**ALV OO – Trabalhando com botões e eventos**

Sem dúvida, uma das atividades mais realizadas por um programador ABAP é a emissão de relatórios.

Como não estamos na era dos dinossauros, “não conhecemos” o relatório “write” e vamos direto aos “ALV’s”.

Este post tem como objetivo exemplificar algumas funcionalidades de um ALV OO (Orientado a Objetos),  são elas:

* Exclusão de botões do toolbar de um ALV OO
* Inclusão de botões no toolbar do ALV
* Utilização de eventos
* Como desabilitar um botão (ele aparece, mas não é possível clicar)
* Geração de um ALV sem necessidade de criação de um container

Veja também:

* Como encontrar ícones
* Utilização de interfaces

A modelagem utilizada, foi construída de forma a permitir a utilização dos pontos acima citados.

**Especificação**

Na tela de seleção, o usuário poderá determinar uma ou mais cias aéreas.  Deve ser gerado um relatório ALV com os dados desta(s) cia(s), a serem extraídos da tabela SCARR.

Neste relatório, ao selecionar uma determinada cia, deve haver um botão que gere um segundo ALV, desta vez exibindo as informações dos vôos (conexões) referentes à seleção realizada (dados a serem extraídos da tabela SPFLI).

E então, vamos ao que interessa!?

No programa principal, devemos criar a tela de seleção e um include, onde codificaremos nossas classes.

Tela de seleção:

Declara-se o select-options para possibilitar a determinação do range das cias aéreas:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | include: z\_classes.    tables: scarr,          spfli.    data: wa\_scarr type scarr.    data: r\_companies type ref to lcl\_companies.    selection-screen begin of block filter with frame title text-001.  select-options s\_carrid for wa\_scarr-carrid.  selection-screen end of block filter. |

[http://abap101.com/wp-content/uploads/2012/02/023-300x59.jpg](http://abap101.com/wp-content/uploads/2012/02/023.jpg)

Selection Screen

Vamos criar também, duas telas (as quais nomeei de “0100” e “0200”, que serão utilizadas na exibição dos ALV’s.

[http://abap101.com/wp-content/uploads/2012/02/161-300x61.jpg](http://abap101.com/wp-content/uploads/2012/02/161.jpg)

Screens

No include criado, segue declaração da interface:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | interface: zif\_data\_reader.    types: ty\_scarr type range of scarr-carrid.      methods: read\_data                importing                  im\_scarr type ty\_scarr,               generate\_alv                importing                  im\_structure  type char30.    endinterface.             "zif\_data\_reader |

Observe que temos os métodos read\_data e generate\_alv, que foram declarados na interface, porque serão utilizados em ambos os relatórios.

Agora vamos à definição da classe lcl\_companies, que será responsável pela leitura dos dados das cias aéreas.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | \*----------------------------------------------------------------------\*  \*       CLASS lcl\_companies DEFINITION  \*----------------------------------------------------------------------\*  \*  \*----------------------------------------------------------------------\*  class lcl\_companies definition.    public section.      interfaces zif\_data\_reader.        methods: constructor,                 set\_toolbar                  for event toolbar of cl\_gui\_alv\_grid                  importing e\_object,                 get\_ucomm                  for event user\_command of cl\_gui\_alv\_grid                  importing e\_ucomm,                 refresh\_table                  for event after\_user\_command                  of cl\_gui\_alv\_grid.      private section.      data: r\_grid        type ref to cl\_gui\_alv\_grid,            r\_connections type ref to lcl\_connections.        data: gt\_scarr      type table of scarr,            gt\_exc\_button type ui\_functions.    endclass.                                "lcl\_companies DEFINITION |

Observe que a classe tem como atributos privados:

* Objeto r\_grid (para criação do grid do alv)
* Objeto r\_connections para referência e criação dos objetos referentes aos vôos (conexões)
* Tabela gt\_scarr, contendo as informações da(s) cia(s) aérea;
* Tabela de botões a serem excluídos

Através da keyword “INTERFACES”, declaramos a utilização da interface zif\_data\_reader, com isso, na classe lcl\_companies *devemos* implementar todos os métodos contemplados na interface.

