

Betriebliche Informationssysteme II

ABAP-Praktikum

Wintersemester 2023/2024

Torsten Munkelt

Legen Sie bitte einen Report **ZAB_IW20xxxxx010*** zum Erstellen einer Liste an. Je Zeile sollen in dieser Liste

1. die **Kreditornummer**,
2. der **Name des Kreditors**,
3. der **Ort des Kreditors**,
4. der **Name des Anlegers** und
5. das **Anlagedatum des Kreditors**

enthalten sein. Dazu ist der **Knoten LFA1** der **logischen Datenbank KREDITOR** auszuwerten.

Für die Praktikumsaufgabe sind die Seiten 3 bis 42 des Vorlesungsskripts relevant.

*Setzen Sie für die fünf kleinen „x“ bitte die Ziffern Ihrer Bibliotheksnummer bzw. die Ziffern Ihrer S-Nummer ein.

Programm Bearbeiten Springen System Hilfe

✓ [Dropdown] << [Icons]

Erste Praktikumsaufgabe

[Icon] [Icon]

Auswahl Kreditor


Kreditorenkonto bis [Icon]

Selektion über Suchhilfe

Suchhilfe-Id

Suchstring

[Icon] Suchhilfe

Ausführen (F8) 

Ergebnis Praktikumsaufgabe 1 mit LDBen (II)

Liste Bearbeiten Springen System Hilfe



Erste Praktikumsaufgabe

Erste Praktikumsaufgabe 1

Kreditornummer	Name des Kreditors	Ort des Kreditors	Name des Anlegers	Anlagedatum des Kreditors
101000	Olympic Protective Gear	Atlanta	BOETTCHER	24.05.2016
101001	Olympic Protective Gear	Atlanta	BOETTCHER	24.05.2016
101002	Olympic Protective Gear	Atlanta	BOETTCHER	24.05.2016



Liste Bearbeiten Springen System Hilfe



Erste Praktikumsaufgabe

Erste Praktikumsaufgabe 1

K01157	Sicherheitstechnik Schneider GmbH	Dresden	TEACH-006	06.09.2021
KB00286	Bumpy Bike Hamann GmbH	Dresden	TEACH-005	08.09.2021
KB01157	Bumpy Byke Schneider GmbH	Dresden	TEACH-006	14.09.2021
KM01157	MyLock Schneider GmbH	Dresden	TEACH-006	13.09.2021



Legen Sie bitte einen Report **ZAB_IW20xxxxx015*** zum Erstellen einer Liste an. Je Zeile sollen in dieser Liste

1. die **Kreditornummer**,
2. der **Name des Kreditors**,
3. der **Ort des Kreditors**,
4. der **Name des Anlegers** und
5. das **Anlagedatum des Kreditors**

enthalten sein. Dazu ist die transparente **Tabelle LFA1** mit SQL auszuwerten.

Für die Praktikumsaufgabe sind zusätzlich die Seiten 43 bis 45 des Vorlesungsskripts relevant.

*Setzen Sie für die fünf kleinen „x“ bitte die Ziffern Ihrer Bibliotheksnummer bzw. die Ziffern Ihrer S-Nummer ein.

Ergebnis Praktikumsaufgabe 1 mit SQL

Liste Bearbeiten Springen System Hilfe



Praktikumsaufgabe 1 mit SQL

Praktikumsaufgabe 1 mit SQL 1

Kreditornummer	Name des Kreditors	Ort des Kreditors	Name des Anlegers	Anlagedatum des Kreditors
119500	Thick Spoke	Heilbronn	BOETTCHER	24.05.2016
119501	Thick Spoke	Heilbronn	BOETTCHER	24.05.2016
119502	Thick Spoke	Heilbronn	BOETTCHER	24.05.2016



Liste Bearbeiten Springen System Hilfe



Praktikumsaufgabe 1 mit SQL

Praktikumsaufgabe 1 mit SQL 1

K01020	Markenräder Norddeutschland GmbH	Hamburg	TEACH-005	22.09.2021
K01030	Zweirad Profi Brüder Hansen GmbH	Hamburg	TEACH-005	22.09.2021
KM01157	MyLock Schneider GmbH	Dresden	TEACH-006	13.09.2021
KB01157	Bumpy Byke Schneider GmbH	Dresden	TEACH-006	14.09.2021



Modifizieren Sie die in Aufgabe 1 angelegten Reports so, dass folgende Kreditoren selektiert und in die Liste übernommen werden:

- a) alle Kreditoren aus Dresden,
- b) alle Kreditoren, die nicht aus Dresden kommen,
- c) alle Kreditoren, die im Zeitraum vom 01.03.2023 bis zum 01.08.2023 angelegt worden sind,
- d) alle Kreditoren aus Dresden, die gleichzeitig Debitor sind, und
- e) alle Kreditoren aus Dresden, die nicht gleichzeitig Debitor und im Zeitraum vom 01.03.2023 bis zum 01.08.2023 angelegt worden sind.

- Bitte modifizieren Sie sowohl das Programm mit der logischen Datenbank KDF als auch das SQL-Programm so, dass beide das Feld „Debitorennummer“ aus dem Knoten/der Tabelle LFA1 pro Kreditor zusätzlich ausgeben.
- Bitte selektieren Sie die entsprechenden Kreditoren für a), b), c), d) und e) jeweils erst über den Selektionsbildschirm, der dem Programm mit der logischen Datenbank KDF und dem Knoten LFA1 vorgeschaltet ist.
- Bitte erweitern Sie erst dann das entsprechende SQL-Programm um WHERE-Klauseln für a), b), c), d) und e).
- Bitte verwenden Sie zu Einschränkung der Ergebnismenge erst Konstanten im Quelltext und erst dann PARAMETERS und Selektionstexte.
- Setzen Sie Aufgabe a) mit PARAMETERS und Selektionstexten um.

Legen Sie einen neuen Report an, in dem je Kreditor **in einer Tabellenüberschrift ausgegeben** werden:

1. die Kreditorennummer,
2. der Name des Kreditors und
3. der Name des Anlegers.

In der Tabelle sind **je Tabellenzeile auszugeben**:

4. die Belegnummer der Kreditorenrechnung,
5. das Buchungsdatum,
6. der Buchungsbetrag,
7. die Belegnummer der Gegenbuchung und
8. das Buchungsdatum der Gegenbuchung.

Jede Tabelle endet mit einer **Summenzeile, in welcher ausgegeben werden**:

9. die Summe der Zahlungen und
10. die Summe der offenstehenden Verbindlichkeiten.

Die Tabelle ist für einen und **zunächst nur für einen Kreditor** anzuzeigen. Dann sind die Tabellen für mehrere Kreditoren anzuzeigen. Sollten für einen Kreditor **keine Buchungsbelege** existieren, so ist die **Ausgabe der leeren Tabelle zu unterdrücken**. Dafür soll der **Hinweis ausgegeben werden, dass keine Buchungssätze existieren**. Suchen Sie sich die benötigten Felder für die Listenerstellung aus der **logischen Datenbank KREDITOR** heraus!

K00001	Zampi Zweirad GmbH Dresden	WING227
--------	----------------------------	---------

BelegNr	BuchDt	Betrag	GBelegNr	GBuchDt
---------	--------	--------	----------	---------

100000081	06.03.2016	7.218,00	1500000011	06.03.2016
-----------	------------	----------	------------	------------

100000087	...			
-----------	-----	--	--	--

...				
-----	--	--	--	--

15.552,00	keine Gegenbuchung offen
-----------	--------------------------

1. Der Knoten der **logischen Datenbank**, der die Daten der Kreditorenrechnung, **Felder 4. bis 8.**, enthält, heißt „**Buchhaltung: Sekundärindex für Kreditoren**“.
2. Für besagten Knoten und die **Ausgabe der Rechnungsdaten pro Kreditor** benötigen Sie einen **weiteren GET-Ereignisblock**.
3. Realisieren Sie bitte im selben GET-Ereignisblock mittels **IF ENDIF.**, dass **nur Belege der Belegart „Kreditorenrechnung“** ausgewertet werden.
4. Selektieren Sie bitte in der vorgeschalteten Selektionsmaske Ihren/einen Kreditor **KLxxxxx** oder **Kxxxxx** für die **Anzeige eines einzelnen Kreditors**.
5. Deklarieren Sie mit **LIKE** und verwenden Sie bitte **zwei Variablen** zur Akkumulation der **Summe der Rechnungen** und der **Summe der offenen Verbindlichkeiten** pro Kreditor.
6. Bitte lassen Sie die **Belegüberschriften nur genau beim ersten Beleg** pro Kreditor ausgeben, so dass Sie auch nicht erscheinen, wenn kein Beleg existiert.
7. Realisieren Sie (die Berechnung und) die **Ausgabe der Summenzeile** bitte im **Ereignisblock GET ... LATE**.
8. Bitte **unterdrücken** Sie im selben Ereignisblock die **Ausgabe der Summenzeile**, und geben Sie stattdessen den **Hinweis** „Es existieren keine Buchungssätze.“ aus, wenn die **Anzahl der Buchungssätze 0** beträgt (**IF ELSE. ... ENDIF.**).

Lösen Sie Praktikumsaufgabe 3 ohne logische Datenbank. Bedenken Sie dabei, dass in der Tabelle BSIK im Gegensatz zur logischen Datenbank nur noch die offenen Posten stehen und in der Tabelle BSAK die ausgeglichenen, so dass Sie beide Tabellen einbeziehen müssen.

1. Verwenden Sie die Tabellen LFA1, BSIK und BSAK zur Vereinbarung der Working Areas (Arbeitsbereiche, „Structs“).
2. Deklarieren Sie mit **LIKE** und verwenden Sie bitte Variablen um für den jeweiligen Lieferanten die **Summe aller Rechnungen**, die **Summe der offenen Verbindlichkeiten** und die **Anzahl der Belege** zu bestimmen.
3. Durchlaufen Sie die Tabellen bitte in **geschachtelten SELECT-Anweisungen**: **außen** die Tabelle **LFA1**, **innen** sequentiell (hintereinander, nicht geschachtelt) die Tabellen **BSAK** und **BSIK**.
4. Schränken Sie im äußeren **SELECT** mittels **WHERE** auf **einen oder wenige Lieferanten** ein.
5. Schränken Sie in den inneren **SELECTS** mittels **WHERE** auf die **Belege** ein, die **zum aktuellen Lieferanten** gehören, und auf die Belege, deren **Belegart Kreditorenrechnung** lautet.
6. Bitte lassen Sie die **Belegüberschriften** **nur genau beim ersten Beleg** pro Kreditor ausgeben, so dass Sie auch nicht erscheinen, wenn kein Beleg existiert.
7. Realisieren Sie (die Berechnung und) die **Ausgabe der Summenzeile** bitte hinter den inneren **SELECTS**.
8. Bitte **unterdrücken** Sie dort auch die **Ausgabe der Summenzeile**, und geben Sie stattdessen bitte den **Hinweis** „Es existieren keine Buchungssätze.“ aus, wenn die **Anzahl der Buchungssätze 0** beträgt.

