ABAP Unit Testing

Zapping bugs in ERP and BTP

Giriş

Unit Test yazan kaç kişi var?

…ve en genci kim?

Unit Test Nedir?

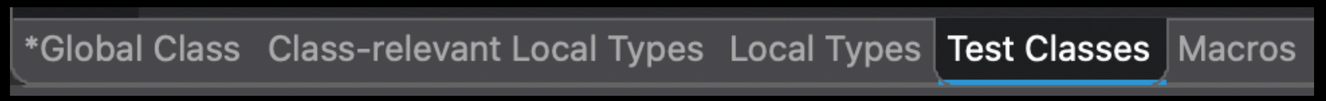
- Kodu test eden kod yazmak
- Pek çok dilde var
- ABAP Unit Framework: Sınıf tabanlı
- Hem ERP hem BTP

Faydaları

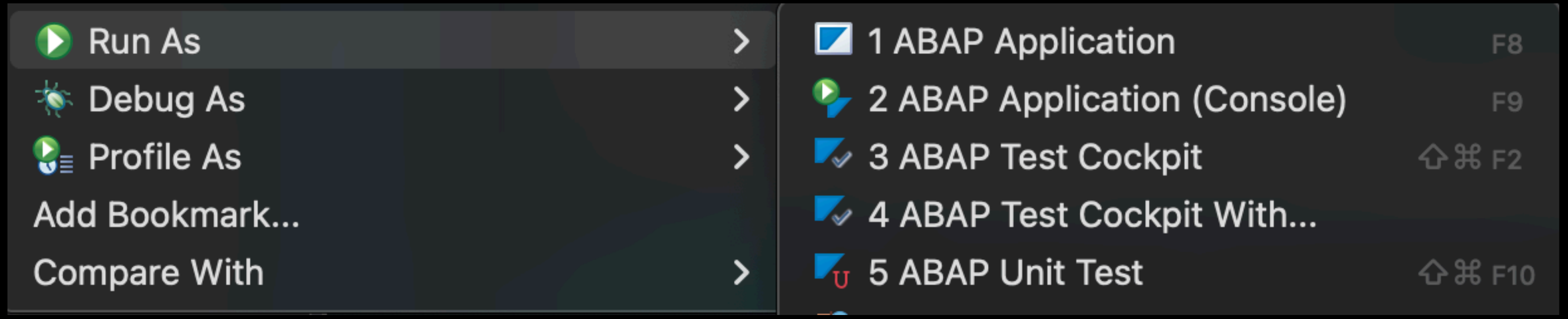
- Mimari iyileşme
- Geliştirmede Bug’ların yakalanması
- Değişikliklerde Bug’ların yakalanması
- Teknik dokümantasyon

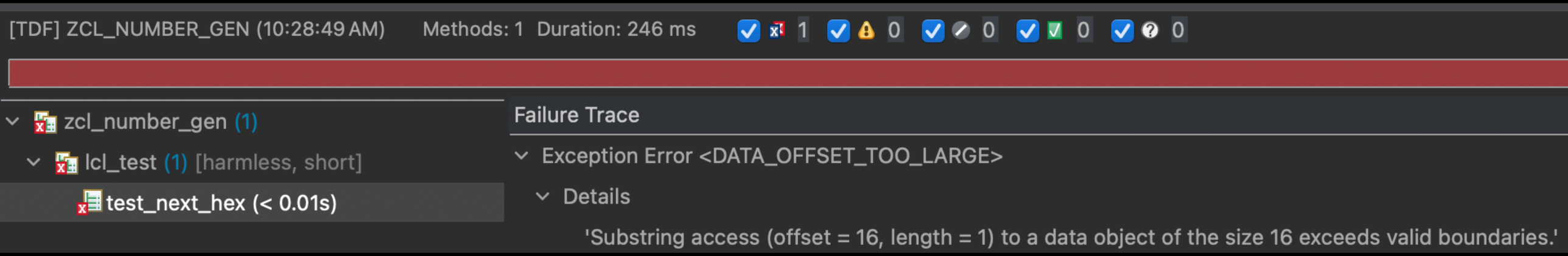
Unit Test Demo

```
CLASS zcl_number_gen DEFINITION
  PUBLIC FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    METHODS get_next_hex
      IMPORTING prev          TYPE char1
      RETURNING VALUE(result) TYPE char1.

  PRIVATE SECTION.
    CONSTANTS hexes TYPE char16 VALUE '0123456789ABCDEF'.
ENDCLASS.


CLASS zcl_number_gen IMPLEMENTATION.
  METHOD get_next_hex.
    DATA hex_pos TYPE i.

    FIND FIRST OCCURRENCE OF prev IN me->hexes MATCH OFFSET hex_pos.
    CHECK sy-subrc = 0.
    hex_pos = hex_pos + 1.
    CHECK hex_pos <= strlen( me->hexes ).
    result = me->hexes+hex_pos(1).
  ENDMETHOD.
ENDCLASS.
```

