

SAP ABAP Programozás Alapjai

Harmadik gyakorlat feladatainak egy lehetséges megoldása

A kiadott jegyzetben a feladatok egy lehetséges megoldását mutatom be. A feladat megoldásai során a gyakorlaton használt elnevezési koncepciótól eltértem, de az könnyen leképezhető a hallgató által használtra.

A megoldások ismertetése során a forráskódot angol változónevekkel és angol nyelvű kommentekkel láttam el. A gyakorlat felépítéséhez hasonlóan a feladatok egymásra épülése miatt a magyarázó kommenteket is egymásra épülve ismertetem, azaz egy már ismert magyarázatot nem feltétlenül ismertetek még egyszer egy nagyobb számú feladatnál.

A feladathoz készített implementációs tervet (osztály diagramm) a gyakorlat során közösen alakítottuk ki.

Minden megoldás során az osztály forráskódját és a konzolra írt kimenetet ismertetem!

Tartalomjegyzék

1. Feladat – University osztály és kivétel osztály létrehozása	2
Exception class.....	2
University interface	3
University class.....	4
2. Feladat – Student osztály létrehozása	5
3. Feladat – Szemeszter osztály létrehozása	6
4. Feladat – Teszt konzol alkalmazás létrehozása	6

1. Feladat – University osztály és kivétel osztály létrehozása

Exception class

Az exception class a CX_STATIC_CHECK osztályból származtatva került létrehozásra. Ha az ADT eszközben már létrehozáskor megadásra kerül az őszosztály, akkor további implementált interfészek és a konstruktor kódja is a generált forráskód része lesz.

The screenshot shows the 'New ABAP Class' dialog box. The 'Name' field is 'ZCX_CL_UNI_2026_UNIVERSITY', the 'Description' is 'University exceptions', and the 'Superclass' is 'CX_STATIC_CHECK'. The 'Interfaces' section is empty. The 'Next >' button is highlighted.

```
CLASS zcx_cl_uni_2026_university DEFINITION
PUBLIC
INHERITING FROM cx_static_check
FINAL
CREATE PUBLIC .

PUBLIC SECTION.

INTERFACES if_t100_dyn_msg .
INTERFACES if_t100_message .

METHODS constructor
```

```

    IMPORTING
        !textid LIKE if_t100_message=>t100key OPTIONAL
        !previous LIKE previous OPTIONAL .
    PROTECTED SECTION.
    PRIVATE SECTION.
ENDCLASS.

CLASS zcx_cl_uni_2026_university IMPLEMENTATION.
    METHOD constructor ##ADT_SUPPRESS_GENERATION.
        CALL METHOD super->constructor
        EXPORTING
            previous = previous.
    CLEAR me->textid.
    IF textid IS INITIAL.
        if_t100_message~t100key = if_t100_message=>default_textid.
    ELSE.
        if_t100_message~t100key = textid.
    ENDIF.
    ENDMETHOD.
ENDCLASS.

```

University interface

A university interface-t a gyakorlaton interatívan, később került bevezetésre.

```

INTERFACE zmk_if_uni_2026_university
    PUBLIC .

    METHODS get_semester
        IMPORTING iv_semester_code TYPE zmk_a190pl_semester_code
        RETURNING VALUE(ro_semester) TYPE REF TO zmk_cl_uni_2026_semester
        RAISING zcx_cl_uni_2026_university.

    METHODS create_semester
        IMPORTING iv_semester_code TYPE zmk_a190pl_semester_code
            iv_semester_name TYPE zmk_a190pl_semester_name OPTIONAL
        RETURNING VALUE(ro_semester) TYPE REF TO zmk_cl_uni_2026_semester
        RAISING zcx_cl_uni_2026_university.

    METHODS create_student
        IMPORTING iv_student_name TYPE zmk_a190pl_student_name OPTIONAL
        RETURNING VALUE(ro_student) TYPE REF TO zmk_cl_uni_2026_student
        RAISING zcx_cl_uni_2026_university.
ENDINTERFACE.