Entwickeln Sie einen Report, der innerhalb eines bestimmten Zeitraumes **sortiert nach Buchungsdatum** von allen Materialbelegen des Einkaufs

1. die Belegnummer,
 2. das Buchungsdatum,
 3. die Materialnummer,
 4. die Menge und
 5. den Sachbearbeiter
- anzeigt.

Am Ende der Liste soll eine **Übersicht** angezeigt werden, **welches Material, in welcher Menge** eingekauft worden ist.

- Verwenden Sie bitte die **logische Datenbank „Lagerbewegungen“** und **nicht** die logische Datenbank „Lagerbewegungen zum Material“.
- **Selektieren** Sie zum Test des Programmes bitte im Selektionsbildschirm nur wenige Materialbelege.
- Legen Sie nötigenfalls bitte **mehrere Materialbelege** mir **mehreren Positionen und mehrmals den gleichen Materialien** an.
- Verwenden Sie zum **Sortieren der Materialbelege** nach Buchungsdatum bitte den **Extrakt(datenbestand)** und **keine** interne Tabelle.

- **Definieren** Sie bitte den **Extrakt**
 - zu Beginn des Programmes mittels dreier **FIELD-GROUPS**, **HEADER**, **KOPF** und **SEGMENT**, und
 - und **fügen Sie dem Extrakt** bitte im **S-O-S** mittels **INSERT ... INTO ...** die entsprechenden **Felder hinzu**.
- **Füllen** Sie den **Extrakt** im Anschluss mittels **EXTRACT** in den **GET-Ereignisblöcken**, wobei der **HEADER automatisch** gefüllt wird.
- **Sortieren** Sie den **Extrakt** bitte im nächsten Schritt im **E-O-S** mittels **SORT BY ... ASCENDING ...**
- **Geben** Sie den **Extrakt** dann bitte im **E-O-S** in einer/m **LOOP aus**, indem Sie bitte in **AT-Ereignisblöcken auf die unterschiedlichen Satzarten (FIELD-GROUPS) reagieren**.

- **Vereinbaren** Sie bitte im E-O-S mit **TYPES** einen **strukturierten Datentyp** für die Aufnahme
 - einer **Materialnummer** und
 - einer **Menge** (des entsprechenden Materials).
- **Deklarieren** Sie bitte im E-O-S im Anschluss
 - einen **Arbeitsbereich vom obigen Typ** und
 - mittels `... TYPE SORTED TABLE OF ... WITH UNIQUE KEY ...` eine **sortierte interne Tabelle** mit **Datensätzen vom obigen Typ**

- **Akkumulieren** Sie bitte die **Mengen pro Material** in obiger Tabelle, indem Sie im **AT-Ereignisblock** für jede **Zeile des Materialbeleges**
 - mittels **READ TABLE ... INTO ... WITH TABLE KEY ...** das Material der Zeile des Materialbeleges aus der Tabelle in den Arbeitsbereich lesen,
 - im Falle des erfolgreichen Lesens (`sy-subrc = 0`)
 - das **Mengenfeld** im Arbeitsbereich **um die Menge** der Zeile des Materialbelegs erhöhen und
 - mittels **MODIFY TABLE ... FROM ...** die Tabellenzeile mit dem Arbeitsbereich aktualisieren und
 - sonst
 - im Arbeitsbereich die **Materialnummer** und die **Menge** aufgrund der aktuellen Zeile des Materialbeleges **setzen** und
 - mittels **INSERT ... INTO TABLE ...** den Inhalt des Arbeitsbereiches in die Tabelle einfügen.

- **Durchlaufen** Sie bitte nach der/dem `LOOP` über dem Extrakt die interne Tabelle mit `LOOP AT . . . INTO . . .`, und geben Sie bitte die **Materialnummern und Mengen** aus.

Ergebnis Praktikumsaufgabe 5

SAP

Materialbelege

K: 49000000000 23.06.2010 WA WEIDNER

S: 0001 561 DXTR1000	100,000
S: 0002 561 DXTR1001	100,000
S: 0003 561 DXTR1002	100,000
S: 0004 561 DXTR1003	100,000
S: 0005 561 DXTR1004	100,000
S: 0006 561 DXTR1005	100,000
S: 0007 561 DXTR1006	100,000
S: 0008 561 DXTR1007	100,000
S: 0009 561 DXTR1008	100,000
S: 0010 561 DXTR1009	100,000
S: 0011 561 DXTR1010	100,000
S: 0012 561 DXTR1011	100,000
S: 0013 561 DXTR1012	100,000
S: 0014 561 DXTR1013	100,000
S: 0015 561 DXTR1014	100,000
S: 0016 561 DXTR1015	100,000
S: 0017 561 DXTR1016	100,000
S: 0018 561 DXTR1017	100,000
S: 0019 561 DXTR1018	100,000
S: 0020 561 DXTR1019	100,000
K: 49000000001 23.06.2010 WA WEIDNER	
S: 0001 561 DXTR1020	100,000

SAP R41 (3) 201 r41z INS

SAP

S: 0020 561 DXTR1039 100,000

Bewegte Materialmenge

Materialnummer	Gesamtmenge
DXTR1000	100,000
DXTR1001	100,000
DXTR1002	100,000
DXTR1003	100,000
DXTR1004	100,000
DXTR1005	100,000
DXTR1006	100,000
DXTR1007	100,000
DXTR1008	100,000
DXTR1009	100,000
DXTR1010	100,000
DXTR1011	100,000
DXTR1012	100,000
DXTR1013	100,000
DXTR1014	100,000
DXTR1015	100,000
DXTR1016	100,000
DXTR1017	100,000

SAP R41 (3) 201 r41z INS

Erzeugen Sie bitte eine Liste wie in Praktikumsaufgabe 2 b).

- a) Die Liste soll nach dem Erstellungsdatum fallend sortiert sein.
Hinweis: Lösen Sie die Aufgabe bitte über eine interne Tabelle.
- b) Die Liste soll wahlweise fallend bzw. steigend sortierbar sein.
Hinweis: Die Art der Sortierung soll über eine Optionsgruppe auf dem Selektionsbildschirm steuerbar sein!

- Verwenden Sie bitte die logische Datenbank „DATENBANK KREDITOR“.
- Verwenden Sie aus der logischen Datenbank bitte den obersten Knoten „Lieferantenstamm (allgemeiner Teil)“.
- Verwenden Sie aus dem Knoten bitte die Felder:
 - Kontonummer des Lieferanten bzw. Kreditors
 - Name 1
 - Name 2
 - Ort
 - Name des Sachbearbeiters, der das Objekt hinzugefügt hat
 - Datum, an dem der Satz hinzugefügt wurde
- Vereinbaren Sie bitte einen Typ `KREDITOR_TYPE`, der Felder enthält, die den obigen Feldern entsprechen.
- Legen Sie bitte einen Arbeitsbereich `KREDITOR` und eine Tabelle `KREDITOR_TABLE` jeweils des obigen Typs an.

- Füllen Sie bitte im `GET` für den Kreditor die Felder des Arbeitsbereiches `KREDITOR`.
- Fügen Sie bitte im selben `GET` danach den Arbeitsbereich `KREDITOR` in die Tabelle `KREDITOR_TABLE` ein.
- Die folgenden Schritte sind bitte im `E-O-S` durchzuführen.
- Geben Sie bitte eine Überschrift für die auszugebenden Felder aus.
- Sortieren Sie die Tabelle `KREDITOR_TABLE` bitte absteigend nach dem „Datum, an dem der Satz hinzugefügt wurde“ und aufsteigend nach der „Kontonummer des Lieferanten bzw. Kreditors“.
- Durchlaufen Sie bitte die Tabelle `KREDITOR_TABLE`, und schreiben Sie bitte in jedem Durchlauf den jeweiligen Datensatz in den Arbeitsbereich `KREDITOR` (Achtung: alles in einer Anweisung).
- Geben Sie die Felder des Arbeitsbereiches `KREDITOR` bitte im Schleifenrumpf aus.

- Schränken Sie beim Test im Selektionsbildschirm bitte auf wenige Lieferanten ein, z. B. auf K*.
- Die nachfolgenden Schritte beziehen sich auf die Lösung der Teilaufgabe b).
- Bitte legen Sie nach der Variablendeklaration den folgenden `SELECTION-SCREEN` an:

```
SELECTION-SCREEN BEGIN OF BLOCK B  
    WITH FRAME TITLE TEXT-S01.
```

```
PARAMETERS:
```

```
    DOWN RADIOBUTTON GROUP G1 DEFAULT 'X',  
    UP RADIOBUTTON GROUP G1.
```

```
SELECTION-SCREEN END OF BLOCK B.
```

- Dieser `SELECTION-SCREEN` erscheint zusätzlich in der Benutzeroberfläche dient der Entgegennahme der Benutzereingabe, ob ab(- oder auf)steigend sortiert werden soll.

- Bitte aktivieren Sie das Programm, bevor Sie das Textsymbol und die Selektionstexte ergänzen.
- Ergänzen Sie bitte das Textsymbol `S01` für den Titel des Rahmens im Selektionsbildschirm und die Selektionstexte `DOWN` und `UP` für die Beschriftung der Radiobuttons.
- Bitte deklarieren Sie im Deklarationsteil des Programmes eine ganzzahlige Variable `ABSTEIGEND`, welche die Benutzereingabe speichert, ob ab(- oder auf)steigend sortiert werden soll.
- Bitte setzen Sie im Ereignis((behandlungs)block) `AT SELECTION-SCREEN ON RADIOBUTTON GROUP G1.` die Variable `ABSTEIGEND` in Abhängigkeit davon, ob der Radiobutton `UP` oder der Radiobutton `DOWN` ausgewählt worden ist.
- Sortieren Sie die Tabelle bitte vor der Schleife zu ihrer Ausgabe aufsteigend oder absteigend nach Datum in Abhängigkeit davon, wie die Variable `ABSTEIGEND` gesetzt worden ist.