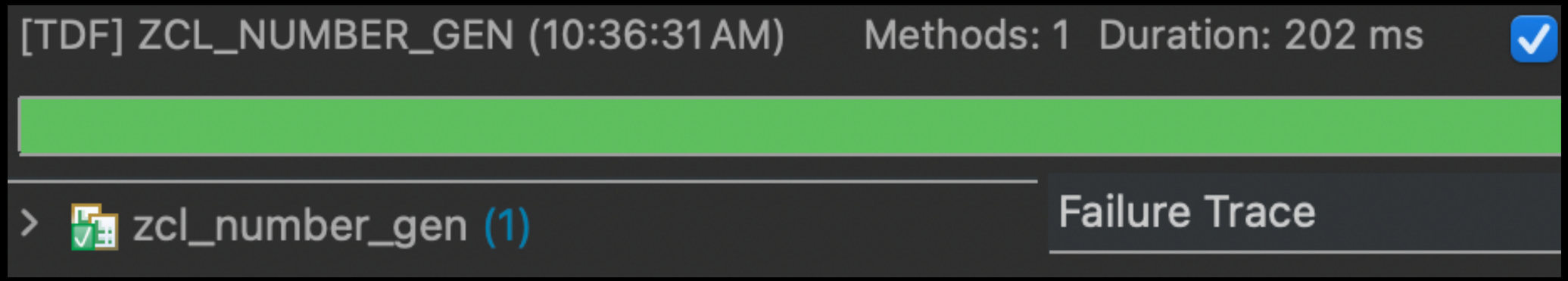

*Global Class
Class-relevant Local Types
Local Types
Test Classes
Macros

```
CLASS lcl_test DEFINITION
  FOR TESTING RISK LEVEL HARMLESS DURATION SHORT.

  PRIVATE SECTION.
    METHODS test_next_hex FOR TESTING.

ENDCLASS.


CLASS lcl_test IMPLEMENTATION.
  METHOD test_next_hex.
    DATA(gen) = NEW zcl_number_gen( ).

    DATA(next) = gen->get_next_hex( '1' ).

    cl_abap_unit_assert=>assert_equals( exp = '2'
                                        act = next ).

    next = gen->get_next_hex( '6' ).

    cl_abap_unit_assert=>assert_equals( exp = '7'
                                        act = next ).

    next = gen->get_next_hex( 'F' ).

    cl_abap_unit_assert=>assert_equals( exp = space
                                        act = next ).

    next = gen->get_next_hex( 'Z' ).

    cl_abap_unit_assert=>assert_equals( exp = space
                                        act = next ).
  ENDMETHOD.
ENDCLASS.
```


Run As
Debug As
Profile As
Add Bookmark...
Compare With
1 ABAP Application F8
2 ABAP Application (Console) F9
3 ABAP Test Cockpit ⇧⌘ F2
4 ABAP Test Cockpit With...
5 ABAP Unit Test ⇧⌘ F10

