Hardcode, Constantes, C_Xs e derivados / ABAPZombie Guide 28 – CONSTANTS

01/09/2011 08:00

Este post vai além de uma explicação de um comando...

Você já viu alguma vez em algum código fonte a seguinte declaração?

```
REPORT zombie_constante_idiota.

CONSTANTS: c_x TYPE c value 'X'.
```

😈

O QUE? **VOCÊ JÁ ESCREVEU ISSO ALGUMA VEZ NO CÓDIGO?**

😈 😈 😈

Neste post vamos entender para que serve este comando, e também entender porque constantes são úteis. Quem sabe, de uma vez por todas, o povo **PARE DE CRIAR CONSTANTES SEM MOTIVO**.

O comando CONSTANTS é usado na declaração de constantes, ou seja, variáveis com uma tipagem fixa e com um valor que nunca irá mudar em tempo de execução.

Exemplo:

```
REPORT zombie_constants.

CONSTANTS c_tipo_doc TYPE char4 VALUE 'WXYZ' .
```

Você pode usar a variável c_tipo_doc tendo a certeza de que seu valor nunca será alterado em tempo de execução. Você pode até criar uma "estrutura constante":

```
REPORT zombie_constants_estrutura.

CONSTANTS: BEGIN OF c_estrutura,
             valor1 TYPE i VALUE '123',
             valor2 TYPE n VALUE '2',
            valor3 TYPE c VALUE 'A',
            END OF c_estrutura.

WRITE c_estrutura-valor1.
```

Ok, isso acaba com a explicação. Vamos agora à pergunta do milhão...

PORQUÊ EU DEVO USAR UMA CONSTANTE?

Imagine que você tem que fazer um programa que irá mostrar dados de uma deteminada empresa, e você terá que usar este código de empresa para fazer diversas seleções e validações dentro do código.

Se você sair escrevendo *literalmente* o valor desta empresa no código – por exemplo, *IF empresa = 'ZMB1'* – o seu código vai funcionar, mas se algum dia o código dessa empresa mudar de ZMB1 para ZMB2 você vai ter que **alterar o código INTEIRO** por conta de uma simples troca de valores.

Ao invés de espalhar o valor ZMB1 pelo código inteiro, você pode criar uma constante – por exemplo *C_EMPRESA* com o valor ZMB1 – e, se um dia o código mudar, você pode alterar somente UMA linha de código, para que seu programa fique correto e use o novo código de empresa.

HARDCODE E A BANALIZAÇÃO DAS CONSTANTES

Quem falou que programadores deveriam evitar hardcodes foi **sábio**. Mas em momento algum ele disse que **TODO VALOR ESCRITO LITERALMENTE NO CÓDIGO É UMA CONSTANTE**. Eu cansei de ver constantes C_X = 'X' espalhadas por códigos fontes. Isso é uma estupidez, porque o código fica horrível de ler, e o **valor da variável não vai poder ser mudado nunca MESMO**. Exemplo:

```
PARAMETERS: p_docto AS CHECKBOX,
            p_users AS CHECKBOX,
            p_soma  AS CHECKBOX.

CONSTANTS c_x TYPE c VALUE 'X'.

IF p_docto = c_x.
* Validações
ENDIF.

IF p_users = c_x.
* Validações
ENDIF.

IF p_soma <> c_x.
* Validações
ENDIF.
```

Se você mudar o valor de c_x para vazio, **isso não quer dizer que todos os lugares onde C_X está sendo usado TEM de ser alterados**. Ou seja, você criou uma variável que não precisava ser criada, que você nunca vai mudar o valor dela na declaração, e que deixou o seu código irritantemente mais chato de ler.

Pode parecer besta se eu colocar só contante burra de exemplo, mas eu já ví reports com mais de 20 contantes que não precisavam estar ali... e isso é um **SACO** para dar manutenção e debuggar. Eu já ví até projetos que obrigavam que você criasse constantes para todos os valores que você tivesse no código...

Uma só palavra quando eu vejo C_X e derivados no código: **ZUADO.**

PIOR é que eu já vi códigos onde a pessoa criou uma outra constante "esperta", já que não podia alterar o valor da C_X:

```
PARAMETERS: p_docto AS CHECKBOX,
            p_users AS CHECKBOX,
            p_soma  AS CHECKBOX.

CONSTANTS c_x     TYPE c VALUE 'X',
          c_space TYPE c VALUE ' '.

IF p_docto = c_x.
* Validações
ENDIF.

IF p_users = c_space.
* Validações
ENDIF.

IF p_soma <> c_x.
* Validações
ENDIF.
```

FUUUUUUUUUUUUUUUUUUUUUUUU

OUTRO TIPO DE CONSTANTE BURRA

Colocar no nome da constante o mesmo que o valor dela é outro exemplo de constante BURRA e ABOMINÁVEL:

```
CONSTANTS c_ZMB1 TYPE CHAR4 VALUR 'ZMB1'.
```

Se um dia mudar 'ZMB1' para 'ZMB2' vai ficar fácil de entender que eu estou usando a C_ZMB1 num IF, mas, na verdade, ela vale ZMB2...

Vamos dar um nome para esse tipo de constante: **Abominável Constante das Neves**

MAS E AÍ, QUANDO EU SEI QUE TENHO QUE USAR UMA CONSTANTE?

