

# ABAP Workbench: The Big Picture

---



**Alex Gönczy**

ABAP DEVELOPER

@abapmentor <http://abapmentor.expertise-team.com/>



# Standard Solution for Everyone?



## SAP **Industry** Solutions



Mercedes AMG  
McLaren Group

Automotive



FC Bayern München  
NBA

Sports and  
Entertainment



Bank of America  
Deutsche Bank

Banking



# What Is ABAP?



Screen Painter: Display Screen for SAPMV45A

Screen number 4440 Active

Attributes Element list Flow logic

```
52 | * Not used because Listbox
53 | * Text zu Bestellart des Kunden ermitteln
54 MODULE T176T_SELECT.
55 MODULE T176T_SELECT_E.
56
57 CALL SUBSCREEN HEAD_USER INCLUDING 'SAPMV45A' '8310'.
58
59 * Vertragsdaten für Subscreen aufbauen.
60 CALL SUBSCREEN SUBSCREEN_VERTRAG
61     INCLUDING 'SAPLV45W' '0400'.
62
63 * PAI ****
64 PROCESS AFTER INPUT.
65 CALL SUBSCREEN HEAD_USER .
66
67 MODULE VBAK_BEARBEITEN_VOR.          " Bach
68
69 * Füllen des Auftragskopfes mit Vorschlagswerten
70 MODULE VBAK_FUELLEN.
71
72 * Prüfen der Gruppenkontaktnummer
```

Source code behind VA01

ABAP | Ln 11 Col 1 | N...

SAP



# ABAP Syntax



```
DATA(day_of_the_week) = selected_date MOD 7.  
  
IF day_of_the_week > 1. " Correction  
    day_of_the_week = day_of_the_week - 1.  
ENDIF.  
  
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto   = 'Budapest'.  
  
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
    ...  
ENDCLASS.  
  
simple_alv->display( ).
```

## ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z O-9 \_ ]
- ◀ Comment: full line



Next

# Standard Solution for Everyone?

ABAP Syntax



```
DATA(day_of_the_week) = selected_date MOD 7.  
  
IF day_of_the_week > 1. " Correction  
    day_of_the_week = day_of_the_week - 1.  
ENDIF.  
  
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto   = 'Budapest'.  
  
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
    ...  
ENDCLASS.  
  
simple_alv->display( ).
```

