

## Tratamento de erro com classe de exceção

11/08/2011 16:38

Erros?! Dump?! NNNãããooooooooo...

"Aquele à qual o programa nunca deu dump, que durrube o primeiro processo pela SM50" :S

Alguns dumps mostram uma tal de "Exceção", essa exceção tem um motivo para aparecer, há quem diga que é apenas enfeite. =o

No exemplo abaixo de dump aparece a classe de exceção CX\_SY\_CONVERSION\_NO\_NUMBER.

### Dump

Este Dump ocorreu pelo seguinte código:

```
DATA: vl_string TYPE char10,
      vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

MOVE vl_string TO vl_numc.
```

Ou seja, por colocar texto em uma variável do tipo de inteiro.

No Help de cada comando, mostra todas as exceções que o comando pode disparar. Exemplo do Help para o comando MOVE:

### Help do comando MOVE

Mas vale lembrar que nem todos os comandos possuem classe de exceção.

Para tratar este erro, a galera iria desenvolver uma lógica ou if para verificar se a variável tem alguma coisa diferente de números.

Para este caso podemos utilizar a classe de exceção CX\_SY\_CONVERSION\_NO\_NUMBER para o tratamento deste erro:

```
DATA: vl_string TYPE char10,
      vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

TRY.

    MOVE vl_string TO vl_numc.

    CATCH cx_sy_conversion_no_number.

        MESSAGE e000(o0) WITH 'Erro de conversão'.

    ENDTRY.
```

Podemos utilizar mais de uma classe de exceção para os tratamentos de erros:

```
DATA: vl_string TYPE char10,
      vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

TRY.

    MOVE vl_string TO vl_numc.

    CATCH cx_sy_conversion_no_number.

        MESSAGE e000(o0) WITH 'Erro de conversão'.

    CATCH cx_sy_conversion_overflow.

        MESSAGE e000(o0) WITH 'Estouro de variável'.

    ENDTRY.
```

Para um tratamento de erro genérico com classe de exceção, podemos usar a classe mãe de todas as classes de exceções, a classe CX\_ROOT.

```
DATA: vl_string TYPE char10,
      vl_numc   TYPE i.

vl_string = 'ABAPZOMBIE.BLOG.BR'.

TRY.

    MOVE vl_string TO vl_numc.

    CATCH cx_root.

        MESSAGE e000(o0) WITH 'Tratamento para todos os erros de classe de exceção'.

    ENDTRY.
```

Agora o “ABAP Espertinho” pode fazer reports sem dumps:

```
REPORT zabapespertinho.

TRY.
```

```
"Toda a Lógica do programa
```

```
CATCH cx_root.
```

```
MESSAGE e000(o0) WITH 'Deu erro, mas não DUMP =)'.

```

```
ENDTRY.
```

Abraços a todos os Zombies!!!

## Comentários

**Mario Eiji Morishita** — 22/05/2016 16:51

Ajuda olá a todos, estou começando minha vida profissional no ABAP agora e gostaria de uma ajuda não sei se aqui seria melhor opção desculpe-me.

Estou tentando usar o comando MOVE de um campo x para um outro campo de uma outra tabela como poderia fazer isto.

**Matheus Tristão** — 03/01/2017 13:22

Mario, crie um field symbol com o mesmo tipo da tabela. Faça um loop na tabela assigning no field symbol. iguale o fieldsymbol-nome\_do\_campo1 ao fieldsymbol-nome\_do\_campo2 dentro desse loop.

Jeito mais fácil de mover =)

**Thiago Losilla** — 12/08/2011 07:56

Ae Maurao!! Mto bom esse post. Classe de exceção é o futuro! OO é o Futuro!  
Quando possível, pode mostrar como funciona a exceção em funções?

Valeu Bro!!

Abrax!

**Matheus Tristão** — 03/01/2017 13:18

Thiago. Você faz com Raise. Primeiro você cria a exceção nas exceptions do function module. Na lógica da função, coloca um raise quando for pra disparar. Daí quando vc estiver no seu report e chamar a função você coloca depois do import e export uma seção EXCEPTIONS e iguala um número a cada exceção. Esse vai ser um sy-subrc. Exemplo:

```
CALL FUNCTION 'ZBC400_MOS_POWER'
EXPORTING
  iv_base = pa_int1
  iv_power = pa_int2
IMPORTING
  ev_result = gv_result
```

```
EXCEPTIONS
power_value_too_high = 1
result_value_too_high = 2
OTHERS = 3.
CASE sy-subrc.
WHEN 0.
WHEN 1.
WRITE 'Max Value Power is 4'.
WHEN 2.
WRITE 'Result value too high'.
WHEN 3.
WRITE 'Unknown error'.
ENDCASE.
```

**Mauro Laranjeira** — 12/08/2011 08:30

Grande Losilla FireFox =D

Cara vc me deu uma ideia de post, rs.

Vou te mostrar como fazer isso com tratamentos em classes o/

Abs,

Mauro Laranjeira