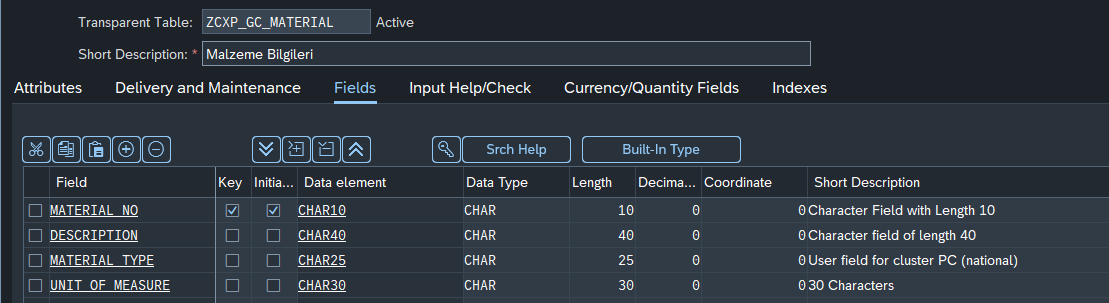
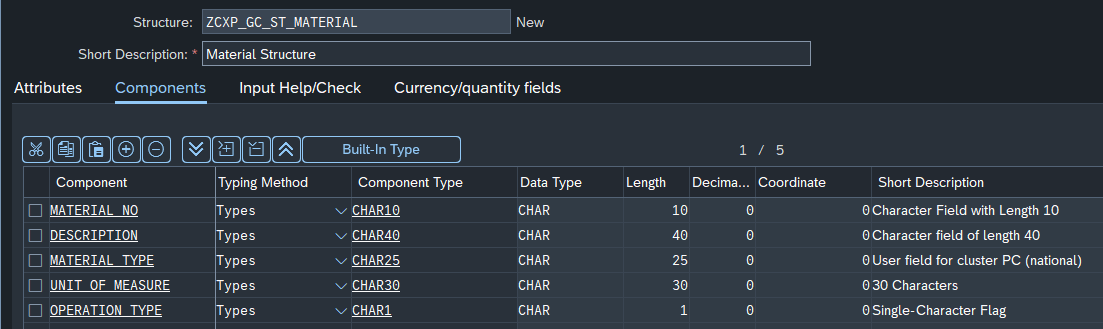
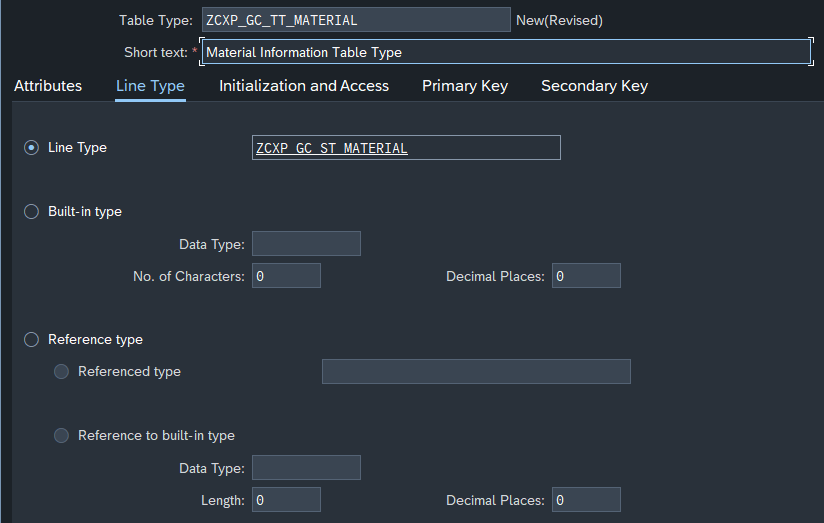
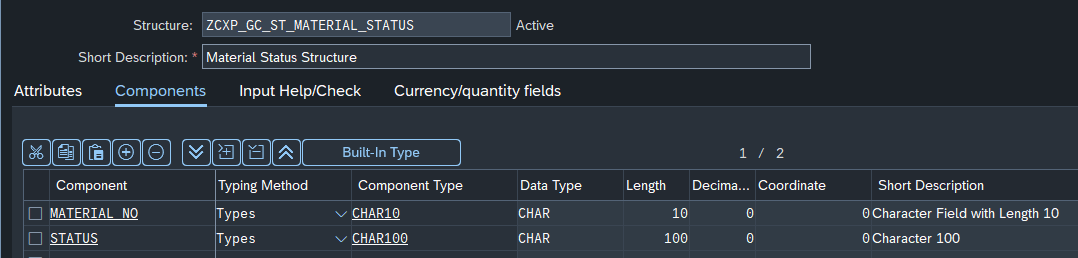
1.

