

ABAPZombie Guide to ABAP – Parte 27 – CHECK

31/08/2011 11:51

Bom, eu estou com um problema chato porque estou sem acesso ao meu ambiente Trial do Netweaver, fiz o favor de destruir a Virtual Machine tentando convertê-la para usá-la em um outro software. Mas como ninguém se importa com isso (😞), vamos analisar mais um comando, o **CHECK**!

Síntaxe do comando: CHECK <expressão lógica>, e só. “Mas ué, não tem um ENDCHECK?” Não.. até porque se tivesse seria igual ao IF, mas é exatamente por não tem um ENDCHECK que a coisa começa a complicar...

O CHECK é um comando mutável, isto é, seu funcionamento muda dependendo de onde ele estiver, sem alteração nenhuma em sua sintaxe.

Se você colocar um CHECK dentro de um laço, como este:

```
REPORT zombie_check_laco.

* Coloquei um DO, mas pode ser um WHILE, um LOOP, SELECT..ENDSELECT,
* qualquer laço da linguagem
DO 10 TIMES.

    WRITE / 'Será que o mundo vai acabar?'.

    * Com este check, o mundo nunca irá acabar
    CHECK 1 = 2.

    WRITE 'O MUNDO PORQUE ACABOU 1 É IGUAL A 2'.

ENDDO.
```

Sempre que um CHECK estiver dentro de um laço e a expressão lógica que você estiver *checkando* for falsa o programa irá voltar para o início do laço e seguir para o próximo registro(loop/select) ou para o próximo passo do laço (do, while).

No caso acima, a mensagem *Será que o mundo vai acabar?* seria impressa 10 vezes.

Porém, se você colocar um CHECK fora do laço de repetição, **ele vai abortar o processamento do bloco onde o CHECK estiver**, seja este bloco um FORM..ENDFORM, uma Função, um Método de uma Classe, ou até mesmo um Report.

Exemplo:

```
REPORT zombie_check_sem_laco.

* Com este check, o mundo nunca irá acabar
CHECK 1 = 2.

WRITE 'O MUNDO PORQUE ACABOU 1 É IGUAL A 2' .
```

No caso acima, o programa não vai escrever nada na tela.

E é aí que mora o perigo... Se você estiver no meio de um programa gigantesco e que não esteja muito bem estruturado, colocar um CHECK e o CHECK retornar falso você pode abortar o **programa INTEIRO**. Portanto, muito cuidado crianças! 😞

Isso cobre os usos mais comuns do CHECK!

Abraços à todos que já destruíram uma execução do Standard colocando um CHECK no meio de uma include perdida dentro de uma EXIT. Vocês não estão sozinhos, acreditem! 😊 😊

Comentários

junilson — 28/03/2022 06:34

Muito Obrigado pelas partilhas de conhecimento.

RICARDO MELO DOS SANTOS — 15/11/2016 12:04

Você pode me ajudar nesse exercício ?

Por favor

Faça a divisão de um valor sucessiva por 2 até chegar a 0,01. Coloque um parameter para inserção do valor na tela. Utilize o DO/ENDDO. O Valor deverá ter a precisão de 4 dígitos e 2 casas decimais.

Marcelo — 18/10/2011 08:40

Já cansei de ver casos dentro de EXIT's onde temos o nosso código e chega um espertinho e insere um CHECK no início da EXIT que só deveria ser válida para seu processamento, mas que acaba pulando todo o código posterior, no caso o meu que deveria ser processado! Putz!

Mauricio Cruz — 18/10/2011 08:59

Uma vez colocaram um check dentro de uma macro no início de uma exit maluca, e eu, junior como era na época, não entendia NEM A PAU porque o programa não passava na minha exit 😞

ahhaha

André Valença — 07/08/2021 10:34

Acredite atualmente esse é o meu caso, e eu não sei como fazer o programa continuar, por isso eu cheguei nesse artigo rsrsrsr