Seguem Implementações:

* Constructor: será responsável pela criação do grid do alv, registrar os eventos a serem utilizados e determinar os botões a serem excluídos.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | method: constructor.      create object r\_grid        exporting          i\_parent          = cl\_gui\_container=&gt;screen0        exceptions          error\_cntl\_create = 1          error\_cntl\_init   = 2          error\_cntl\_link   = 3          error\_dp\_create   = 4          others            = 5.        if sy-subrc ne 0.        message id sy-msgid type sy-msgty number sy-msgno                   with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.      endif.    \*--------------------------------------------------------------------\*  \* Registra os eventos a serem utilizados  \*--------------------------------------------------------------------\*      set handler get\_ucomm     for r\_grid.      set handler set\_toolbar   for r\_grid.      set handler refresh\_table for r\_grid.  \*--------------------------------------------------------------------\*  \* Informa os botões a serem excluídos  \*--------------------------------------------------------------------\*      append cl\_gui\_alv\_grid=&gt;mc\_fc\_graph to gt\_exc\_button.      endmethod.                              "constructor |

Nota: quando informamos “cl\_gui\_container=>screen0″ no parâmetro “i\_parent” não temos a necessidade de declarar um container (recomenda-se gerar o ALV sem container apenas em casos de real necessidade, pois esteticamente o programa será prejudicado, visto que o ALV ocupará a tela inteira).

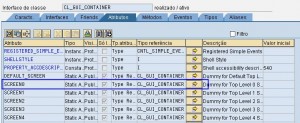
Para entender melhor, vá até a transação SE24, na classe CL\_GUI\_ALV\_GRID.

Na aba “métodos” você encontrará o “constructor”, clique em “Parâmetros” e verá que o parâmetro “i\_parent” é do tipo “CL\_GUI\_CONTAINER”.

[](http://abap101.com/wp-content/uploads/2012/02/22.jpg)

Parameter - "I\_PARENT"

Clicando em CL\_GUI\_CONTAINER, na aba atributos, você encontrará “Screen0″.

[](http://abap101.com/wp-content/uploads/2012/02/23.jpg)

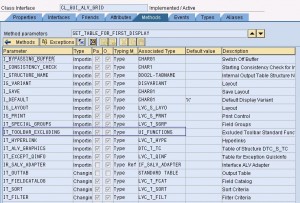
Attribute - Screen0

Ou seja, o atributo “Screen0″ utiliza a tela como nosso container.

**Entendendo a exclusão de botões**

Novamente na transação SE24, consulte o método “SET\_TABLE\_FOR\_FIRST\_DISPLAY”.

Ao clicar em “parâmetros” você encontrará o “IT\_TOOLBAR\_EXCLUDING”, de tipo “UI\_FUNCTIONS”.

[](http://abap101.com/wp-content/uploads/2012/02/toolbar_excluding.jpg)

Parameter - toolbar\_excluding

Vamos agora para a aba de atributos da classe, onde encontraremos os botões.

[](http://abap101.com/wp-content/uploads/2012/02/Bot%C3%B5es1.jpg)

CL\_GUI\_ALV\_GRID - Botões

Desta forma, basta inserirmos os botões a serem excluídos na tabela declarada, e informá-la no momento em que chamarmos o método “SET\_TABLE\_FOR\_FIRST\_DISPLAY” (neste exercício, excluí apenas o botão utilizado para geração de gráficos).

O próximo passo é selecionarmos os dados da tabela SCARR conforme dados da tela de seleção.

* zif\_data\_reader~read\_data

Observe que quando implementamos um método de uma interface, este é denominado: nome\_da\_interface**~**nome\_ do\_ método.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | method: zif\_data\_reader~read\_data.          clear gt\_scarr.      refresh gt\_scarr.        select \*        from scarr        into table gt\_scarr        where carrid in im\_scarr[].        if sy-subrc ne 0.        message: text-002 type 'E'.        else.        call method me-&gt;zif\_data\_reader~generate\_alv          exporting            im\_structure = 'SCARR'.      endif.    endmethod.                               "zif\_data\_reader~read\_data |

Neste método, selecionamos os dados conforme critérios informados na tela de seleção.

Quando sy-subrc não é “0” (valores não encontrados), exibimos uma mensagem de erro, caso contrário, chamamos o  método “zif\_data\_reader~generate\_alv”, vide implementação abaixo:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | method: zif\_data\_reader~generate\_alv.        call method r\_grid-&gt;set\_table\_for\_first\_display        exporting          i\_structure\_name              = im\_structure          it\_toolbar\_excluding          = gt\_exc\_button        changing          it\_outtab                     = gt\_scarr        exceptions          invalid\_parameter\_combination = 1          program\_error                 = 2          too\_many\_lines                = 3          others                        = 4.        if sy-subrc NE 0.        message id sy-msgid type sy-msgty number sy-msgno                   with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.      endif.        call screen 100.    endmethod.                               "zif\_data\_reader~generate\_alv |

Com os dados selecionados e exibidos na tela, precisamos incluir o botão para visualizarmos as conexões das cias aéreas.