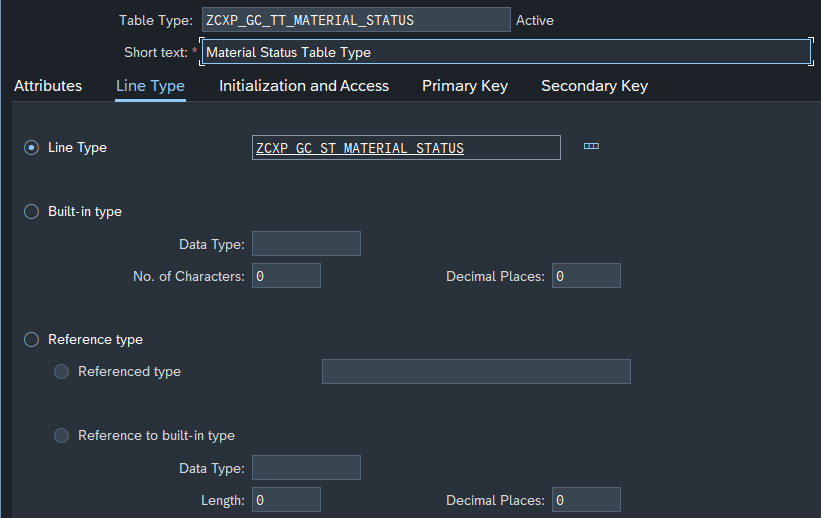


2. ‘se11’ -> ‘data type’ -> structure olarak oluşturuldu.