- Übersetzen Sie bitte die deutschen Texte für das Textsymbol S01 und für die Selektionstexte UP and DOWN in die englische Sprache.
- Tragen Sie bitte die Übersetzungen für das Textsymbol und für die Selektionstexte ins SAP-System ein.
- Aktivieren Sie bitte das Textsymbol und die Selektionstexte erneut.
- Melden Sie sich bitte vom SAP-System ab und in englischer Sprache (EN) wieder an.
- Bitte nehmen Sie nötigenfalls Änderungen vor, bis die Texte für das Textsymbol und die Selektionstexte bei der Ausführung des Programmes in englischer Sprache erscheinen.

Ergebnis Praktikumsaufgabe 6

Programme Bearbeiten Springen System Hilfe

Sortierte Kreditoren

Auswahl Kreditor

Kreditorenkonto bis

Selektion über Suchhilfe

Suchhilfe-Id

Suchstring

Sortierung

☒ fallend nach Datum

☐ steigend nach Datum

Ausführen (F8) R41 (3) 20:

Liste Bearbeiten Springen System Hilfe

Sortierte Kreditoren

Sortierte Kreditoren 1

Kreditorenliste

Nummer	Name1	Name2
KL00286_14	Fa. Zampi Zweirad Hamann	GmbH Dresden
KL00111	Zampi Zweirad	Buschendorf Gmb
KL00655	Zampi Zweirad Buschendorf	GmbH Dresden
KL00007	Fa. Zampi Zweirad Rüger2	GmbH Dresden
KL00286	Fa. Zampi Zweirad name GmbH Dresden	
KL00862	Fa. Zampi Zweirad REichelt GmbH Dre	
KL00643	Zampi Zweirad Buschendorf GmbH	

R41 (3) 201 r41z INS

The screenshot shows the SAP Laboratory 6 interface. At the top, there is a menu bar with 'Program', 'Edit', 'Goto', 'System', and 'Help'. Below the menu bar is a toolbar with various icons. The main content area is titled 'Laboratory 6' and contains several sections:

- Vendor selection**: This section includes a 'Vendor account' field with a yellow highlight and a magnifying glass icon, followed by a 'to' field and a button with a folder icon.
- Selection using search help**: This section includes a 'Search help ID' field, a 'Search string' field, and a 'Search help' button with a folder icon.
- Sort**: This section is highlighted with an orange rectangle and contains two radio buttons: 'Descending according to date' (selected) and 'Ascending according to date'.

The SAP logo is visible at the bottom center of the interface.

Lösen Sie die folgende Aufgabe bitte über Verzweigungslisten. In der Grundliste sollen die Stammdaten des Kreditors angezeigt werden. Über Zeilenauswahl wird die zugehörige Verzweigungsliste der Kreditorenrechnungsbelege (vergleiche Praktikumsaufgabe 4) angezeigt.

- Lösen Sie die Aufgabe bitte **nicht** mit logischen Datenbanken, sondern mittels `SELECT (SQL)`.
- Verwenden Sie bitte die Tabellen
 - „Lieferantenstamm (allgemeiner Teil)“,
 - „Buchhaltung: Sekundärindex für Kreditoren“ und
 - „Buchhaltung: Sekundärindex für Kreditoren (ausgegl. Posten“.
- Legen Sie bitte die Arbeitsbereiche `WA_LFA1` für die erste und `WA_BSXX` für die zweite und die dritte Tabelle an.
- Lassen Sie sich bitte mittels `SELECT` eine Liste der Kreditoren anzeigen (wie in der SQL-Version der Praktikumsaufgabe 1).
- Was passiert, wenn Sie auf eine Zeile im Report klicken?
- Fügen Sie bitte folgenden Ereignisbehandlungsblock ein
`AT LINE-SELECTION.`
`WRITE: 'Verzweigungslistennummer: ', sy-lsind.`
, und testen Sie das Programm bitte mit Doppelklick auf Reportzeile.

- „Merken“ Sie sich bitte nach der Ausgabe eines Lieferanten am Ende (innerhalb) der `SELECT`-, „Schleife“ die Lieferantennummer zur Zeilennummer im Report: `HIDE WA_LFA1-LIFNR.`
- Ergänzen Sie am Ende von `AT LINE-SELECTION` bitte:
`WRITE: / 'Nummer der "doppelgeklickten" Zeile: ',`
`sy-lilli.`
`WRITE: / 'Für diese Zeile gemerkte`
`Lieferantennummer: ', WA_LFA1-LIFNR.`
- Testen Sie bitte, ob in der 1. Verzweigungsliste Zeile und Nummer des Lieferanten angezeigt werden, dessen Rechnungsbelege später angezeigt werden sollen.

- Ersetzen Sie bitte den Ereignisbehandlungsblock AT LINE-SELECTION durch folgenden Quelltext:
AT LINE-SELECTION. "Globaler Ereignisblock
CASE sy-lsind.
 "Unterscheidung nach Listenindex, ...
 WHEN 1. "... was angezeigt werden soll:
 PERFORM KreditorenRechnungen
 USING WA_LFA1-LIFNR.
ENDCASE.
- Fügen Sie bitte am Ende des Programmes ein Unterprogramm zum Anzeigen der Kreditorenrechnungen ein, und testen Sie bitte:
FORM KreditorenRechnungen USING MY_LIFNR.
 WRITE: /1 'Rechnungen von Kreditör ', MY_LIFNR.
 WRITE: /1 'BelegNr', 12 'BuchDt', 23 'Betrag', 39
 'Belegart'.
 "Kreditorenrechnungen später hier anzeigen lassen
ENDFORM.

- Selektieren Sie bitte im Unterprogramm `KreditorenRechnungen` sowohl die offenen als auch die ausgeglichenen Posten (hintereinander) zum aktuellen Lieferanten, und lassen Sie sich die Posten bitte anzeigen.
- Orientieren Sie sich dabei bitte an der Lösung zu Praktikumsaufgabe 4.
- Testen Sie das Programm bitte mit {Doppelklick|F2|Kontextmenü „Auswählen“} auf Lieferanten, zu denen Posten existieren.

Ergebnis Praktikumsaufgabe 7 (I)

SAP

Nummer	Name 1	Name 2	Anleger
K00001	Fa. Sicherheitstechnik Ziemann GmbH Dresden		BW1370991
K00002	Fa Sicherheitstechnik Müller Dresde		BW1370911
K00007	Fa. Sicherheitstecnik Rügerl	GmbH Dresden	RUEGER
K00111	Sicherheitstechnik Buschendorf	GmbH Dresden	SBUSCHEN
K00286	Fa. Sicherheitstechnik Hamann	GmbH Dresden	HAMANNA

K00286 Hilfe F1
Auswählen F2
Zurück F3
Einnahmelmöglichkeiten F4

SAP

Rechnungen von Kreditor K00286

BelegNr	BuchDt	Betrag	Belegart
1900000006	07.10.2014	7.218,00	KR
1900000478	20.10.2015	7.218,00	KR
1900000538	22.10.2015	1.190,00	KR

SAP R41 (5) 201 r41z INS

Bitte ergänzen Sie die Lösung der Praktikumsaufgabe 7 durch eine zweite Verzweigungsliste, welche die Daten der Zahlungsbelege zur Kreditorenrechnung enthält, die in der ersten Verzweigungsliste ausgewählt worden ist.

- Bitte erweitern Sie Ihre Lösung zur Praktikumsaufgabe 7 zur Lösung der Praktikumsaufgabe 8.
- Bitte ergänzen Sie Ihr Programm im Hauptprogramm, im Unterprogramm und im `WHEN` an der passenden Stellen um
 - die Befehle `CLEAR` und
 - den Test `CHECK NOT . . . IS INITIAL.`
- ..., so dass bei der Auswahl einer Zeile, zu der keine Variablenwerte im `HIDE`-Bereich hinterlegt worden sind, keine Reaktion erfolgt.
- Testen Sie bitte ausführlich, dass keine Reaktion erfolgt - auch nach Rücksprung in die Kreditorenliste und erneuter Auswahl einer Zeile ohne im `HIDE`-Bereich hinterlegten Kreditor.
- „Verstecken“ Sie bitte im Unterprogramm `KreditorenRechnungen` die Lieferantenummer, die Belegnummer und die Belegnummer des Ausgleichbeleges (aus dem Arbeitsbereich `BSXK`) im `HIDE`-Bereich zur Nutzung in der zweiten Verzweigungsliste.


- Legen Sie bitte ein weiteres Unterprogramm `AusgleichBelege` an, dem als Parameter die Lieferantenummer, die Belegnummer und die Belegnummer des Ausgleichbelegs als Werteparameter übergeben werden sollen.
- Legen Sie im `AT LINE SELECTION` bitte mit `WHEN` einen weiteren Fall für die zweite Verzweigungsliste an, in dem das Unterprogramm `AusgleichBelege` mit den entsprechenden Parametern aufgerufen wird.
- Ergänzen Sie das Unterprogramm bitte um die Ausgabe der übergebenen Parameter, und testen Sie bitte, ob die Parameterübergabe funktioniert.
- Durchlaufen Sie im Unterprogramm bitte wiederum nacheinander die Tabellen `BSIK` und `BSAK`, und selektieren Sie bitte jeweils nach der übergebenen Lieferantenummer **und** bei der Belegnummer nach der übergeben Belegnummer des Ausgleichbelegs.

- Testen Sie das Programm bitte mit Kreditoren, für die offene und ausgeglichene Posten existieren.
- Fügen Sie bitte – sofern nicht bereits geschehen – in die erste Verzweigungsliste mindestens eine Überschriftenzeile **und** eine die Liste abschließende Zeile ein, für die jeweils keine Variablenwerte im HIDE-Bereich versteckt werden.
- Bitte probieren Sie aus, was geschieht, wenn Sie in der ersten Verzweigungsliste eine Überschriftenzeile und eine die Liste abschließende Zeile auswählen, für die jeweils keine Variablenwerte im HIDE-Bereich versteckt worden sind – auch nach Rücksprung in die erste Verzweigungsliste und erneuter Auswahl einer Überschriftenzeile oder einer abschließenden Zeile.
- Bitte ergänzen Sie Ihr Programm in beiden Unterprogrammen und im WHEN an den richtigen Stellen um
 - die Befehle `CLEAR` und
 - den Test `CHECK NOT . . . IS INITIAL.`
- Bitte testen Sie das Programm danach erneut ausführlich.