```

University class

```
CLASS zmk_cl_uni_2026_university DEFINITION
PUBLIC
FINAL
CREATE PUBLIC .

PUBLIC SECTION.

    INTERFACES zmk_if_uni_2026_university.

PROTECTED SECTION.
PRIVATE SECTION.
ENDCLASS.

CLASS zmk_cl_uni_2026_university IMPLEMENTATION.

    METHOD zmk_if_uni_2026_university~get_semester.

        " Get semester data from database
        " Instead of direct database read a private method would be an
option
        " also reuse of the function modules from the previous exercise
        SELECT SINGLE *
            FROM zmk_a190pl_sms
            WHERE code = @iv_semester_code
            INTO @DATA(ls_semester_data).
        IF sy-subrc NE 0.
            " Error, semester does not exists in the database -> raise
exception defined in the interface
            RAISE EXCEPTION TYPE zcx_cl_uni_2026_university.
        ELSE.
            ro_semester = NEW #( is_semester_data = CORRESPONDING
#( ls_semester_data ) ).
        ENDIF.

    ENDMETHOD.

    METHOD zmk_if_uni_2026_university~create_semester.

        " First check if semester already exist in the database or not
        TRY.
            DATA(ls_semester) =
zmk_if_uni_2026_university~get_semester( iv_semester_code =
iv_semester_code ).
        CATCH zcx_cl_uni_2026_university.
            " Nothing to do here
```

```

ENDTRY.

IF ls_semester IS NOT INITIAL.
    RAISE EXCEPTION TYPE zcx_cl_uni_2026_university.
ENDIF.

" Semester does not exists -> create it
ro_semester = NEW #(
                VALUE #(
                    code = iv_semester_code
                    name = iv_semester_name ) ).

ENDMETHOD.

METHOD zmk_if_uni_2026_university~create_student.

    " From this point this was declared as homework
    ro_student = NEW #( ).

ENDMETHOD.

ENDCLASS.

```

2. Feladat – Student osztály létrehozása

A hallgató osztálynál a globális barát mechanizmust mutattuk be, az osztály teljes implementálása egyéni hallgatói feladat volt.

```

CLASS zmk_cl_uni_2026_student DEFINITION
    PUBLIC
    FINAL
    CREATE PRIVATE
    GLOBAL FRIENDS zmk_cl_uni_2026_university .
    .

    PUBLIC SECTION.
    PROTECTED SECTION.
    PRIVATE SECTION.
ENDCLASS.

CLASS zmk_cl_uni_2026_student IMPLEMENTATION.
ENDCLASS.

```

3. Feladat – Szemeszter osztály létrehozása

A hallgató osztálynál a globális barát mechanizmust, illetve a private constructort mutattuk be, az osztály teljes implementálása egyéni hallgatói feladat volt.

```
CLASS zmk_cl_uni_2026_semester DEFINITION
    PUBLIC
    FINAL
    CREATE PRIVATE
    GLOBAL FRIENDS zmk_cl_uni_2026_university .

    PUBLIC SECTION.
    PROTECTED SECTION.
    PRIVATE SECTION.

        DATA: semester_data TYPE zmk_a190pl_s_semester.

        METHODS constructor
            IMPORTING is_semester_data TYPE zmk_a190pl_s_semester.

ENDCLASS.

CLASS zmk_cl_uni_2026_semester IMPLEMENTATION.
    METHOD constructor.
        semester_data = is_semester_data.
    ENDMETHOD.
ENDCLASS.
```

4. Feladat – Teszt konzol alkalmazás létrehozása

```
CLASS zmk_cl_uni_2026_university_tst DEFINITION
    PUBLIC
    FINAL
    CREATE PUBLIC .

    PUBLIC SECTION.

        INTERFACES if_oo_adt_classrun .
    PROTECTED SECTION.
    PRIVATE SECTION.

ENDCLASS.

CLASS zmk_cl_uni_2026_university_tst IMPLEMENTATION.
```

```

METHOD if_oo_adt_classrun~main.

" Referencia változó
DATA university TYPE REF TO zmk_if_uni_2026_university.

out->write( 'hello' ).

" Példány létrehozása
university = NEW zmk_cl_uni_2026_university( ).

" Még nem létező ID-val szemeszter létrehozása
TRY.
    DATA(semester_1) = university->create_semester(
        EXPORTING
            iv_semester_code = '9999'
    ).
    out->write( `New semester has been created` ).
CATCH zcx_cl_uni_2026_university.
    " Nothing to do, this is the expected behaviour
ENDTRY.

" Már létező ID-val szemeszter létrehozása
TRY.
    DATA(semester_2) = university->create_semester(
        EXPORTING
            iv_semester_code = '2025202601'
    ).
CATCH zcx_cl_uni_2026_university.
    out->write( `As expected - this semester cannot be created
again` ).
ENDTRY.

ENDMETHOD.
ENDCLASS.

```