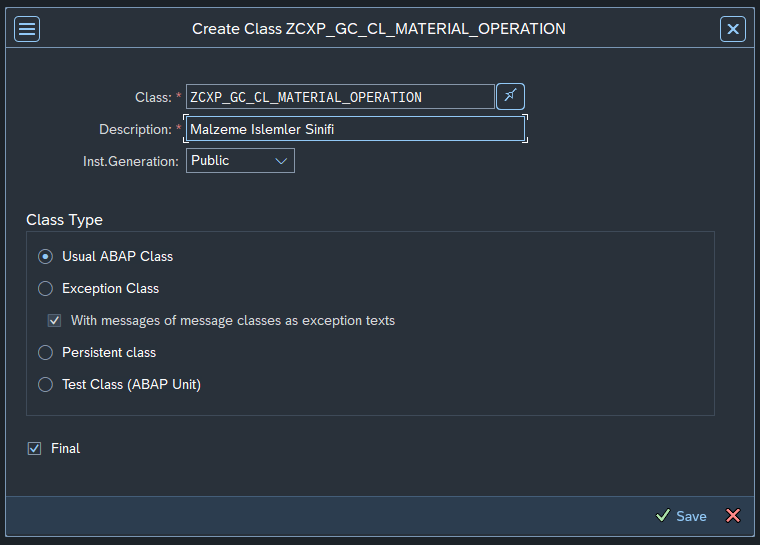


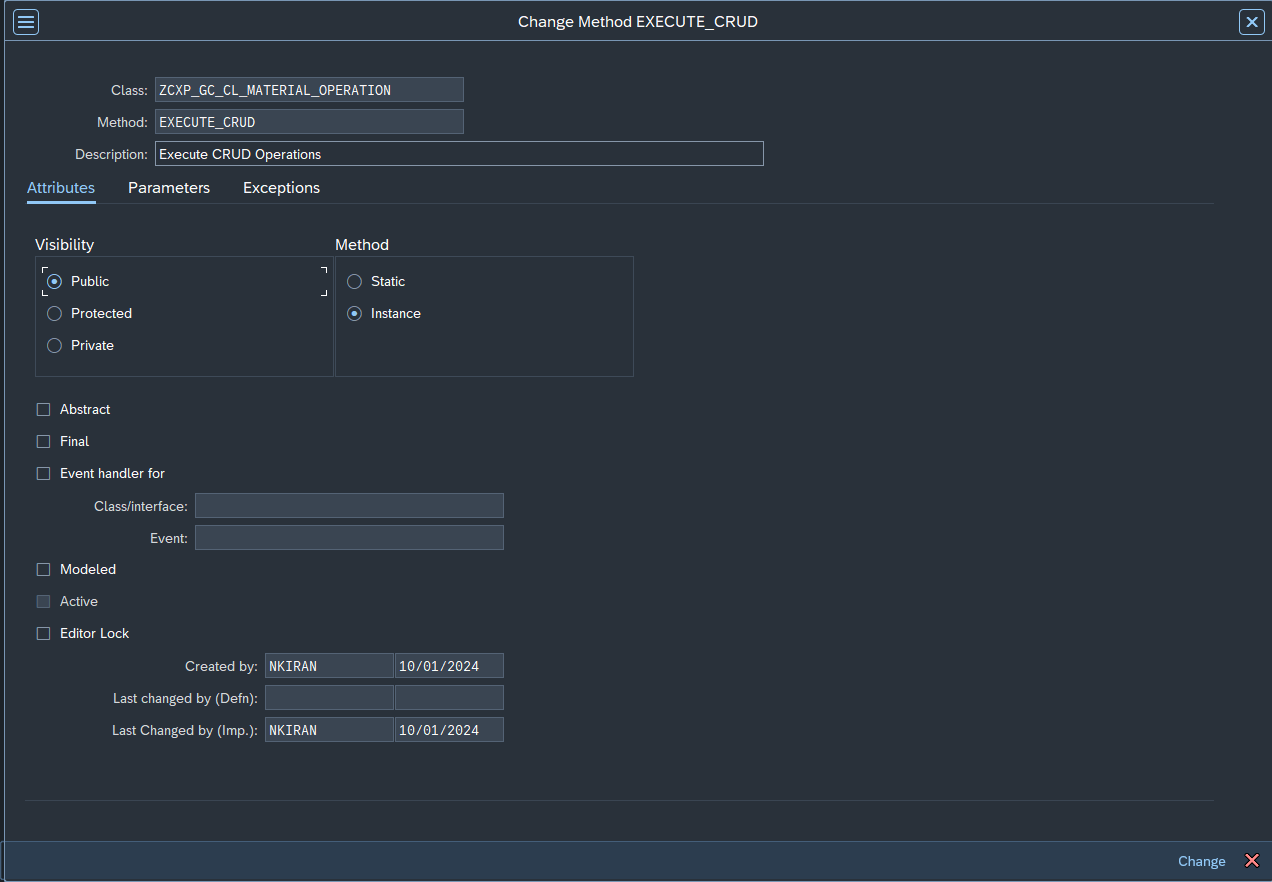


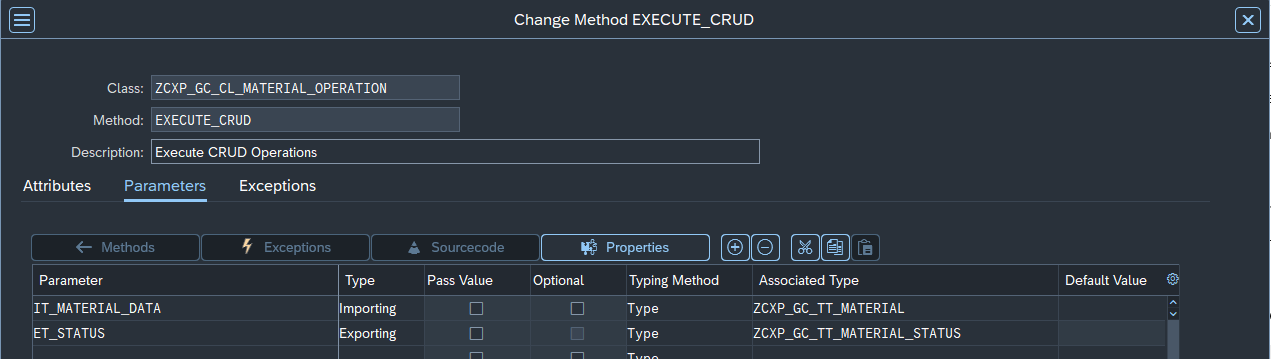




3. ‘se24’ -> class





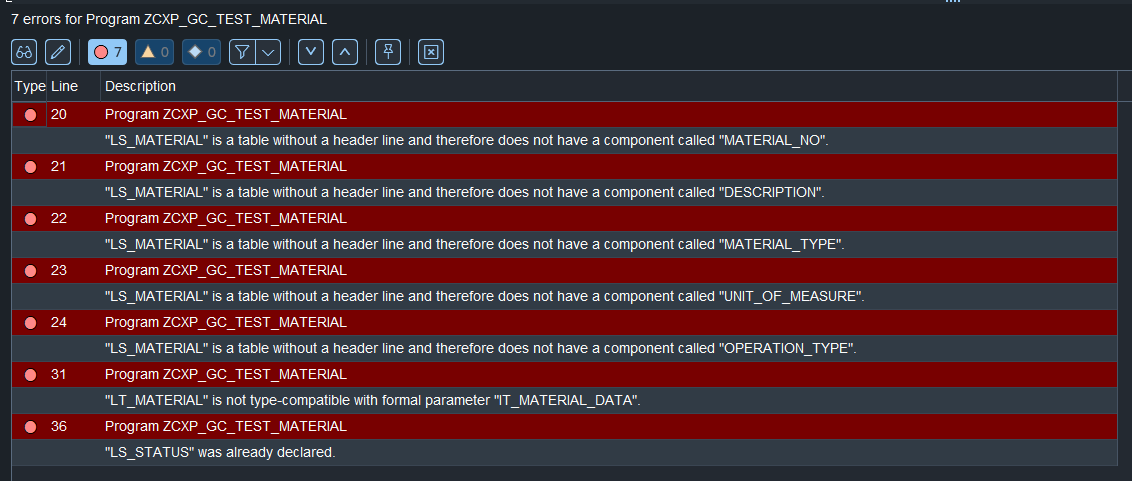


Method Implementation Bölümü

|  |
| --- |
| METHOD execute\_crud.       *" İç tablo ve çıktı tablosu tanımlamaları*       DATA: lv\_material\_no TYPE zcxp\_gc\_material-material\_no,             lv\_status      TYPE zcxp\_gc\_st\_material\_status-status.        LOOP AT it\_material\_data INTO DATA(ls\_material).         *" İşlem tipine göre farklı CRUD işlemleri yapacağız.*          CASE ls\_material-operation\_type.            WHEN 'C'. *" Create*             INSERT INTO zcxp\_gc\_material VALUES ls\_material.             IF sy-subrc = 0.               lv\_status = 'Material created successfully'.             ELSE.               lv\_status = 'Error creating material'.             ENDIF.            WHEN 'R'. *" Read*             SELECT SINGLE \* INTO CORRESPONDING FIELDS OF ls\_material               FROM zcxp\_gc\_material               WHERE material\_no = ls\_material-material\_no.             IF sy-subrc = 0.               lv\_status = 'Material read successfully'.             ELSE.               lv\_status = 'Material not found'.             ENDIF.            