- Bitte testen Sie die fachliche Korrektheit Ihres Programmes anhand der „Kreditoren Einzelposten Liste“, Transaktion S_ALR_87012103, im SAP-System (siehe Praktikum 5), und nehmen Sie bitte ggf. Korrekturen vor.
- Ermöglichen Sie bitte mittels HOTSPOT, dass eine „Hand“ über den Lieferantenzeilen in der Grundliste und über den Belegzeilen in der ersten Verzweigungsliste angezeigt wird, wenn man die Maus über die jeweilige Zeile hält, und beim Klicken auf die jeweilige Lieferanten- und Belegzeile ein einfacher Klick genügt, um in die nächste Listenstufe zu verzweigen.

Ergebnis Praktikumsaufgabe 8


Liste Bearbeiten Springen System Hilfe



Nummer	Name 1	Name 2
K00286	[Fa. Sicherheitstechnik Hamann] GmbH Dr
KL00286	Fa. Zampi Zweirad na	GmbH Dresden
Fertig. :-)		




Liste Bearbeiten Springen System Hilfe




Rechnungen von Kreditor K00286

BelegNr	BuchDt	Betrag	Belegart	GBelegNr	GBuchDat
1900000006	07.10.2014	7.218,00	KR	[1500000004]	07.10.2014
1900000478	20.10.2015	7.218,00	KR	1500000527	20.10.2015
1900000538	22.10.2015	1.190,00	KR	1500000581	22.10.2015

Zeileninhalt:




Liste Bearbeiten Springen System Hilfe



Ausgleichbelege von Kreditor K00286 zu Beleg 1900000006 :

BelegNr	BuchDt	Betrag	Belegart	GBelegNr	GBuchD
1500000004	07.10.2014	7.218,00	KZ	1500000004	07.10.

Zeileninhalt:



R41 (5) 201 r41z INS

Entwickeln Sie bitte einen interaktiven Report, mit dem man über ein Menü andere Reports auswählen und abarbeiten kann. Legen Sie zur Bedienerführung bitte eigene Befehlsschaltflächen an.

- Die für die Praktikumsaufgabe wichtigste Tabelle ist die Tabelle REPOSRC („Report Sourcen“). Sie enthält sämtliche Reports.
- Legen Sie bitte, um die Tabelle auszulesen, einen Arbeitsbereich an, welcher Teile der Datensätze der Tabelle aufnehmen kann:

DATA:

```
BEGIN OF WA_REPOSRC,  
  PROGRAMME TYPE C LENGTH 40,  
  R3STATE TYPE C LENGTH 1,  
END OF WA_REPOSRC.
```

- Durchlaufen Sie bitte Tabelle REPOSRC mittels SELECT, und schränken Sie dabei bitte unbedingt auf Ihre Programme (PROGRAMME LIKE 'ZAB_IWyyxxxxx%') und aktive Programme (R3STATE = 'A') ein, weil inaktive Programme nicht von extern ausgeführt werden können.
- (yy bitte durch Ihr Immatrikulationsjahr und xxxxx bitte durch Ihre Bibliotheksnummer ersetzen.)
- Geben Sie im SELECT bitte den Namen und den Status des jeweiligen Programmes aus.

- Selektieren Sie bitte **innerhalb** der bestehenden `SELECT`–Schleife mittels `SELECT SINGLE . . .` einzig den `R3STATE` der eventuell vorhandenen inaktiven Version (`R3STATE = 'I'`) des aktuellen Programmes (`PROGNAME = WA_REPOSRC-PROGNAME`) in eine zuvor zu deklarierende Variable `HELP_STATE` des passenden Datentyps.
- Sofern eine inaktive Version des Programmes vorliegt, ist das Programm inaktiv, auch wenn eine veraltete aktive Version existiert.
- Wenn eine inaktive Version des Programmes vorliegt (das Programm nicht aktiv ist), soll es nicht ausgegeben, sondern mit dem nächsten Programm fortgesetzt werden.
- Wenn das `SELECT SINGLE . . .` keinen Datensatz selektiert, enthält die Variable `HELP_STATE` den Wert, den sie vorab innehatte und der **vorab mittels `CLEAR` initialisiert** werden muss.
- Brechen Sie bitte mittels `CHECK HELP_STATE IS INITIAL.` den aktuellen Schleifendurchlauf ab, falls `HELP_STATE` nicht den initialen Wert, sondern ein `'I'` für eine inaktive Programmversion aufweist.

- Bitte (de)aktivieren Sie einige Ihrer Programme – über den ABAP Editor (SE38) oder über den Object Navigator (SE80) –, falls keines Ihrer Programme (in)aktiv sein werden sollte.
- Die Tabelle `TRDIRT` („Titeltexte zu Programmen“) enthält die sprachabhängigen Bezeichnungen zum jeweiligen Programm.
- Selektieren Sie bitte wiederum **innerhalb** der bestehenden `SELECT`–Schleife mittels eines weiteren `SELECT SINGLE . . .` den `TitelTEXT` des aktuellen Programmes in deutscher Sprache (`SPRSL = 'D'`) aus der Tabelle `TRDIRT`.
- Geben Sie den selektierten Titeltext des aktuellen Programmes bitte zusätzlich zum Programmnamen und dem Programmstatus mit im anschließenden `WRITE: . . .` aus.
- Verstecken Sie bitte nach dem `WRITE: . . .` den Programmnamen `WA_REPOSRC-PROGNAME` oder den gesamten Arbeitsbereich `WA_REPOSRC` mittels `HIDE`, damit er bei der Auswahl einer Listenzeile zur Verfügung steht.

- Legen Sie bitte den Ereignisblock `AT LINE-SELECTION.` an passender Stelle im Programm an, nehmen Sie in ihm bitte eine Fallunterscheidung anhand der Listenstufe `sy-lsind` vor, so dass nur eine erste Verzweigungsliste aufgerufen werden kann.
- Rufen Sie im Fall für die erste Listenstufe das Unterprogramm `ProgrammAusfuehren` mit dem Werteparameter `WA_REPOSRC-PROGNAME.`
- Definieren Sie bitte das Unterprogramm `ProgrammAusfuehren` mit dem entsprechenden Werteparameter und geben Sie ihm bitte folgenden Inhalt: `SUBMIT (MY_PROGNAME) AND RETURN.`
- Testen Sie bitte ihr Programm, das Ihnen eine Liste Ihrer aktiven Programme anzeigen, bei Auswahl einer Listenzeile eines Programmes, z. B. mittels Doppelklick, das entsprechende Programm ausführen und im Anschluss in die Liste Ihrer aktiven Programme zurückkehren soll.
- Versehen Sie die Liste der Programme bitte mit Kopf- und Fußzeile, und sichern Sie es bitte mit `CLEAR` und `CHECK` gegen versehentliche Auswahl der Kopf- und Fußzeile ab.

- Setzen Sie bitte direkt hinter dem Deklarationsteil den GUI-Titel mittels: `SET TITLEBAR 'TITEL100' WITH ':-) ' .`
- Klicken Sie bitte auf `'TITEL100'` doppelt („Vorwärtsnavigation“), und legen Sie den Titel mit Titeltext „Report über &1 Reports“ bitte an.
- Aktivieren Sie bitte den GUI-Titel bzw. die GUI des Programmes im Object Navigator (SE80) und Ihr Programm im ABAP Editor (SE38).
- Testen Sie bitte, ob der GUI-Titel ordnungsgemäß mit dem :-)
angezeigt wird.
- Fügen Sie bitte folgende Zeile nach dem Setzen des GUI-Titels ein:
`SET PF-STATUS 'STATUS100' .`
- Klicken Sie bitte auf `'STATUS100'` doppelt („Vorwärtsnavigation“), legen Sie den Status bitte mit dem Kurztext STATUS100 an, speichern Sie ihn bitte, und kehren Sie bitte zu Ihrem Programm zurück.

- Öffnen Sie den Programmstatus STATUS100 für Ihr Programm bitte im Menu Painter („SAP Menü → Werkzeuge → ABAP Workbench → Entwicklung → Oberfläche → Menu Painter“ oder SE41).
- Erweitern Sie bitte die „Menüleiste“, die „Druckastenleiste“ und die „Funktionstasten“ jeweils durch einen Klick auf „+“, und sehen Sie sich die größtenteils leeren Menübestandteile bitte an.
- Gleichen Sie den aktuellen STATUS100 bitte mit dem Vorlagestatus ab, indem Sie im Menü „Zusätze → Vorlage abgleichen“ und im sich öffnenden Dialog den Radiobutton „Liststatus“ auswählen.
- Überprüfen Sie bitte das Ergebnis des Abgleichs mit dem Vorlagestatus, und erweitern Sie dazu bitte nötigenfalls wieder die Menübestandteile mit „+“.

- Legen Sie bitte in der Menüleiste ein Hauptmenü „Extras“ mit zwei Untermenüs mit den Funktionscodes (FCs) „AUSFUEHREN“ und „DETAILS“ und mit den Oberflächentexten „Ausführen“ und „Details ...“ an.
- Ergänzen Sie in der Drucktastenleiste bitte die Positionen 6 und 7 um die bereits vorhandenen FCs „AUSFUEHREN“ und „DETAILS“.
- Ergänzen Sie bitte die frei belegbaren Funktionstasten F5 und F6 um die bereits vorhandenen FCs „AUSFUEHREN“ und „DETAILS“.
- Speichern und aktivieren Sie bitte den Programmstatus.
- Fügen Sie bitte direkt hinter dem Ereignisblock `AT LINE-SELECTION.` den nachstehenden Ereignisblock `AT USER-COMMAND.` ein, in dem Sie auf die Wahl von Menüeinträgen, das Drücken von Drucktasten (Knöpfen, Schaltflächen) und Funktionstasten reagieren.

- Ereignisblock zur Reaktion auf Benutzerbefehle:

```
AT USER-COMMAND.  
  CASE sy-ucomm.  
    WHEN 'AUSFUEHREN'.  
      PERFORM ProgrammAusfuehren  
        USING WA_REPOSRC-PROGNAME.  
    WHEN 'DETAILS'.  
      PERFORM ProgrammDetailsAnzeigen  
        USING WA_REPOSRC-PROGNAME.  
    WHEN OTHERS.  
      MESSAGE 'Unbehandeltes sy-ucomm!' TYPE 'I'.  
  ENDCASE.
```

- Ergänzen Sie diesen Ereignisblock bitte um CHECKS, dass auch tatsächlich eine Zeile mit einem Programmnamen ausgewählt worden ist, und CLEARen Sie die entsprechenden Variablen bitte an den entsprechenden Stellen, sofern nicht bereits geschehen.