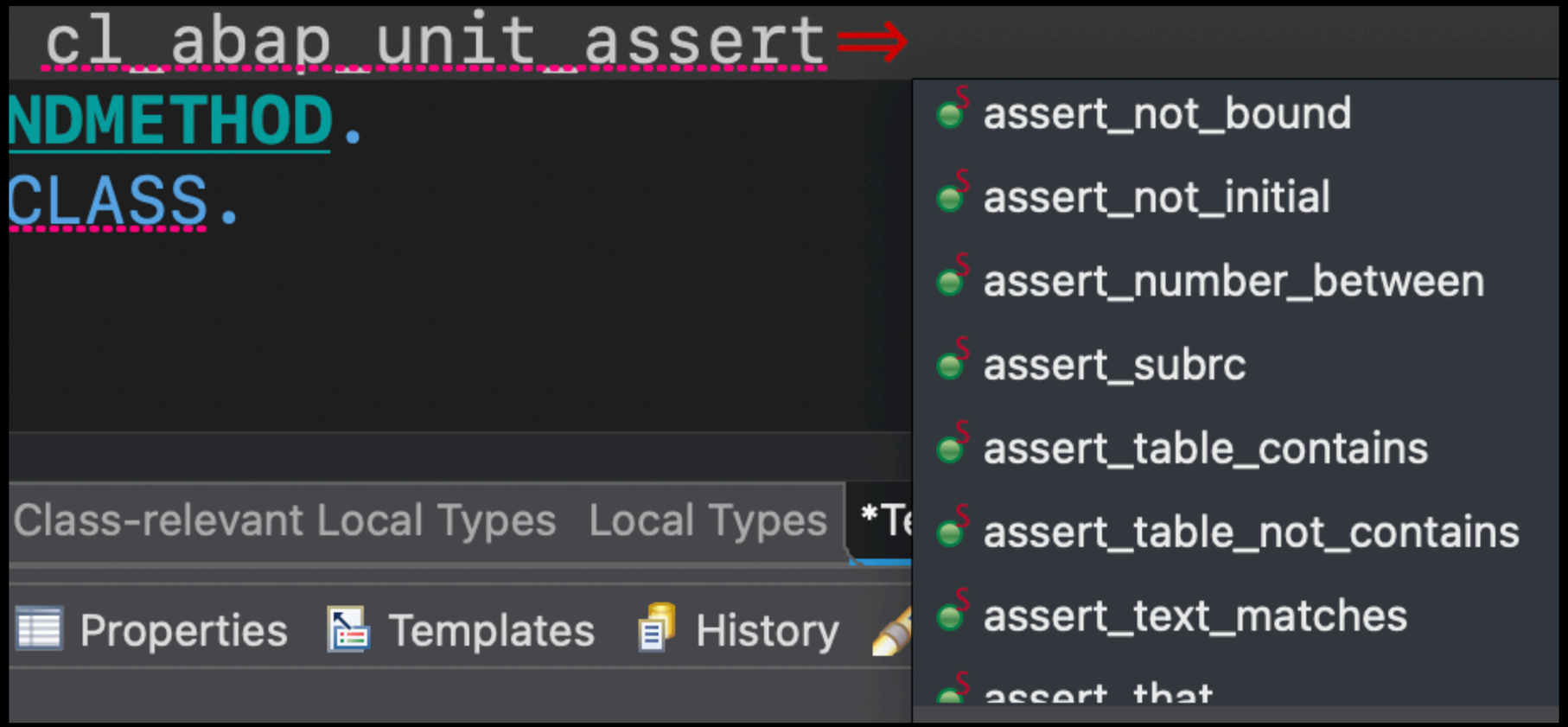

[TDF] ZCL_NUMBER_GEN (10:28:49 AM) Methods: 1 Duration: 246 ms
1
0
0
0
0
zcl_number_gen (1)
lcl_test (1) [harmless, short]
test_next_hex (< 0.01s)
Failure Trace
Exception Error <DATA_OFFSET_TOO_LARGE>
Details
'Substring access (offset = 16, length = 1) to a data object of the size 16 exceeds valid boundaries.'

```
CHECK hex_pos < strlen( me->hexes ).
```


[TDF] ZCL_NUMBER_GEN (10:36:31 AM)
Methods: 1
Duration: 202 ms
zcl_number_gen (1)
Failure Trace