WHEN 'U'. *" Update*             UPDATE zcxp\_gc\_material SET               description = @ls\_material-description,               material\_type = @ls\_material-material\_type,               unit\_of\_measure = @ls\_material-unit\_of\_measure             WHERE material\_no = @ls\_material-material\_no.             IF sy-subrc = 0.               lv\_status = 'Material updated successfully'.             ELSE.               lv\_status = 'Error updating material'.             ENDIF.            WHEN 'D'. *" Delete*             DELETE FROM zcxp\_gc\_material WHERE material\_no = ls\_material-material\_no.             IF sy-subrc = 0.               lv\_status = 'Material deleted successfully'.             ELSE.               lv\_status = 'Error deleting material'.             ENDIF.             *" Geçersiz İşlem Tipi*           WHEN OTHERS.             lv\_status = 'Invalid operation type'.         ENDCASE.          *" Status output*         APPEND VALUE #( material\_no = ls\_material-material\_no                         status = lv\_status ) TO et\_status.       ENDLOOP.      ENDMETHOD. |

* + **TEST**

|  |
| --- |
| *\*&---------------------------------------------------------------------\** *\*& Report ZCXP\_GC\_TEST\_MATERIAL* *\*&---------------------------------------------------------------------\** *\*&* *\*&---------------------------------------------------------------------\** REPORT zcxp\_gc\_test\_material.  