- Ergänzen Sie Ihr Programm um folgendes Unterprogramm, um sich die Details zum jeweils ausgewählten Programm anzeigen zu lassen:

```
FORM PROGRAMMDETAILSANZEIGEN USING P_PROGNAME.
  DATA WA_TRDIR LIKE TRDIR.
  WRITE: 71 'Details von Programm ', P_PROGNAME.
  SELECT * FROM TRDIR INTO CORRESPONDING FIELDS OF WA_TRDIR WHERE NAME = P_PROGNAME.
  WRITE:
    / 'Anwendung: ', WA_TRDIR-APPL,
    / 'Erstellungsdatum: ', WA_TRDIR-CDAT,
    / 'Programmklasse: ', WA_TRDIR-CLAS,
    / 'Autor: ', WA_TRDIR-CNÄM,
    / 'Datenbank aus der Anwendung: ', WA_TRDIR-DBAPL,
    / 'log. Datenbank: ', WA_TRDIR-DBNA,
    / 'Editorsperrvermerk: ', WA_TRDIR-EDTX,
    / 'Festpunktarithmetik: ', WA_TRDIR-FIXPT,
    / 'Generierung eigenes Selektionsbild: Datum: ', WA_TRDIR-IDATE,
    / 'Generierung eigenes Selektionsbild: Uhrzeit: ', WA_TRDIR-ITIME,
    / 'Name der LDB: ', WA_TRDIR-LDBNAME,
    / 'Stand: ', WA_TRDIR-LEVL,
    / 'ABAP-Programmname: ', WA_TRDIR-NAME,
    / 'Maschinell generiertes Programm: ', WA_TRDIR-OCCURS,
    / 'Mastersprache: ', WA_TRDIR-RLoad,
    / 'Mandant: ', WA_TRDIR-RMAND,
    / 'Status: ', WA_TRDIR-RSTAT,
    / 'Generierung Standardselektionsbild: Datum: ', WA_TRDIR-SDATE,
    / 'Berechtigungsgruppe: ', WA_TRDIR-SECU,
    / 'Quelltextschutz: ', WA_TRDIR-SQLX,
    / 'Start nur ueber Variante: ', WA_TRDIR-SSET,
    / 'Generierung Standardselektionsbild: Uhrzeit: ', WA_TRDIR-STIME,
    / 'Programmtyp: ', WA_TRDIR-SUBC,
    / 'Selektionsbildversion: ', WA_TRDIR-TYPE,
    / 'Flag, ob Unicodeprüfungen durchgeführt werden: ', WA_TRDIR-UCCHECK,
    / 'Änderungsdatum: ', WA_TRDIR-UDAT,
    / 'Name des letzten Änderers: ', WA_TRDIR-UNAM,
    / 'Groß-/Kleinschreibung: ', WA_TRDIR-VARCL,
    / 'Versionsnummer: ', WA_TRDIR-VERN.
ENDSELECT.
ENDFORM.
```

- Speichern und aktivieren Sie bitte all Ihre Programmkomponenten.

- Testen Sie das Ausführen eines in der Liste markierten Programmes bitte via:
 - Doppelklick auf Programmzeile,
 - Funktionstaste F5,
 - Menü „Extras ➔ Ausführen“,
 - Schaltfläche „Ausführen“ und
 - Kontextmenü „Ausführen“.
- Testen Sie die Detailanzeige zu einem in der Liste markierten Programm bitte via:
 - Funktionstaste F6,
 - Menü „Extras ➔ Details ...“,
 - Schaltfläche „Details ...“ und
 - Kontextmenü „Details ...“.
- Bitte testen Sie auch die Auswahl von Kopf- und Fußzeilen.

Report über :-> Reports

Ausführen Details ...

Programme über Programmen

Reportname	Status	Beschreibung
ZZE_IW1401014070	A	Sortierte Kr
ZZE_IW1401014083	A	Aufgabe 8 au

Fußzeile zum T

Hilfe F1
Auswählen F2
Zurück F3
Eingabemöglichkeiten F4
Ausführen F5
Details ... F6
Abbrechen F12
Beenden Umsch+F3

Details von Programm ZZE_IW1401014083

Ausführen Details ...

Anwendung: *
Erstellungsdatum: 14.11.2016
Programmklass:
Autor: DEVZE01

Aufgabe 8 ausprobieren

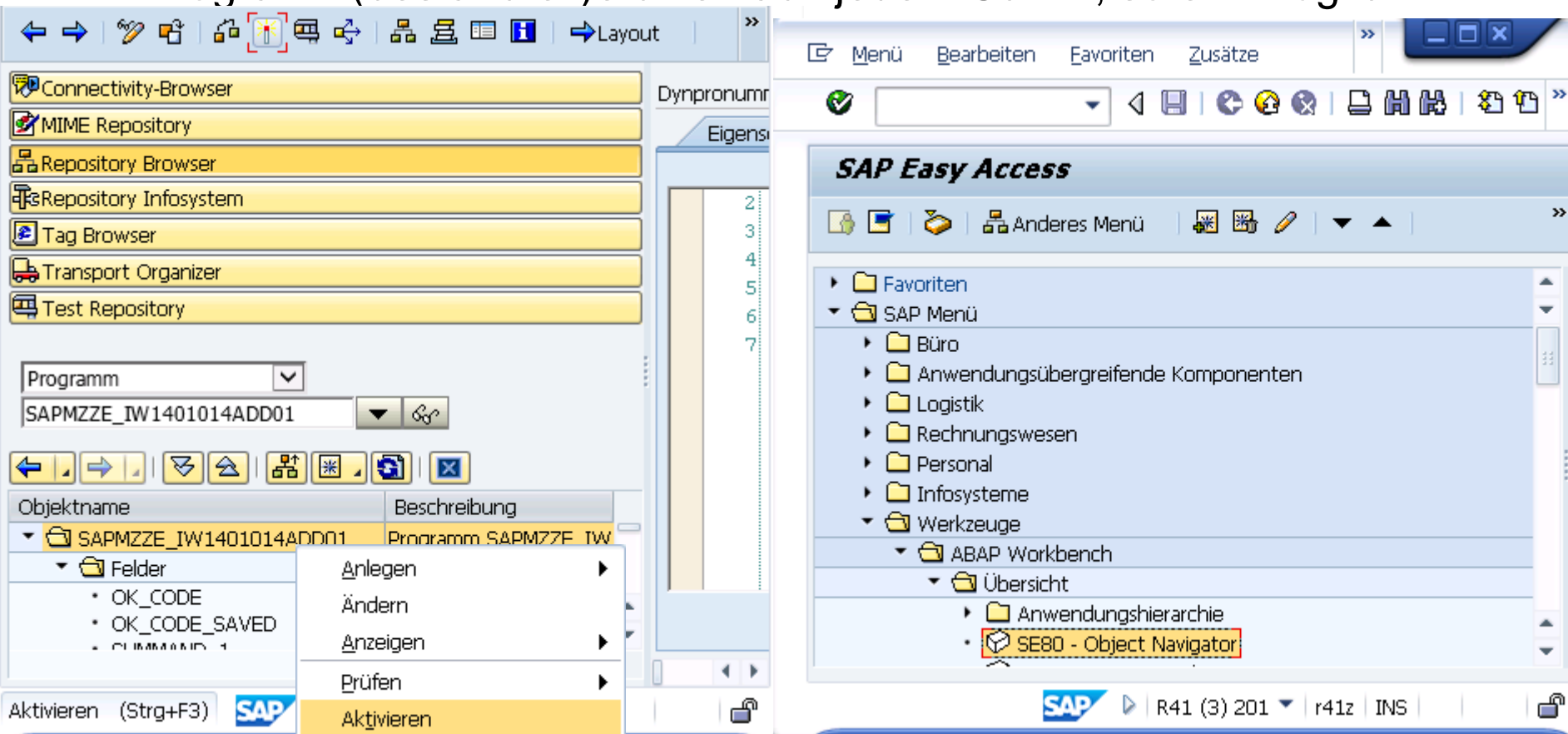
Aufgabe 8 ausprobieren 1

Nummer	Name 1	Name 2
K00286	Fa. Sicherheitstechnik Hamann	GmbH Dresden
KL00286	Fa. Zampi Zweirad name	GmbH Dresden

Fertig. :-)

Entwickeln Sie bitte eine Transaktion mit eigenen Dynpros und Buttons (zur Bedienerführung) für die Ermittlung des durchschnittlichen Benzinverbrauchs pro 100 km. Einzugeben sind die gefahrenen Kilometer und der Benzinverbrauch. Das Programm ermittelt anschließend den Durchschnittsverbrauch und zeigt ihn an. Der Dialog soll beliebig oft wiederholbar sein, bis das Programm über einen Button beendet wird. Legen Sie bitte auch eine Transaktion für das Programm an.

- Bei der Programmierung bewegen Sie sich **ausschließlich** im **Object Navigator** (SE80).
- Speichern **und** aktivieren Sie die angelegten oder veränderten Programm(bestandteil)e bitte nach jedem Schritt, sofern möglich.



