DE Award Submission: Best Productive Code - Timelessness: Process Object Layer uses Enumeration Classes

Name of the submission:

Process Object Layer uses Enumeration Classes

Name(s) of author(s):

Jan Bernhardt, Florian Dörr, Andreas Heinzmann, Hans-Martin Latuske, Marc-André Möller, Alexander Schmidt (D037602), Stephan Schub

Reason for the nomination (max 150 words):

Enumerations are a concept supported by many programming languages to identify a set of named values which behave usually like constants.

In ABAP the concept of enumerations is not supported. Therefor during development phase a number of constants for those values are defined on interfaces, classes, includes etc..
Mostly those are based on data elements and domains and unfortunately there are a number of those definitions in a system which heavily contradicts the “don’t repeat yourself” principle.

This can be avoided by Enumeration Classes based on ABAP Classes which uses a number of ABAP specific language elements (aka idioms), e.g. CLASS_CONSTRUCTOR, CREATE PRIVATE, READ-ONLY attributes.

Advantages:

- Only 1 place where the values can be defined
- Only the defined values can be used
- Type safety
- Easy to check if a value is valid (= IS BOUND)
- As these are regular ABAP classes, additional functionality can be added

Instructions on where and how to find the software (e.g. ABAP system name, perforce repository, class name, function name, etc.):

ABAP system name: RPM(200)

ABAP Package: /PL1/DESIGN_TIME & sub packages

Enumeration Classes: All classes following naming convention /PL1/CE_*, e.g. /PL1/CE_AUTHORIZATION_ACTIVITY from package /PL1/DT_GENERAL

Code snippets (that demonstrate the quality of the nomination):

Enumeration Class Definition:

```
class /pl1/ce_authorization_activity definition
  public
  final
  create private

  global friends /pl1/ct_abap_unit .
```

```
public section.

  types ty_description type string .
  types:
    tab type standard table of ref to /pl1/ce_authorization_activity with default key .
  types ty_value type /pl1/authorization_activity .

  class-data display type ref to /pl1/ce_authorization_activity read-only .
  class-data change type ref to /pl1/ce_authorization_activity read-only .
  class-data create type ref to /pl1/ce_authorization_activity read-only .

  class-methods class_constructor .
  class-methods get_all_instances
    returning
      value(r_instances) type tab .
  class-methods get_instance_by_value
    importing
      !i_value type ty_value
    returning
      value(r_instance) type ref to /pl1/ce_authorization_activity .
  methods get_value
    returning
      value(r_value) type ty_value .
  methods get_description
    returning
      value(r_description) type string .
protected section.
private section.

  types:
    begin of ty_instance_by_value,
      value    type ty_value,
      instance type ref to /pl1/ce_authorization_activity,
    end of ty_instance_by_value .
  types:
    ty_instances_sorted_by_value type sorted table of ty_instance_by_value with unique key value .

  constants:
    begin of con_activity,
      begin of display,
        id type ty_value value '01',
      end of display,
      begin of change,
        id  type ty_value value '02',
      end of change,
      begin of create,
        id  type ty_value value '03',
      end of create,
     end of con_activity .
  interface /pl1/if_domain_helper load .
  constants con_domain_name type /pl1/if_domain_helper=>ty_domain_name value '/PL1/AUTHORIZATION_ACTIVITY'. "#EC NOTEXT
  data mv_value type ty_value .
  data mv_description type ty_description .
  class-data st_buffer type tab .
  class-data st_buffer_by_value type ty_instances_sorted_by_value .
endclass.
```

Enumeration Implementation (Excerpt):

```
class /pl1/ce_authorization_activity implementation.
method class_constructor.
  data: lr_domain_helper     type ref to /pl1/if_domain_helper,
        ls_instance_by_value type ty_instance_by_value.

  define mac_create_enum.
    create object &1.
    ls_instance_by_value-instance = &1.
    &1->mv_value = ls_instance_by_value-value = con_activity-&1-id.
    &1->mv_description = lr_domain_helper-
>get_description_for_value( i_domain_name = con_domain_name
                                                                      i_val
ue       = '' && con_activity-&1-id ).
    append &1 to st_buffer.
    insert ls_instance_by_value into table st_buffer_by_value.
  end-of-definition.

  lr_domain_helper = /pl1/cl_domain_helper=>create( ).

  mac_create_enum: display, change, create.
endmethod.

method get_all_instances.
  r_instances = st_buffer.
endmethod.

method get_description.
  r_description = mv_description.
endmethod.

method get_instance_by_value.
  data: ls_instance_by_value type ty_instance_by_value.

  read table st_buffer_by_value with table key value = i_value into ls_inst
ance_by_value.
  r_instance = ls_instance_by_value-instance.
endmethod.

method get_value.
  r_value = mv_value.
endmethod.
endclass.
```