DATA: lt\_material TYPE TABLE OF zcxp\_gc\_tt\_material,    *" Malzeme tablosu (IT\_MATERIAL\_DATA)*       ls\_material TYPE zcxp\_gc\_tt\_material,      *" Malzeme yapısı*       lt\_status   TYPE TABLE OF zcxp\_gc\_tt\_material\_status,  *" Durum tablosu (ET\_STATUS)*       ls\_status   TYPE zcxp\_gc\_tt\_material\_status. *" Durum yapısı* |



Problemin table type olarak tanımladığın bir öğeyi structure gibi kullanmandan kaynaklı.

Table type'ı kullanırken type table of kullanmamıza gerek yok, kullanırsak bir satırı table olmuş bir tablomuz olur

Doğruda lt\_tablename type z\_table\_type yapabiliyoruz

bu arada eğer bir şekilde elinde internal table varsa ls\_structure\_name like line of lt\_internal\_table kullanabilirsin.

### Çözüm Adımları:

* + Öncelikle, **table type** ve **structure** arasında doğru bir ayrım yapmanız gerekiyor.
  + Table type ile çalışırken APPEND ve LOOP işlemlerini kullanabilirsiniz. Ancak table type’ı structure gibi kullanmaya çalıştığınızda bu hatayı alırsınız.

1. **Table Type ve Structure Tanımlamalarını Düzenleme**

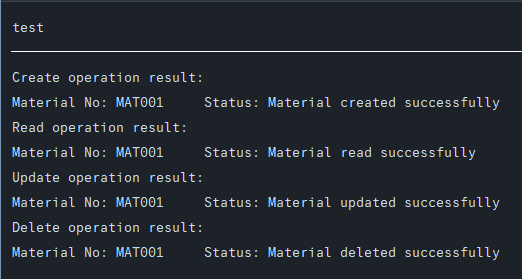
|  |
| --- |
| DATA: lt\_material TYPE zcxp\_gc\_tt\_material, " Table Type  ls\_material LIKE LINE OF lt\_material. " Structure |