* set\_toolbar – será chamado cada vez que o evento “TOOLBAR” (registrado no constructor!) for disparado pela classe CL\_GUI\_ALV\_GRID.

Consultando os parâmetros deste evento, na transação SE24, podemos verificar que ele exporta o parâmetro “E\_OBJECT”, de tipo  “CL\_ALV\_EVENT\_TOOLBAR\_SET”.

Nos atributos desta classe, encontraremos o “MT\_TOOLBAR”, de tipo “TTB\_BUTTON”, finalmente tipo “STB\_BUTTON”, onde estão as informações que precisamos.

Veja a implementação do método “set\_toolbar”:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | method: set\_toolbar.      data: wa\_toolbar type stb\_button.      case sy-dynnr.        when '1000'.            clear wa\_toolbar.          wa\_toolbar-butn\_type = 3.               "Separador          append wa\_toolbar to e\_object-&gt;mt\_toolbar.            clear wa\_toolbar.          wa\_toolbar-function  = '100\_CONN'.          wa\_toolbar-icon      = icon\_object\_folder.          wa\_toolbar-quickinfo = 'Visualiza Conexões'.          wa\_toolbar-butn\_type = 0.              "Botão Normal          wa\_toolbar-text      = 'Visualizar Conexões'.          append wa\_toolbar to e\_object-&gt;mt\_toolbar.          when '0100'.          clear wa\_toolbar.          wa\_toolbar-butn\_type = 3.              "Separador          append wa\_toolbar to e\_object-&gt;mt\_toolbar.            clear wa\_toolbar.          wa\_toolbar-function    = '100\_CONN'.          wa\_toolbar-icon        = icon\_object\_folder.          wa\_toolbar-quickinfo   = 'Visualiza Conexões'.          wa\_toolbar-butn\_type   = 0.              "Botão normal          wa\_toolbar-text        = 'Visualizar Conexões'.          if sy-ucomm            = 'BACK'.            wa\_toolbar-disabled  = ' '.          else.            wa\_toolbar-disabled  = 'X'.          endif.          append wa\_toolbar to e\_object-&gt;mt\_toolbar.        when others.      endcase.      endmethod.                                 "set\_toolbar |

Observe que temos duas tratativa diferentes para o botão: quando estamos na tela 1000 (chamando a tela 0100, que contém os dados das cias aéreas), quando deve estar habilitado, e quando estamos na tela 0100 (chamando a tela 0200, que conterá os dados das conexões), quando o botão deve estar desabilitado.

Para melhor compreensão, faça este exercício, coloque um BREAK-POINT no método e debugue! Assim você entenderá os momentos em que ele é chamado e sua tratativa!

Dica: lembre-se de utilizar text-symbols. Na implementação acima, escrevi as informações diretamente no código para facilitar o entendimento!

Nota: informamos o ícone: icon\_object\_folder, este nome é encontrado na tabela “ICON”, porém, algumas vezes precisamos do parâmetro “internal” quando trabalhamos com ícones. Este pode ser encontrado no programa “SHOWICON”.

* get\_ucomm – é através deste que nosso botão funcionará:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | method: get\_ucomm.      data: lv\_row   type lvc\_s\_row,            wa\_scarr like line of gt\_scarr.        data: r\_cias  type range of scarr-carrid,            wa\_cias like line of r\_cias.        if e\_ucomm eq '100\_CONN'.        call method r\_grid-&gt;get\_current\_cell          importing            es\_row\_id = lv\_row.    \*--------------------------------------------------------------------\*        if lv\_row is initial.          message: text-003 type 'I'.  \*--------------------------------------------------------------------\*        else.          read table gt\_scarr into wa\_scarr index lv\_row-index.            clear wa\_cias.          wa\_cias-sign    = 'I'.          wa\_cias-option  = 'EQ'.          wa\_cias-low     = wa\_scarr-carrid.  \*        wa\_cias-high   = ' ' .          append wa\_cias to r\_cias.            create object r\_connections            exporting              im\_grid = r\_grid.            call method r\_connections-&gt;zif\_data\_reader~read\_data            exporting              im\_scarr = r\_cias.  \*--------------------------------------------------------------------\*        endif.      endif.    endmethod.                             "get\_ucomm |

Lembrando que este método será chamado cada vez que o evento “user\_command” for disparado (conforme sua declaração), devemos verificar se o “sy-ucomm” é “100\_CONN”, ou seja, se nosso botão foi “clicado”.