- Modulpool SAPMZAB_IW20xxxxx100 vom Typ „Programm“ anlegen
- mit TOP-Include (SAPMZAB_IW20xxxxx100TOP)
- Typ: Modulpool
- Status: Testprogramm
- Anwendung: Anwendungsübergreifend
- Paket: \$TMP
- als „Lokales Objekt“ speichern

- im TOP-Include globale Variablen zum Datentransport zwischen ABAP-Programm und Dynpro (und zurück) ergänzen:

DATA:

```
VOLUMEN TYPE p DECIMALS 3 VALUE '0',  
STRECKE LIKE VOLUMEN,  
VERBRAUCH LIKE VOLUMEN,  
OK_CODE LIKE sy-ucomm,  
OK_CODE_SAVED LIKE OK_CODE.
```


- Unterhalb des Programmes Dynpro „Eingabe Treibstoffvolumen und gefahrene Strecke“ mit Nr. 100 anlegen
- Im Dynpro Knopf „➔ Layout“ (Strg + F7) drücken
- Im Layout zwei Textfelder, `VOLUMEN_TEXT` mit Text „Treibstoffvolumen“ und `STRECKE_TEXT` mit Text „gefahrene Strecke“, untereinander anlegen
- Mit F6 (oder Bildschirmknopf mit orangefarbenem Rechteck) Liste der Programmfelder einblenden lassen
- Felder `VOLUMEN` und `STRECKE` aus Programm holen und im Layout neben den entsprechenden Textfeldern platzieren
- Knopf `BERECHNEN_KNOPF` mit Text „Verbrauch berechnen“ anlegen und Funktionscode `CALCULATE` vergeben
- Zurück zu Elementliste des Dynpros 100
- in Feld vom Typ `OK` als Name `OK_CODE` eintragen

- GUI-Status `STATUS100` zum Programm anlegen
- mit Vorlagestatus (Liststatus) abgleichen
- Menüleiste um Menü „Extras“ mit Untermenü `CALCULATE` (Funktionscode) und „Verbrauch berechnen“ (Text) ergänzen
- Drucktastenleiste (Position 6) und Funktionstasten (F8) mit Funktionscode `CALCULATE` ergänzen
- Über Funktionstaste mit grünem Kreis und weißem Häkchen Funktionscode `CALCULATE` ergänzen

- in Ablauflogik des Dynpros 100 `MODULE STATUS_0100`.
„einkommentieren“ und mit Doppelklick auf das Modul anlegen
- im `MODULE STATUS_0100 OUTPUT`. soeben anlegten Status `STATUS100` setzen (auskommentierte Zeile nutzbar)

- in Ablauflogik des Dynpros 100 MODULE USER_COMMAND_0100. „einkommentieren“ und mit Doppelklick auf das Modul anlegen
- Im MODULE USER_COMMAND_0100 INPUT. folgenden Quelltext ergänzen:

```
CASE ok_code_saved.  
  WHEN 'CALCULATE'.  
    PERFORM VerbrauchBerechnen.  
    LEAVE TO SCREEN 200.  
  WHEN 'BACK' OR '%EX' OR 'RW'.  
    LEAVE TO SCREEN 0.  
ENDCASE.
```

- in Ablauflogik des Dynpros 100 **vor** MODULE USER_COMMAND_0100. das MODULE clear_ok_code. aufrufen und mit Doppelklick auf das Modul anlegen
- Im MODULE clear_ok_code INPUT. folgenden Quelltext ergänzen:
OK_CODE_SAVED = OK_CODE.
CLEAR OK_CODE.

- Im MODULE USER_COMMAND_0100 INPUT. mit Doppelklick auf Unterprogrammaufruf VerbrauchBerechnen Unterprogramm anlegen
- In Unterprogramm VerbrauchBerechnen folgenden Quelltext ergänzen:

```
IF STRECKE <> 0.  
    VERBRAUCH = VOLUMEN / STRECKE * 100.  
ELSE.  
    VERBRAUCH = 9999.  
ENDIF.
```

- Unterhalb des Programmes Dynpro „Ausgabe Verbrauch“ mit Nr. 200 anlegen
- Im Dynpro Knopf „➔ Layout“ (Strg + F7) drücken
- Im Layout ein Textfeld, `VERBRAUCH_TEXT` mit Text „Durchschnittsverbrauch“, anlegen
- Mit F6 (oder Bildschirmknopf mit orangefarbenem Rechteck) Liste der Programmfelder einblenden lassen
- Feld `VERBRAUCH` aus Programm holen und im Layout neben dem entsprechenden Textfeld platzieren
- Feld `VERBRAUCH` nur für die Ausgabe zulassen
- Knopf `ZURUECK_KNOPF` mit Text „Zurück“ anlegen und Funktionscode `BACK` vergeben
- Zurück zu Elementliste des Dynpros 200
- in Feld vom Typ `OK` als Name `OK_CODE` eintragen

- GUI-Status `STATUS200` zum Programm anlegen
- mit Vorlagestatus (Liststatus) abgleichen
- Menüleiste um Menü „Extras“ mit Untermenü `BACK` (Funktionscode) und „Zurück“ (Text) ergänzen

- in Ablauflogik des Dynpros 200 `MODULE STATUS_0200`.
„einkommentieren“ und mit Doppelklick auf das Modul anlegen
- im `MODULE STATUS_0200 OUTPUT`. soeben anlegten Status `STATUS200` setzen (auskommentierte Zeile nutzbar)

- in Ablauflogik des Dynpros 200 `MODULE USER_COMMAND_0200.` „einkommentieren“ und mit Doppelklick auf das Modul anlegen
- Im `MODULE USER_COMMAND_0200 INPUT.` folgenden Quelltext ergänzen:

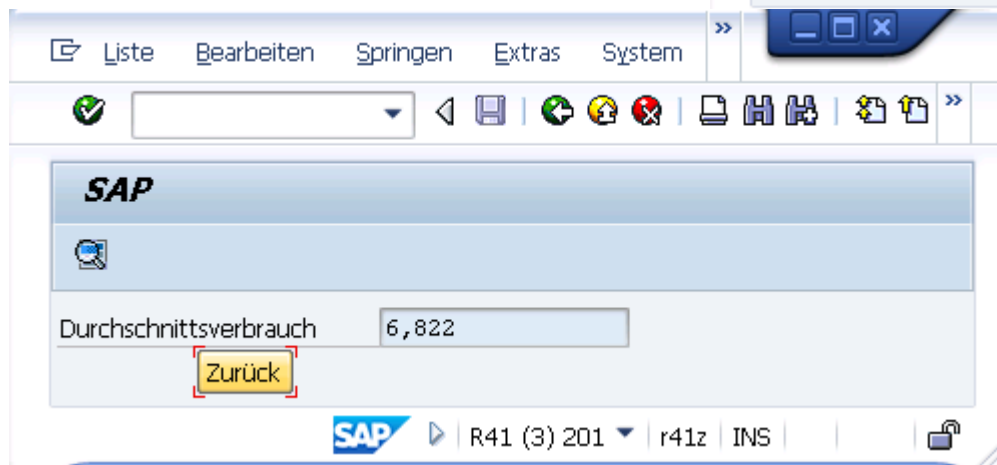
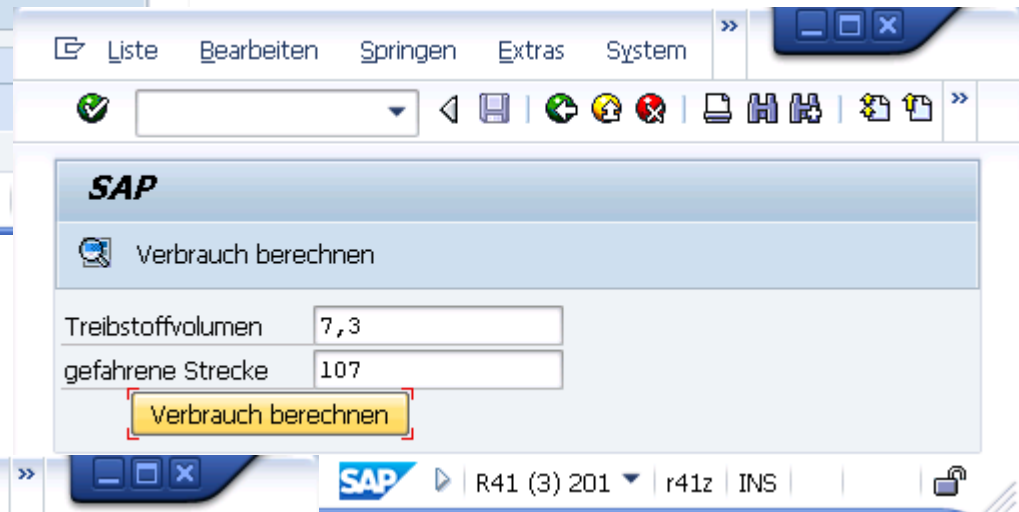
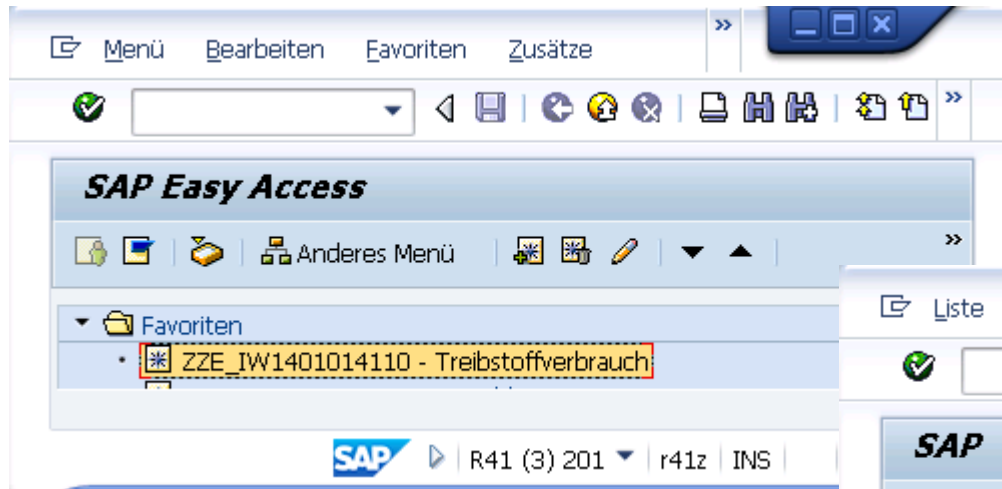
```
CASE ok_code_saved.  
  WHEN 'BACK'.  
    LEAVE TO SCREEN 100.  
  WHEN '%EX' OR 'RW'.  
    LEAVE TO SCREEN 0.  
ENDCASE.
```

- in Ablauflogik des Dynpros 200 **vor** `MODULE USER_COMMAND_0200.` das bereits existierende `MODULE clear_ok_code.` aufrufen

- Transaktion ZAB_IW20xxxxx100 „Treibstoffverbrauch“ zu Programm anlegen
- in neuem SAP-GUI-Fenster Transaktion im SAP-Easy-Access-Menü in die Favoriten eintragen

- Bitte ausführlich testen
- Wird korrekt gerechnet?
- Was passiert wenn 0 Kilometer Strecke eingegeben werden?
- Wirken alle Knöpfe, Menüeinträge, Funktionstasten wie erwartet?
 - Funktionieren alle Knöpfe zum Berechnen?
 - Gelangt man von Dynpro 200 zurück zu Dynpro 100?
 - Kann man das Programm über „Abbrechen“ (weißes Kreuz auf rotem Kreis und „Verlassen“ (weißer Pfeil nach oben of gelbem Kreis) in jedem Dynpro abbrechen?
 - ...

Ergebnis Praktikumsaufgabe 10



- Ergänzen Sie das Programm aus Praktikumsaufgabe 10 durch ein Statistik-Modul. Bevor man das Programm beendet, soll man über einen Button „Statistik“ in dieses Modul verzweigen können. Das Modul soll anzeigen, wie viele Fahrten durchgeführt, wie viele Kilometer insgesamt zurückgelegt wurden und wie hoch der Durchschnittsverbrauch insgesamt ist.