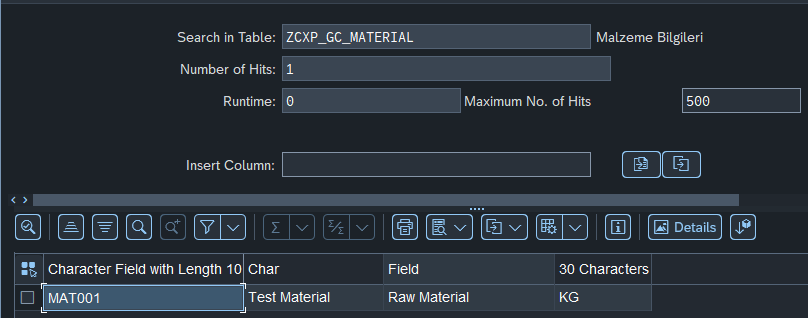
* + lt\_material, bir table type olmalıdır (birden fazla satırı temsil eder).
  + ls\_material, lt\_material'ın her bir satırını temsil eden structure'dır. LIKE LINE OF kullanarak her bir satırı alabilirsiniz.

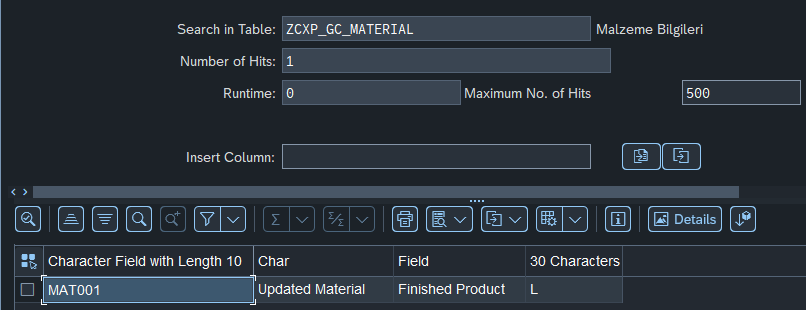
Test programınızda da benzer şekilde table type ve structure arasındaki farkları göz önünde bulundurun. Örneğin, bir tabloyu aşağıdaki gibi tanımlayın:

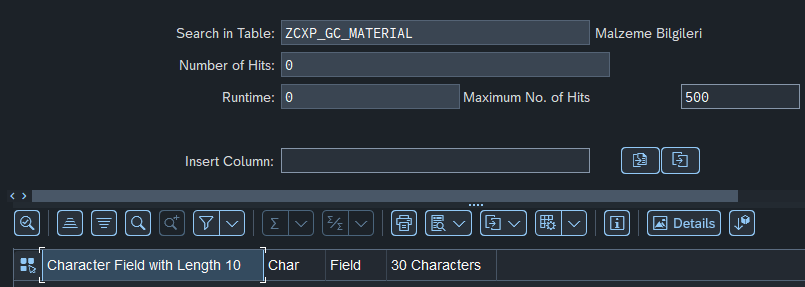
CRUD işlemlerini çağırırken tablo ve structure'ları bu şekilde kullandığınızdan emin olun.