Vale o bom senso e a correta análise da regra de negócio.

Se você tem um valor que você sabe que vai ser usado em diversos pontos do programa, que o cliente/funcional não deve nunca poder mudar sem requisitar uma alteração na especificação, e que importa para o PROCESSO do desenvolvimento (tipo de documento, uma empresa, um número de conta contábil, milhões de exemplos aqui), use uma constante.

Se o caso for igual ao acima, mas o cliente/funcional podem querer alterar ou expandir os valores, vá de TVARV (vou explicar sobre a TVARV num post futuro).

Porém, se o seu caso não se encaixar neste dois, ou for usado somente em um lugar do código... **coloque o valor literal e seja feliz.** 🙂

Abraços à todos aqueles que já discutiram com pessoas que acreditam que o uso de C_X em códigos com mais de 10mil linhas é correto.

😉

Comentários

> **Vivian** — *05/04/2013 11:37*
>
> Muito bom! Eu mesmo usava isso pensando que estava correto, não por ter a "síndrome do robô", mas por simplesmente pensar que era melhor e pronto, sem pensar no real uso de constants. Ótima explicação e com

certeza irei levar para toda a vida! rsrs

Paulo Rios — *21/09/2011 07:18*

Cara, muito bom esse post. Já declarei muita constante burra por que o QA enche o saco. É ridículo mas o C_X é bem freqüente nos programas que vejo. Agora vou evitar ao máximo e indicar esse post aos que ainda relutam em criar constantes burras...

Abraços!

Mauro Laranjeira — *20/09/2011 14:58*

Galera,

Só para lembrar, aqui no abapzombie tem uma classe para auxiliar nas seleções de parâmetro da STVARV, segue o link:
http://www.abapzombie.blog.br/dicas-abap/2011/01/...

Segue o tuto de como criar migrar uma classe local para uma global:
http://www.abapzombie.blog.br/dicas-abap/2011/01/...

Sem zombizadas galera =D
Abs,

Mauro

Julio — *19/09/2011 11:41*

Cara, eu também já passei por essa de TER que colocar C_X... Tantas vezes que já peguei esse vício maldito e lembro de diversos códigos meus poluídos por constantes burras...

Flávio Furlan — *13/09/2011 14:11*

Uma explicação que ouvi uma vez para o uso de C_X ou C_ZMB1 é que a empresa usava um programa para detectar literais no código e com as aberrações acima o programador enganava esse programa e assim evitava a dor de cabeça de ficar justificando o uso do "X" como hard code!

Ainda acho que a sugestão do Fábio muito boa. Eu mesmo adotei até nos treinamentos que ministro o uso de abap_true e abap_false.

Abs!

Custodio — *01/09/2011 17:29*

abap_true e abap_false sao (boas) gambetas da SAP ja que o ABAPao nao tem "booleans reais". acho que de todos os exeplos do post o pior eh o tal de c_zmb1.

Mauro Laranjeira — *01/09/2011 15:27*

Uma vez alterei um programa gigante que tinha constantes burras, fiquei na duvida se mudava também o nome da constante rsrsrs, deu muito trabalho, mas fiz o melhor que pude rsrs...

Fábio Pagoti — *01/09/2011 10:34*

Sem dúvida, muito ABAPeiros tem essa mania e que acaba sendo contagiosa (já fiz muita constante burra – e não por opção).
Caso alguém não esteja convencido de que este post tem total sentido, economize ao menos o c_x e o space (ou c_space) pelas constantes abap_true e abap_false.

Abraços!

> **Mauricio Cruz** — *01/09/2011 11:27*
>
> Sim!
>
> Acho que quase todas as liguagens tem uma constante já pronta para true ou false, e para o ABAP não é diferente.
>
> E eu também já fui obrigado a fazer constantes burras, mas isso não durou muito – e acho que por isso que eu já entrei em tantas discussões enérgicas sobre hahaha

ABAP da Depress&atil — *01/09/2011 07:50*

Isso é realmente uma das coisas mais simples e irritantes que existem, não só em ABAP como em qualquer linguagem. E usar a TVARV acaba com aqueles chamados abertos só pra alterar um C_BUKRS ou C_WERKS.

Além disso, C_X é um absurdo quando usado para testar flags... custa usar NOT ... IS INITIAL?

> **Mauricio Cruz** — *01/09/2011 07:57*
>
> Eu, particularmente, gosto de uma que eu vi uma vez:
>
> c_aeiou
>
> Bonito o nome, né? hahahahaha

Daiane — *01/09/2011 06:34*

Você disse tudo!!!!!! Também acho que houve uma banalização das constantes... e daí vem a "síndrome do robô" e não analisa a real necessidade do hardcode.
Eu já briguei, bati o pé, pois no QA queriam que eu retirasse os hardcodes 'X'.... AHHHHHHH!!!!!
Mas, como eu tenho juízo eu obedeço (mesmo totalmente contrariada) rsrs

> **Mauricio Cruz** — *01/09/2011 07:58*
>
> Aposto que deve ter gente que já declara um C_X quando inicia um report, tenho CERTEZA! hahaahahah

> **Gustavo** — *01/09/2011 05:19*
>
> Show de bola esse post!