- im TOP-Include globale Variablen zum Datentransport zwischen ABAP-Programm und Dynpro (und zurück) ergänzen:

```
VOLUMEN_GESAMT LIKE VOLUMEN,  
STRECKE_GESAMT LIKE STRECKE,  
ANZAHL_FAHRTEN TYPE i VALUE 0,  
DURCHSCHNITTSVERBRAUCH LIKE VERBRAUCH,  
LETZTES_DYNPRO LIKE sy-dynnr,
```

- Unterhalb des Programmes Dynpro „Verbrauchsstatistik“ mit Nr. 300 anlegen
- Im Dynpro Knopf „➔ Layout“ (Strg + F7) drücken
- Im Layout vier Textfelder untereinander anlegen:
 - VOLUMEN_GESAMT_TEXT mit Text „Treibstoffgesamtvolumen“,
 - STRECKE_GESAMT_TEXT mit Text „insgesamt zurückgelegte Strecke“,
 - ANZAHL_FAHRTEN_TEXT mit Text „Anzahl Fahrten“ und
 - DURCHSCHNITTsverbrauch_TEXT mit Text „Durchschnittsverbrauch“.
- Mit F6 (oder Bildschirmknopf mit orangefarbenem Rechteck) Liste der Programmfelder einblenden lassen
- Felder VOLUMEN_GESAMT, STRECKE_GESAMT, ANZAHL_FAHRTEN und DURCHSCHNITTsverbrauch aus Programm holen und im Layout neben den entsprechenden Textfeldern platzieren
- alle vier Felder nur für die Ausgabe zulassen
- Knopf ZURUECK_KNOPF mit Text „Zurück“ anlegen und Funktionscode BACK vergeben
- Zurück zu Elementliste des Dynpros 300
- in Feld vom Typ OK als Name OK_CODE eintragen

- GUI-Status `STATUS300` zum Programm anlegen
- mit Vorlagestatus (Liststatus) abgleichen
- Menüleiste um Menü „Extras“ mit Untermenü `BACK` (Funktionscode) und „Zurück“ (Text) ergänzen

- in Ablauflogik des Dynpros 300 `MODULE STATUS_0300`. „einkommentieren“ und mit Doppelklick auf das Modul anlegen
- im `MODULE STATUS_0300 OUTPUT`. soeben anlegten Status `STATUS300` setzen (auskommentierte Zeile nutzbar)
- neuen Modulaufruf `MODULE DurchschnVerbrauchBerechnen`. im PBO eintragen und mit Doppelklick auf das Modul `MODULE DurchschnVerbrauchBerechnen OUTPUT`. anlegen
- im Modul mittels `PERFORM BerechneDurchschnittsverbrauch`. Unterprogramm zur Durchschnittsverbrauchsberechnung aufrufen

- mittels Doppelklick auf Unterprogrammaufruf
BerechneDurchschnittsVerbrauch entsprechendes
Unterprogramm anlegen
- Unterprogramm wie folgt ausgestalten:

```
IF STRECKE_GESAMT <> 0.  
    DURCHSCHNITTSVERBRAUCH  
        = VOLUMEN_GESAMT / STRECKE_GESAMT * 100.  
ELSE.  
    DURCHSCHNITTSVERBRAUCH = 9999.  
ENDIF.
```

- in Ablauflogik des Dynpros 300 `MODULE USER_COMMAND_0300.` „einkommentieren“ und mit Doppelklick auf das Modul anlegen
- Im `MODULE USER_COMMAND_0300 INPUT.` folgenden Quelltext ergänzen:

```
CASE ok_code_saved.  
  WHEN 'BACK'.  
    LEAVE TO SCREEN LETZTES_DYNPRO.  
  WHEN '%EX' OR 'RW'.  
    LEAVE TO SCREEN 0.  
ENDCASE.
```

- in Ablauflogik des Dynpros 300 **vor** `MODULE USER_COMMAND_0300.` das bereits existierende `MODULE clear_ok_code.` aufrufen

- sowohl für GUI-Status `STATUS100` als auch für GUI-Status `STATUS200`:
 - im Menü „Extras“ Untermenü `STATS` (Funktionscode) und „Statistik ...“ (Text) ergänzen
 - Drucktastenleiste (Position 7) mit Funktionscode `STATS` ergänzen
 - Funktionstasten (F7) mit Funktionscode `STATS` ergänzen

- sowohl in `MODULE user_command_0100 INPUT.` als auch in `MODULE user_command_0200 INPUT.` Behandlung des Funktionscodes STATS ergänzen:

```
WHEN 'STATS'.  
    LETZTES_DYNPRO = sy-dynnr.  
    LEAVE TO SCREEN 300.
```

- im `MODULE user_command_0100 INPUT.` im `WHEN 'CALCULATE'.` Unterprogrammaufruf `PERFORM VolStreckeFahrtenAkkumulieren.` ergänzen
- Unterprogramm durch Doppelklick auf `VolStreckeFahrtenAkkumulieren` anlegen
- Unterprogramm mit folgende Zeilen füllen:
`VOLUMEN_GESAMT = VOLUMEN_GESAMT + VOLUMEN.`
`STRECKE_GESAMT = STRECKE_GESAMT + STRECKE.`
`ANZAHL_FAHRTEN = ANZAHL_FAHRTEN + 1.`

- Bitte testen Sie Ihr Programm ausführlich.
- Testen Sie bitte insbesondere die Navigation zwischen den drei Dynpros und dem Aufruf der Transaktion.
- Testen Sie bitte insbesondere die Aggregation der Größen und die Berechnung des Durchschnittsverbrauchs.

Ergebnis Praktikumsaufgabe 11

The screenshots illustrate the following steps in the SAP Easy Access interface:

- SAP Easy Access**: The 'Favoriten' (Favorites) menu is expanded, showing the entry **ZZE_IW14010141110 - Treibstoffverbrauch**.
- SAP**: The 'Verbrauch berechnen' (Calculate Consumption) button is highlighted.
- SAP**: The 'Statistik ...' (Statistics ...) button is highlighted.
- SAP**: The 'Zurück' (Back) button is highlighted.

The final screenshot shows the calculated results:

Field	Value
Treibstoffgesamt volumen	24,000
insgesamt zurückgelegte Strecke	300,000
Anzahl Fahrten	3
Durchschnittsverbrauch	8,000

- a) Zeigen Sie bitte (selektierte) Lieferantenstammdaten in einem TableControl in einem Dynpro 2000 an. Die Selektionsparameter für den Ort und Anlegezeitraum sind in einem Startdynpro 1000 einzugeben.
- b) Im TableControl in Dynpro 2000 sollen Wertänderungen möglich sein, die dann im Dynpro 3000 angezeigt werden.
- c) Im TableControl im Dynpro 2000 sollen Zeilen ausgewählt werden können, die dann in einem neuen Dynpro 3000 angezeigt werden.

- Modulpool SAPMZAB_IW20xxxxx120 vom Typ „Programm“ anlegen
- mit TOP-Include (MZAB_IW20xxxxx120TOP)
- Typ: Modulpool
- Status: Testprogramm
- Anwendung: Anwendungsübergreifend
- Paket: \$TMP
- als „Lokales Objekt“ speichern

- Dynpronummer: 2000
- Kurzbeschreibung: Lieferanten anzeigen und ändern

12.2 Table-Control-Bereich anlegen

- im Änderungsmodus zum Layout des Dynpros 2000 navigieren (Knopf „➔Layout“)
- Table-Control (**nicht mittels Wizard!**) einfügen, das das Dynprofenster möglichst gut ausfüllt
- Table-Control benennen; Name: TC

- Dict/Programmfelder-Fenster (F6, Knopf „orangefarbenes Rechteck auf blauem Hintergrund“) öffnen
- Tabellen-/Feldname: LFA1
- Knopf „Holen aus Dict“
- Felder LIFNR, NAME1, ORT01, ERNAM, ERDAT und TELF1 markieren und übernehmen
- Felder in Table-Control TC „fallenlassen“
- Spalte „Telefon-1“ in zweiter Tabellenzeile direkt hinter Spalte „Name“ ziehen

- Attribute-Fenster (F2, Knopf „gelbes Rechteck auf blauem Hintergrund“) öffnen
- **Nur die folgenden Änderungen vornehmen, keine anderen**
- Trennlinien horizontal und vertikal
- Zeilenmarkierung mehrfach
- Spaltenmarkierung ohne
- Mit Markierungsspalte; Name der Markierungsspalte: MARK
- Fixe Spalten: 1

- Name OK_CODE beim Typ des Dynproelementes OK eintragen (erste Spalte, letzte Zeile)
- Bei den Feldern vom Typ I/O LFA1-LIFNR, LFA1-NAME1, LFA1-ORT, LFA1-ERDAT und LFA1-ERNAM jeweils „Eingabe“ ab- und „Nur Ausgabe“ anwählen
- Beim Feld vom Typ I/O LFA1-TELF1 „Eingabe“ an- und „Nur Ausgabe“ abwählen
(Außer der Telefonnummer soll im Table-Control nichts geändert werden können.)