|  |
| --- |
| *\*&---------------------------------------------------------------------\** *\*& Report ZCXP\_GC\_TEST\_MATERIAL* *\*&---------------------------------------------------------------------\** *\*& Test programı malzeme CRUD işlemlerini gerçekleştirecek* *\*&---------------------------------------------------------------------\** REPORT zcxp\_gc\_test\_material.  DATA: lt\_material TYPE zcxp\_gc\_tt\_material,    *" Malzeme tablosu (Table Type)*       ls\_material LIKE LINE OF lt\_material,    *" Malzeme yapısı (Structure)*       lt\_status   TYPE zcxp\_gc\_tt\_material\_status,  *" Durum tablosu (Table Type)*       ls\_status   LIKE LINE OF lt\_status.      *" Durum yapısı (Structure)*  *" Sınıf için nesne yaratma* DATA: material\_operation\_obj TYPE REF TO zcxp\_gc\_cl\_material\_operation.  CREATE OBJECT material\_operation\_obj.  START-OF-SELECTION.  *" ------------------- CREATE İşlemi -------------------* *" Create (C) işlemi için malzeme verisini doldur* ls\_material-material\_no = 'MAT001'. ls\_material-description = 'Test Material'. ls\_material-material\_type = 'Raw Material'. ls\_material-unit\_of\_measure = 'KG'. ls\_material-operation\_type = 'C'.  *" Create işlemi*  *" Malzeme tablosuna (lt\_material) satırı ekle* APPEND ls\_material TO lt\_material.  *" Sınıf metodunu çağırarak CRUD işlemlerini çalıştır* CALL METHOD material\_operation\_obj->execute\_crud   EXPORTING     it\_material\_data = lt\_material   IMPORTING     et\_status        = lt\_status.  *" İşlem sonuçlarını ekrana yazdır* WRITE: / 'Create operation result:'. LOOP AT lt\_status INTO ls\_status.   WRITE: / 'Material No:', ls\_status-material\_no, 'Status:', ls\_status-status. ENDLOOP.  *" LT\_MATERIAL ve LT\_STATUS tablolarını sıfırlama* CLEAR: lt\_material, lt\_status, ls\_material.  *" ------------------- READ İşlemi -------------------* *" Read (R) işlemi için malzeme verisini doldur* ls\_material-material\_no = 'MAT001'. ls\_material-operation\_type = 'R'.  *" Read işlemi*  APPEND ls\_material TO lt\_material.  *" CRUD işlemleri için metodu çağır* CALL METHOD material\_operation\_obj->execute\_crud   EXPORTING     it\_material\_data = lt\_material   IMPORTING     et\_status   = lt\_status.  *" Read işlemi sonuçlarını yazdır* WRITE: / 'Read operation result:'. LOOP AT lt\_status INTO ls\_status.   WRITE: / 'Material No:', ls\_status-material\_no, 'Status:', ls\_status-status. ENDLOOP.  *" LT\_MATERIAL ve LT\_STATUS tablolarını sıfırlama* CLEAR: lt\_material, lt\_status, ls\_material.  *" ------------------- UPDATE İşlemi -------------------* *" Update (U) işlemi için malzeme verisini doldur* ls\_material-material\_no = 'MAT001'. ls\_material-description = 'Updated Material'. ls\_material-material\_type = 'Finished Product'. ls\_material-unit\_of\_measure = 'L'. ls\_material-operation\_type = 'U'.  *" Update işlemi*  APPEND ls\_material TO lt\_material.  *" CRUD işlemleri için metodu çağır* CALL METHOD material\_operation\_obj->execute\_crud   EXPORTING     it\_material\_data = lt\_material   IMPORTING     et\_status   = lt\_status.  *" Update işlemi sonuçlarını yazdır* WRITE: / 'Update operation result:'. LOOP AT lt\_status INTO ls\_status.   WRITE: / 'Material No:', ls\_status-material\_no, 'Status:', ls\_status-status. ENDLOOP.  *" LT\_MATERIAL ve LT\_STATUS tablolarını sıfırlama* CLEAR: lt\_material, lt\_status, ls\_material.  *" ------------------- DELETE İşlemi -------------------* *" Delete (D) işlemi için malzeme verisini doldur* ls\_material-material\_no = 'MAT001'. ls\_material-operation\_type = 'D'.  *" Delete işlemi*  APPEND ls\_material TO lt\_material.  *" CRUD işlemleri için metodu çağır* CALL METHOD material\_operation\_obj->execute\_crud   EXPORTING     it\_material\_data = lt\_material   IMPORTING     et\_status   = lt\_status.  *" Delete işlemi sonuçlarını yazdır* WRITE: / 'Delete operation result:'. LOOP AT lt\_status INTO ls\_status.   WRITE: / 'Material No:', ls\_status-material\_no, 'Status:', ls\_status-status. ENDLOOP. |









