envolve um método chamando ele mesmo, enquanto os laços de repetição usam estruturas padrões para iterar sobre um bloco de algoritmo.

Laços de repetição são estruturas explícitas e a recursividade é implícita na definição do método.

Laços controlam a repetição por meio de condições e a recursão se baseia no próprio método.



Outra diferença é na performance, porque a recursão pode ser mais lenta e consumir mais memória.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02
Setor Leste - Gama - DF
CEP: 72445-020