Com isso, chamamos o método get\_current\_cel para verificarmos qual foi a linha selecionada.

Verificamos em nossa tabela qual é  cia aérea referente a linha selecionada, para então criarmos o objeto referente à classe das conexões e ler seus dados.

Observe que utilizamos o range r\_cias para informar a cia aérea. O range é como se fosse o select-options, por isso precisamos informar:

* sign      = “I” (Include)
* Option = “EQ” (Equal, porque informaremos uma única cia aérea.
* Low      = wa\_scarr-carrid (cia aérea da linha selecionada).
* High    = Não precisamos informar, pois utilizamos “option = EQ”.

Para a classe lcl\_companies, ainda precisamos implementar o método:

* refresh\_table:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | method refresh\_table.        check sy-dynnr = 100 and            sy-ucomm = 'BACK'.        call method me-&gt;zif\_data\_reader~generate\_alv        exporting          im\_structure = 'SCARR'.      endmethod.                              "refresh\_table |

Este método será chamado após o user\_command.

É necessário para atualizar as informações do grid quando voltamos da tela 200 para a tela 100. Sem ele, os dados das conexões permanecem na tela.

Para melhor compreensão, debugue!

Finalmente, vamos à declaração da classe “lcl\_connections”.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | \*----------------------------------------------------------------------\*  \*       CLASS lcl\_connections DEFINITION  \*----------------------------------------------------------------------\*  \*  \*----------------------------------------------------------------------\*  class lcl\_connections definition.    public section.      interfaces: zif\_data\_reader.        methods: constructor                  importing                    im\_grid type ref to cl\_gui\_alv\_grid.      private section.      data: gt\_spfli       type standard table of spfli,            gt\_exc\_button  type ui\_functions,            gv\_carrid      type scarr-carrid.        data: r\_grid\_conn type ref to cl\_gui\_alv\_grid.    endclass.                              "lcl\_connections DEFINITION |

Como podem ver, esta é uma classe bem simples. Verifiquem sua implementação:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | class lcl\_connections implementation.    method: constructor.      r\_grid\_conn = im\_grid.    endmethod.                              "constructor      method: zif\_data\_reader~read\_data.      clear   gt\_spfli.      refresh gt\_spfli.        select \*        from spfli        into table gt\_spfli        where carrid in im\_scarr.        if sy-subrc = 0.        call method me-&gt;zif\_data\_reader~generate\_alv          exporting            im\_structure = 'SPFLI'.      else.        message: text-004 type 'I'.      endif.      endmethod.                    "zif\_data\_reader~read\_data      method: zif\_data\_reader~generate\_alv.      data: wa\_exclude like line of gt\_exc\_button.        call method r\_grid\_conn-&gt;set\_table\_for\_first\_display        exporting          i\_structure\_name              = im\_structure        changing          it\_outtab                     = gt\_spfli        exceptions          invalid\_parameter\_combination = 1          program\_error                 = 2          too\_many\_lines                = 3          others                        = 4.        call screen 200.      endmethod.                    "zif\_data\_reader~generate\_alv    endclass. |

Observe que o constructor é responsável apenas por “importar” o grid gerado na classe lcl\_companies.

Vejam abaixo o resultado do nosso exercício:

[](http://abap101.com/wp-content/uploads/2012/02/tela-100.jpg)

Tela 0100 - Cias aéreas

Tela 0100, com os as informações das cias aéreas e botão habilitado.

Observe que não há o botão responsável pela geração de gráficos.

[http://abap101.com/wp-content/uploads/2012/02/tela-200-300x40.jpg](http://abap101.com/wp-content/uploads/2012/02/tela-200.jpg)

Tela 0200 - Conexões

Tela 0200, com informações das conexões e botão desabilitado.

E então, gostaram? baixe o código ([Z\_ALV\_OO](http://abap101.com/wp-content/uploads/2012/03/Z_ALV_OO_2.zip) ), estude e nos envie suas dúvidas e sugestões!