Unit Framework Özellikleri

Assert Zenginliği

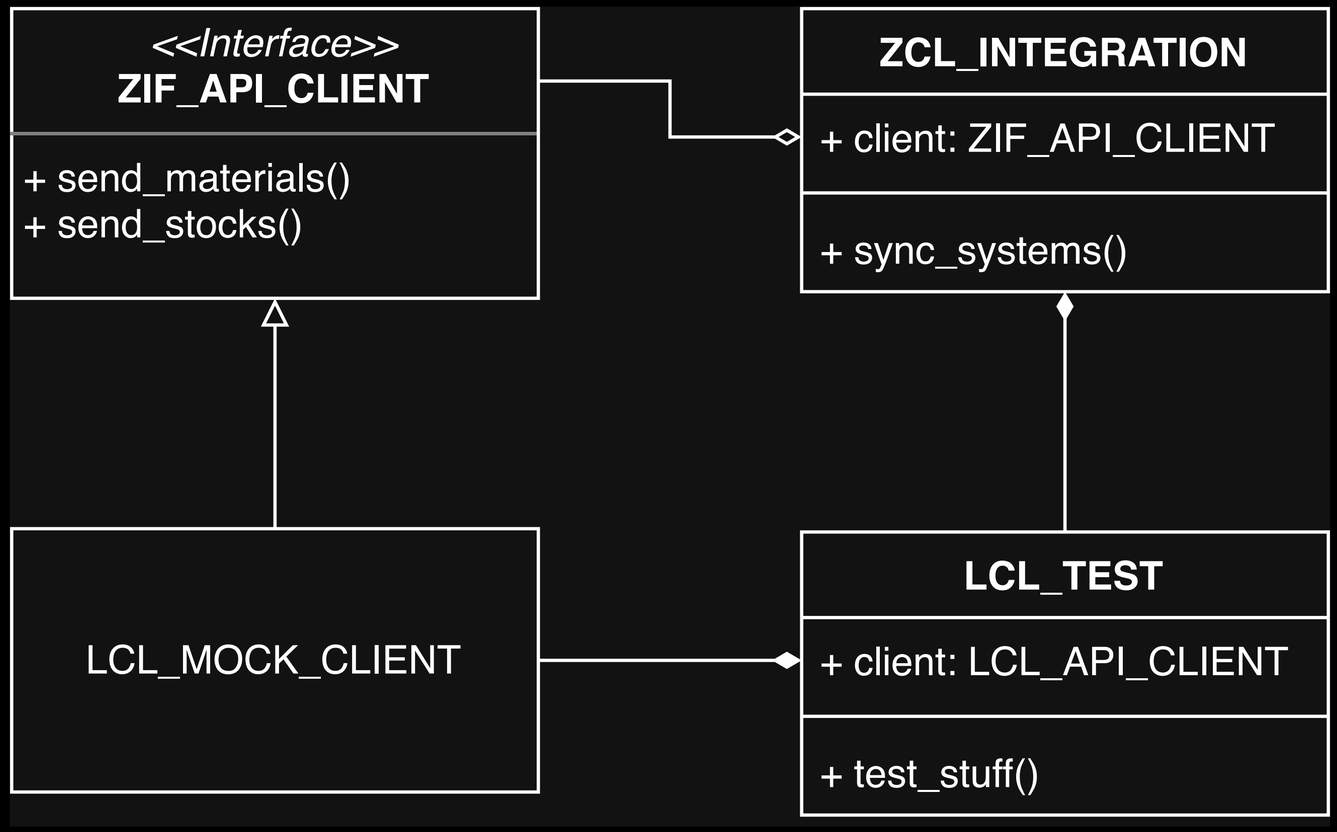


Özel Yordamlar

Yordam	Çalışma anı
class_setup()	Tüm testlerden önce
setup()	Her bir testten önce
teardown()	Her bir testten sonra
class_teardown()	Tüm testlerden sonra

Mocking Yöntemleri

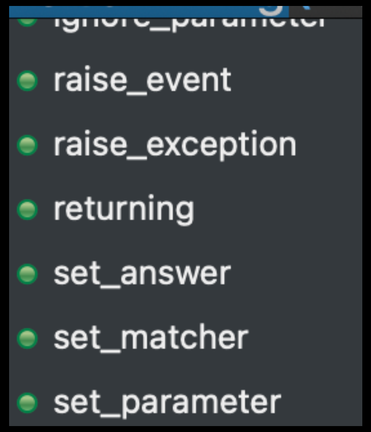
Dependency Injection



ABAP Test Double

```
INTERFACE zif_api_client
  PUBLIC.

  METHODS send_materials EXPORTING success TYPE abap_bool.
  METHODS send_stocks    EXPORTING success TYPE abap_bool.
ENDINTERFACE.
```

```
CLASS zcl_integration DEFINITION
  PUBLIC FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    DATA api_client TYPE REF TO zif_api_client.

    METHODS sync_systems EXPORTING success TYPE abap_bool.
ENDCLASS.


CLASS zcl_integration IMPLEMENTATION.
  METHOD sync_systems.
    me->api_client->send_materials( IMPORTING success = DATA(suc1) ).
    me->api_client->send_stocks( IMPORTING success = DATA(suc2) ).
    success = xsdbool( suc1 = abap_true AND suc2 = abap_true ).
  ENDMETHOD.
ENDCLASS.
```




```
CLASS lcl_test IMPLEMENTATION.
  METHOD test_stuff.
    DATA(mock_api) = CAST zif_api_client( cl_abap_testdouble=>create( 'ZIF_API_CLIENT' ) ).
    cl_abap_testdouble=>configure_call( mock_api )->returning( abap_true ).
    mock_api->send_stocks( ).

    NEW zcl_integration( )->sync_systems( IMPORTING success = DATA(success) ).

    cl_abap_unit_assert=>assert_true( success ).
  ENDMETHOD.
ENDCLASS.
```

Test Seam

```
METHOD delete_user.
  " ABAP, ABAP, ABAP, ABAP

  TEST-SEAM user_deletion.
    CALL FUNCTION 'BAPI_USER_DELETE'
      EXPORTING username = user.

    success = abap_true.
  END-TEST-SEAM.

  " ABAP, ABAP, ABAP, ABAP
ENDMETHOD.
```

```
METHOD test3.
  TEST-INJECTION user_deletion.
    success = abap_true. " Silinmiş gibi yap
  END-TEST-INJECTION.

  " TEST ABAP TEST ABAP TEST ABAP
ENDMETHOD.
```

Final

Öneriler

- Test Driven Development (Mimari + Neg -> Code -> Pos)
- Public mekanik kısımlar test edilmeli
- Private test ihtiyacı = Yeni sınıf
- ITAB veya Z'li tablodan Test Case alınabilir
- BTP: Joule Unit Test Generation (başlangıç noktası)

ABAP Unit Testing

Soru & Cevap