Test kısmında metodu çoklu veriyle tek seferde çağırıp dönüşleri değerlendirebilirsen

|  |
| --- |
| *\*&---------------------------------------------------------------------\** *\*& Report ZCXP\_GC\_TEST\_MATERIAL* *\*&---------------------------------------------------------------------\** *\*& Test programı malzeme CRUD işlemlerini gerçekleştirecek* *\*&---------------------------------------------------------------------\** *\*&---------------------------------------------------------------------\** *\*& Report ZCXP\_GC\_TEST\_MATERIAL* *\*&---------------------------------------------------------------------\** *\*& Malzeme CRUD işlemleri - Clean Code* *\*&---------------------------------------------------------------------\** REPORT zcxp\_gc\_test\_material.  DATA: lt\_material TYPE zcxp\_gc\_tt\_material, *" Malzeme tablosu (Table Type)*       ls\_material LIKE LINE OF lt\_material, *" Malzeme yapısı (Structure)*       lt\_status TYPE zcxp\_gc\_tt\_material\_status, *" Durum tablosu (Table Type)*       ls\_status LIKE LINE OF lt\_status. *" Durum yapısı (Structure) " Sınıf için nesne yaratma*  DATA: material\_operation\_obj TYPE REF TO zcxp\_gc\_cl\_material\_operation.  CREATE OBJECT material\_operation\_obj.  START-OF-SELECTION.    *" ------------------- Çoklu Malzeme CRUD İşlemleri -------------------*   PERFORM fill\_material\_data.    *" CRUD işlemlerini tek seferde çalıştır*   CALL METHOD material\_operation\_obj->execute\_crud     EXPORTING       it\_material\_data = lt\_material     IMPORTING       et\_status        = lt\_status.    *" CRUD sonuçlarını değerlendir*   PERFORM display\_results.  *" ------------------- Malzeme Verilerini Doldurma -------------------* FORM fill\_material\_data.    PERFORM add\_material USING 'MAT001' 'Test Material 1' 'Raw Material' 'KG' 'C'.   PERFORM add\_material USING 'MAT002' 'Test Material 2' 'Finished Product' 'L' 'C'.   PERFORM add\_material USING 'MAT001' '' '' '' 'R'.   PERFORM add\_material USING 'MAT002' '' '' '' 'R'.   PERFORM add\_material USING 'MAT001' 'Updated Material 1' 'Finished Product' 'L' 'U'.   PERFORM add\_material USING 'MAT002' 'Updated Material 2' 'Raw Material' 'KG' 'U'.   PERFORM add\_material USING 'MAT001' '' '' '' 'D'.   PERFORM add\_material USING 'MAT002' '' '' '' 'D'.  ENDFORM.  *" ------------------- Malzeme Ekleme Yardımcı Fonksiyonu -------------------* FORM add\_material USING p\_material\_no p\_description p\_type p\_uom p\_operation.    CLEAR ls\_material.   ls\_material-material\_no = p\_material\_no.   ls\_material-description = p\_description.   ls\_material-material\_type = p\_type.   ls\_material-unit\_of\_measure = p\_uom.   ls\_material-operation\_type = p\_operation.    APPEND ls\_material TO lt\_material.  ENDFORM.  *" ------------------- CRUD Sonuçlarını Yazdırma -------------------* FORM display\_results.    LOOP AT lt\_status INTO ls\_status.     CASE ls\_material-operation\_type.       WHEN 'C'. WRITE: / 'Create - Material No:', ls\_status-material\_no, 'Status:', ls\_status-status.       WHEN 'R'. WRITE: / 'Read - Material No:', ls\_status-material\_no, 'Status:', ls\_status-status.       WHEN 'U'. WRITE: / 'Update - Material No:', ls\_status-material\_no, 'Status:', ls\_status-status.       WHEN 'D'. WRITE: / 'Delete - Material No:', ls\_status-material\_no, 'Status:', ls\_status-status.       WHEN OTHERS. WRITE: / 'Unknown Operation - Material No:', ls\_status-material\_no, 'Status:', ls\_status-status.     ENDCASE.   ENDLOOP.  ENDFORM. |

