Desvendando o Gato do ALV – Veja como Funciona!

02/02/2011 07:30

O meu grande amigo Mauro, sabiamente, mostrou no post de ontem como funciona o **"Gato do ALV"** com o *código de game* para mostrar um relatório sobre a consistência de um alv.

Eu fiquei *"fuçando"* um pouco e descobri como a coisa toda funciona. Vou explicar à grosso modo, para simplificar as coisas:

– Quando você segura aqueles botões (SHIFT+ALT+CTRL) e dá o duplo clique, o framework da SAPGUI reage, transformando esse comando maluco em um "*sy-ucomm*" (aliás, eu acho que o framework faz isso para qualquer ação do usuário).

– Com esse *sy-ucomm* gerado, o SAP direciona o programa para um método da classe **CL_GUI_ALV_GRID** que se chama **CONSISTENCY_CHECK** (método PRIVATE)

– Esse método chama a função **ALV_CONSISTENCY_CHECK,** que mostra aquele log bonitão do post do Mauro, mostrando a consistência do ALV.

Parou parou….

Função?!?

Isso que dizer que eu posso codificar isso no meu relatório? 😮

Sim!

Fiz um ALV utilizando classes para exemplificar. Veja o código abaixo:

```
REPORT zaz_consistencia_alv.

* Objetos Para Referencia
DATA: o_alv    TYPE REF TO cl_gui_alv_grid,
      o_cont   TYPE REF TO cl_gui_docking_container,
      o_parent TYPE REF TO cl_gui_container.

* Tabela com os dados Mestres
DATA: t_flight TYPE TABLE OF sflight.

*----> Classe para tratar o evento de Duplo Clique do ALV

*---------------------------------------------------------------------*
*       CLASS lcl_event DEFINITION
*---------------------------------------------------------------------*
CLASS lcl_event DEFINITION.

  PUBLIC SECTION.
    METHODS handle_double_click
        FOR EVENT double_click OF cl_gui_alv_grid.

ENDCLASS.                    "lcl_event DEFINITION
*---------------------------------------------------------------------*
*       CLASS lcl_event IMPLEMENTATION
*---------------------------------------------------------------------*
CLASS lcl_event IMPLEMENTATION.

  METHOD handle_double_click.
```

```
*   Declarações do Método
    DATA: t_fcat TYPE lvc_t_fcat,
          wa_layout TYPE lvc_s_layo,
          v_variant TYPE disvariant,
          v_save TYPE char1.

*   Busca os valores do ALV que está sendo exibido
    o_alv->get_frontend_fieldcatalog( IMPORTING
                                      et_fieldcatalog = t_fcat ).

    o_alv->get_frontend_layout( IMPORTING
                                es_layout = wa_layout ).

    o_alv->get_variant( IMPORTING
                        es_variant = v_variant
                        e_save     = v_save ).

*   Função que mostra a consistência do seu ALV
    CALL FUNCTION 'ALV_CONSISTENCY_CHECK'
      EXPORTING
        i_save     = v_save
        is_layo    = wa_layout
        is_variant = v_variant
        it_fcat    = t_fcat
        it_outtab  = t_flight.

  ENDMETHOD.                    "handle_double_click

ENDCLASS.                    "lcl_event IMPLEMENTATION

* Esta declaração deve ficar após a declaração da Classe Local
DATA:  o_event  TYPE REF TO lcl_event.

*---------------------------------------------------------------------*
* Evento START-OF-SELECTION
*---------------------------------------------------------------------*
START-OF-SELECTION.

* Eu utilizei a SFLIGHT pois ela normalmente é usada nos exemplos dos
* programas de exemplo de ALVs (BCALV* na SE38) Se você quiser
* pode trocar para outra tabela.
  SELECT * FROM sflight INTO TABLE t_flight.

* Crie essa tela e descomente o módulo status_9000 OUTPUT.
  CALL SCREEN 9000.

*&---------------------------------------------------------------------*
*&      Module  STATUS_9000  OUTPUT
*&---------------------------------------------------------------------*
MODULE status_9000 OUTPUT.

* Técnica para você não ter que desenhar o container na tela nova
* Essa classe cria o container pra você, e depois nós referenciamos
* estre container para o objeto do ALV OO.

* Cria o Container para a tela do ALV
  CREATE OBJECT o_cont
    EXPORTING
      side      = cl_gui_docking_container=>dock_at_top
      repid     = sy-repid
      dynnr     = '9000'
      extension = 1000
    EXCEPTIONS
      OTHERS    = 6.

* Casting para o tipo genério
```

```
  o_parent = o_cont.

* Criia Instância do ALV, referenciando o Container gerado acima
  CREATE OBJECT o_alv
    EXPORTING
      i_parent = o_parent.

* Registra tratamento do Evento de Duplo Clique
  CREATE OBJECT o_event.
  SET HANDLER o_event->handle_double_click FOR o_alv.

* Exibe o ALV
  CALL METHOD o_alv->set_table_for_first_display
    EXPORTING
      i_structure_name = 'SFLIGHT'
    CHANGING
      it_outtab        = t_flight.

ENDMODULE.                    " STATUS_9000  OUTPUT
```

Importante: Para o código acima funcionar, você deve criar uma tela de número 9000 e descomentar o módulo do PBO (status_9000 OUTPUT).

Fora a criação da tela, é só dar ctrl+c ctrl+v num program local. Use o /H para ver como funciona!

O que vai acontecer:

Quando você rodar o programa, irá aparecer esse ALV sem status gui nem nada, mas não importa. Agora, você pode dar duplo clique em qualquer lugar do relatório...

... que ele vai mostrar a mesma consistência do post do Mauro. Você pode colocar essa consistência em qualquer lugar: num botão,

> numa ação por sy-ucomm do status_gui, em
> outro evento...

Pois é, **Gato desvendado!**

Você pode usar o post como exemplo para ALV OO também 🙂

Abraços a todos aqueles que destroem o sonho das crianças que acreditam em mágica (evil).

Comentários

> **Marcos** — *09/03/2021 17:37*
>
> olá, ajudou muito este post, queria saber se consigo ajustar este mesmo código para usar ele com uma linha selecionavel
> grande abraço

> **Fabiana Rocha** — *26/04/2018 15:18*
>
> Oi.... estou com um problema e não consigo uma solução.
> Ao executar o ALV, criar uma linha e entrar com os dados (sem dar enter) somente com tab e depois clicar no botão SALVAR, a linha criada fica com os dados da linha antes dela.
>
> Se tiver um erro, irá fazer a checagem mostrando o erro., acerto porém não dou ENTER e clico no SALVAR, dados gravados porém errados.
>
> como fazer para o programa entender o ENTER (que funciona corretamente) fazer a mesma coisa se não clicar

> > **Leonel Sanches** — *23/01/2019 10:07*
> >
> > Fabiana,
> >
> > Use o método check_changed_data do próprio classe do ALV que irá obter todas as alterações.
> > Ex: CALL METHOD o_alv->check_changed_data.