- Im TOP-INCLUDE:

```
CONTROLS TC TYPE TABLEVIEW USING SCREEN '2000'.
```


- im TOP-Include:

```
TYPES: BEGIN OF st_tc,  
    LIFNR LIKE LFA1-LIFNR,  
    NAME1 LIKE LFA1-NAME1,  
    ORT01 LIKE LFA1-ORT01,  
    ERDAT LIKE LFA1-ERDAT,  
    ERNAM LIKE LFA1-ERNAM,  
    TELF1 LIKE LFA1-TELF1,  
END OF st_tc.
```

- Im TOP-INCLUDE:
`TABLES LFA1 .`

- Im Top-Include:

DATA:

```
wa_tc TYPE st_tc,  
it_tc TYPE TABLE OF st_tc,  
mark, "für die Zeilenmarkierung"  
OK_CODE TYPE sy-ucomm,  
OK_CODE_SAVED TYPE sy-ucomm.
```

```
PROCESS BEFORE OUTPUT.
```

```
  MODULE STATUS_2000.
```

```
  MODULE laden_interne_tabelle.
```

```
  LOOP AT it_tc INTO wa_tc WITH CONTROL TC.
```

```
    MODULE tc_zeile_laden.
```

```
  ENDLOOP.
```

```
PROCESS AFTER INPUT.
```

```
  LOOP AT it_tc.
```

```
    MODULE aktualisieren_interne_tabelle.
```

```
  ENDLOOP.
```

```
  MODULE USER_COMMAND_2000.
```

- jeweils durch Doppelklick auf den Modulaufruf in der Ablauflogik

```
MODULE status_2000 OUTPUT.
```

```
    SET PF-STATUS SPACE.
```

```
    *  SET TITLEBAR 'xxx'.
```

```
ENDMODULE.
```

```
MODULE laden_interne_tabelle OUTPUT.
```

```
    SELECT * FROM LFA1
```

```
        INTO CORRESPONDING FIELDS OF TABLE it_tc.
```

```
    DESCRIBE TABLE it_tc LINES TC-lines.
```

```
ENDMODULE.
```

```
MODULE tc_zeile_laden OUTPUT.
```

```
    MOVE-CORRESPONDING wa_tc TO LFA1.
```

```
ENDMODULE.
```

- jeweils durch Doppelklick auf den Modulaufruf in der Ablauflogik

```
MODULE aktualisieren_interne_tabelle INPUT.
```

```
  MOVE-CORRESPONDING LFA1 TO wa_tc.
```

```
  MODIFY it_tc FROM wa_tc INDEX TC-current_line.
```

```
ENDMODULE.
```

```
MODULE user_command_2000 INPUT.
```

```
  ok_code_saved = ok_code.
```

```
  CLEAR ok_code.
```

```
  CASE ok_code_saved.
```

```
    WHEN 'BACK' OR '%EX' OR 'RW'.
```

```
      LEAVE TO SCREEN 0.
```

```
    WHEN OTHERS.
```

```
      MESSAGE 'Die Programmfunktion ist nicht belegt.'
```

```
      TYPE 'I'.
```

```
  ENDCASE.
```

```
ENDMODULE.
```

- Transaktionscode: ZAB_IW20xxxxx120
- Kurztext: Lieferanten-Table-Control
- Programm: SAPMZAB_IW20xxxxx120
- Dynpronummer: 2000
- Transaktion im SAP-Menü in die Favoriten einfügen

- Weiter- bzw. Enter-, Zurück-, Verlassen- und Abbrechen-Knopf testen
- Änderungen in der „Telefonspalte“ vornehmen
- Zeilen markieren

- Dynpronummer: 1000
- Kurzbeschreibung: „Selektionsparameter eingeben“

- Ins Layout des Dynpros 1000 wechseln
- Subscreen-Bereich anlegen, der das Dynpro-Fenster gut ausfüllt
- Subscreen-Bereich benennen; Name: SEL_SUB_SCR

- Name OK_CODE beim Typ des Dynproelementes OK eintragen (erste Spalte, letzte Zeile) in Dynpro 1000

- ins TOP-Include wechseln
- direkt vor dem Table-Control bzw. Table-View folgenden Selection-Screen vereinbaren:

```
SELECTION-SCREEN BEGIN OF SCREEN 1100 AS SUBSCREEN.  
  SELECT-OPTIONS:  
    so_ort01 FOR LFA1-ORT01 LOWER CASE NO INTERVALS,  
    so_erdat FOR LFA1-ERDAT.  
SELECTION-SCREEN END OF SCREEN 1100.
```

PROCESS BEFORE OUTPUT.

```
CALL SUBSCREEN SEL_SUB_SCR  
    INCLUDING sy-cprog '1100'.  
MODULE STATUS_1000.
```

PROCESS AFTER INPUT.

```
CALL SUBSCREEN SEL_SUB_SCR.  
MODULE USER_COMMAND_1000.
```

```
MODULE status_1000 OUTPUT.  
    SET PF-STATUS SPACE.  
ENDMODULE.  
  
MODULE user_command_1000 INPUT.  
    ok_code_saved = ok_code.  
    CLEAR ok_code.  
    CASE ok_code_saved.  
        WHEN ''.  
            LEAVE TO SCREEN 2000.  
        WHEN 'BACK' OR '%EX' OR 'RW'.  
            LEAVE TO SCREEN 0.  
        WHEN OTHERS.  
            MESSAGE  
                'Die Programmfunktion ist nicht belegt.'  
            TYPE 'I'.  
        ENDCASE.  
    ENDMODULE.
```

12.21 Füllen der internen Tabelle in Dynpro 2000 aufgrund der Selektionsparameter einschränken

```
MODULE laden_interne_tabelle OUTPUT.  
  SELECT * FROM LFA1  
        INTO CORRESPONDING FIELDS OF TABLE it_tc  
        WHERE ORT01 IN so_ort01  
        AND ERDAT IN so_erdat.  
  DESCRIBE TABLE it_tc LINES TC-lines.  
ENDMODULE.
```

12.22 Zurück-Knopf in Table-Control-Dynpro auf Selektionsparameter-Dynpro verweisen lassen

```
MODULE user_command_2000 INPUT.  
    ok_code_saved = ok_code.  
    CLEAR ok_code.  
    CASE ok_code_saved.  
        WHEN 'BACK'.  
            LEAVE TO SCREEN 1000.  
        WHEN '%EX' OR 'RW'.  
            LEAVE TO SCREEN 0.  
        WHEN OTHERS.  
            MESSAGE  
                'Die Programmfunktion ist nicht belegt.'  
            TYPE 'I'.  
    ENDCASE.  
ENDMODULE.
```


- Start-Dynpro (Dynpronummer) der Transaktion ZAB_IW20xxxxx120 von 2000 auf 1000 ändern

- Selektion nach Ort testen, z. B. Dresden
- Selektion nach Ort mit Wildcards testen, z. B. Dres*, *esd* oder *sden
- Selektion nach Zeitraum testen, z. B. 01.12.2021 bis 31.03.2022
- Verlassen des Dynpros 1000 testen
- Rücksprung vom Dynpro 2000 testen

12.25 Dynpro 3000 zur Anzeige geänderter/markierter Datensätze anlegen

- Dynpronummer: 3000
- Kurzbeschreibung: „Auswahl/Änderung anzeigen“

12.26 Interne Tabellen für geänderte und markierte Datensätze anlegen

- in das Top-Include wechseln
- in der DATA-Section interne Tabelle für markierte Datensätze ergänzen:

DATA:

```
it_mrkd TYPE TABLE OF st_tc,  
it_chngd TYPE TABLE OF st_tc.
```

- Im PAI des Dynpros 2000 **vor dem LOOP** folgende Zeile ergänzen:

```
MODULE leeren_itab_markierter_zeilen.
```

```
MODULE leeren_itab_geaenderter_zeilen.
```

- beide Module wie folgt ausgestalten:

```
MODULE leeren_itab_markierter_zeilen INPUT.
```

```
  CLEAR it_mrkd.
```

```
ENDMODULE.
```

```
MODULE leeren_itab_geaenderter_zeilen INPUT.
```

```
  CLEAR it_chngd.
```

```
ENDMODULE.
```

- Im PAI des Dynpros 2000 am Ende (innerhalb) des `LOOPS` folgende Zeile ergänzen:

```
FIELD mark
  MODULE sammeln_markierter_zeilen
  ON REQUEST.
FIELD LFA1-TELF1
  MODULE sammeln_geaenderter_zeilen
  ON REQUEST.
```

- beide Module wie folgt ausgestalten:

```
MODULE sammeln_markierter_zeilen INPUT.
  MOVE-CORRESPONDING LFA1 TO wa_tc.
  APPEND wa_tc TO it_mrkd.
ENDMODULE.

MODULE sammeln_geaenderter_zeilen INPUT.
  MOVE-CORRESPONDING LFA1 TO wa_tc.
  APPEND wa_tc TO it_chngd.
ENDMODULE.
```

- Name OK_CODE beim Typ des Dynproelementes OK eintragen (erste Spalte, letzte Zeile) in Dynpro 3000

PROCESS BEFORE OUTPUT.

MODULE STATUS_3000.

MODULE ausgeben_geanderter_markierter.

PROCESS AFTER INPUT.


```
MODULE status_3000 OUTPUT.  
    SET PF-STATUS SPACE.  
ENDMODULE.
```

12.33 Ausgabe im Table-Control geänderter und markierter Datensätze

```
MODULE ausgeben_geanderter markierter OUTPUT.  
  WRITE: /1 'Geänderte Zeilen:'.  
  WRITE: /1 'LIFNR', 'NAME1', 'ORT01', 'TELF1',  
    'ERDAT', 'ERNAM'.  
  DATA wa_lfa1 like LFA1.  
  LOOP AT it_chngd INTO wa_tc.  
    WRITE: /1 wa_tc-LIFNR, wa_tc-NAME1,  
      wa_tc-ORT01, wa_tc-te1f1, wa_tc-ERDAT,  
      wa_tc-ERNAM.  
  ENDLOOP.  
  WRITE: /1 '---'.  
  WRITE: /1 'Markierte Zeilen:'.  
  WRITE: /1 'LIFNR', 'NAME1', 'ORT01', 'TELF1',  
    'ERDAT', 'ERNAM'.  
  LOOP AT it_mrkd INTO wa_tc.  
    WRITE: /1 wa_tc-LIFNR, wa_tc-NAME1,  
      wa_tc-ORT01, wa_tc-te1f1, wa_tc-ERDAT,  
      wa_tc-ERNAM.  
  ENDLOOP.  
  LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0.  
  SUPPRESS DIALOG.  
ENDMODULE.
```

12.34 Weiter-Knopf in Table-Control-Dynpro 2000 das Dynpro 3000 rufen lassen

```
MODULE user_command_2000 INPUT.  
    ok_code_saved = ok_code.  
    CLEAR ok_code.    CASE ok_code_saved.  
        WHEN ''.  
            CALL SCREEN 3000.  
        WHEN 'BACK'.  
            LEAVE TO SCREEN 1000.  
        WHEN '%EX' OR 'RW'.  
            LEAVE TO SCREEN 0.  
        WHEN OTHERS.  
            MESSAGE  
                'Die Programmfunktion ist nicht belegt.'  
                TYPE 'I'.  
    ENDCASE.  
ENDMODULE.
```

12.35 Testen der Anzeige geänderter und markierter Datensätze

- Telefonnummern mehrerer Datensätze ändern
- Mehrere Datensätze markieren
- Ein paar Datensätze markieren, ein paar andere ändern
- Ein paar Datensätze markieren, ein paar andere ändern, ein paar weitere ändern und markieren
- Ein paar Datensätze markieren und ändern
- Immer über Zurück-Button testen, um das Initialisieren der Tabelle der markierten Datensätze zu überprüfen