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
- ◀ Comment: full line



# Standard Solution for Everyone?

---



~50K



# Car Manufacturing



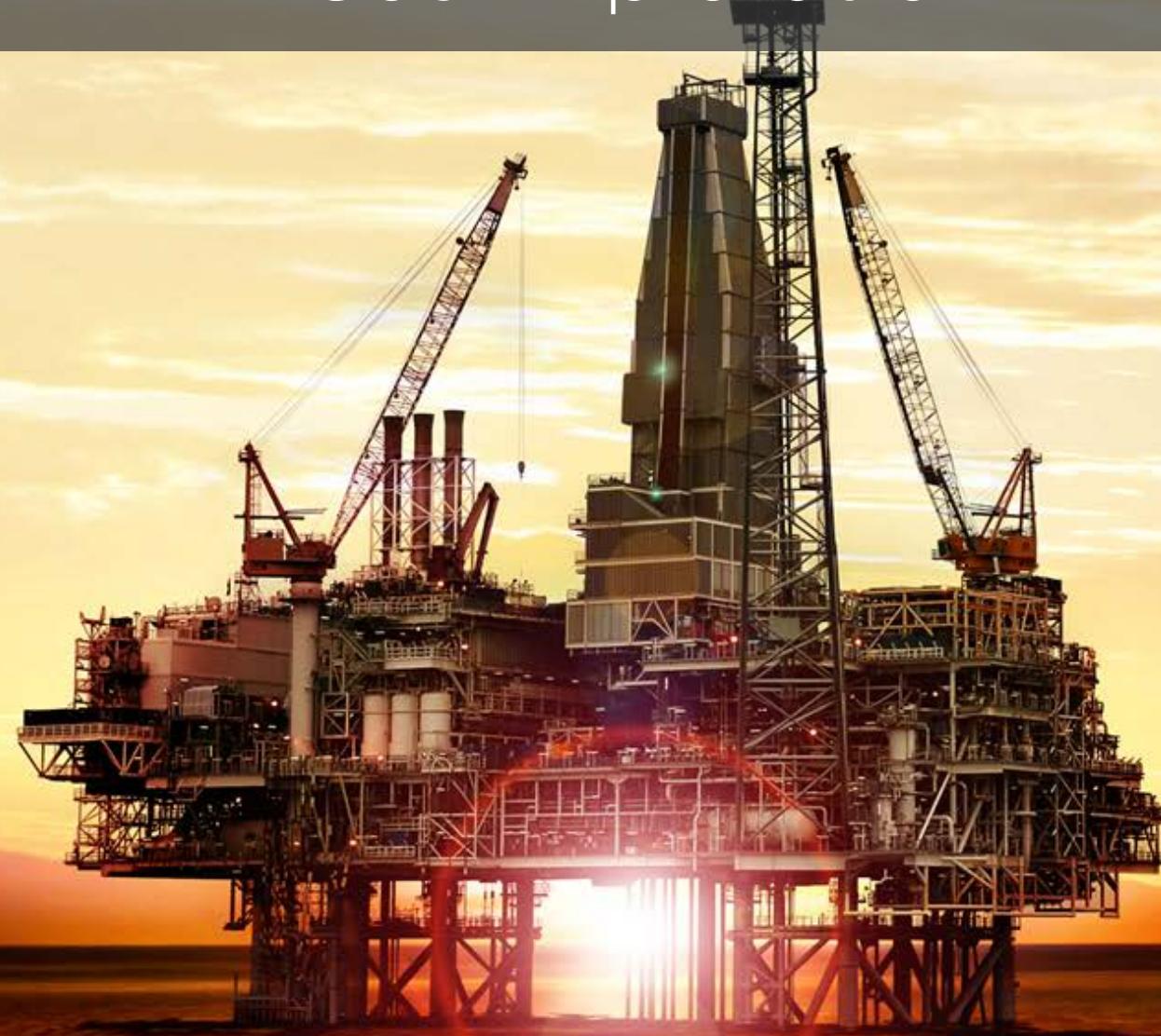
# Food Production



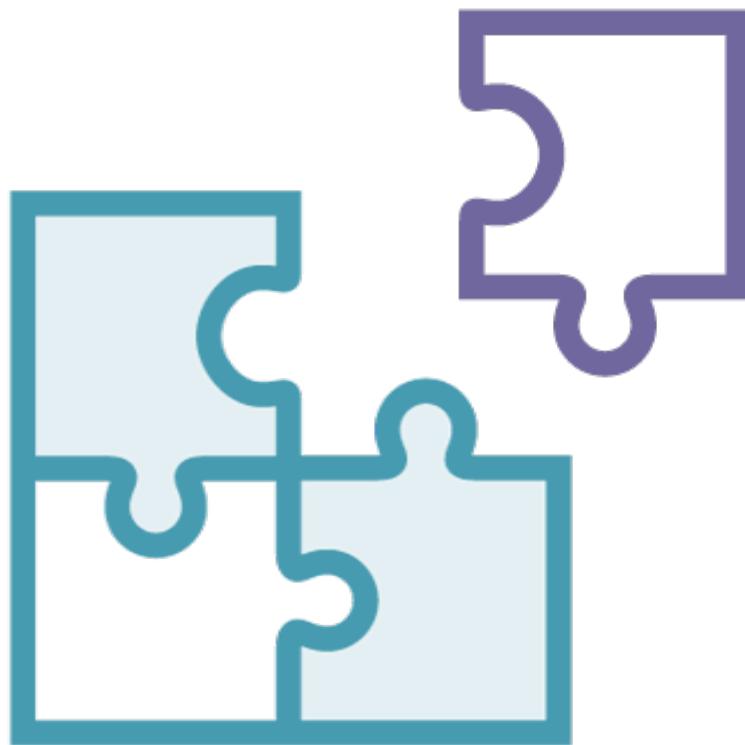
# Banking



# Gas Exploration



Standard Solution for **Everyone**



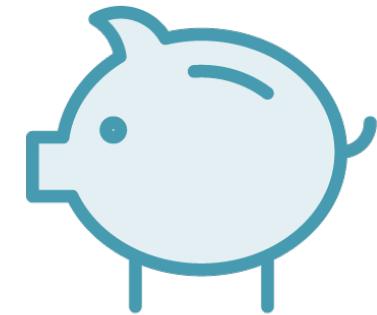
# Standard Solution for **Everyone**



**Mercedes AMG  
McLaren Group**



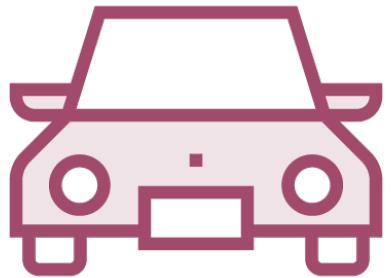
**FC Bayern München  
NBA**



**Bank of America  
Deutsche Bank**



# SAP **Industry** Solutions



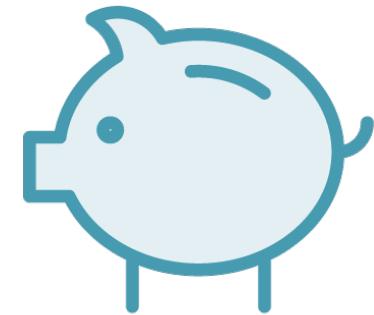
**Mercedes AMG  
McLaren Group**

**Automotive**



**FC Bayern München  
NBA**

**Sports and  
Entertainment**



**Bank of America  
Deutsche Bank**

**Banking**







**SAP** ERP

+

Automotive

**SAP** ERP

+

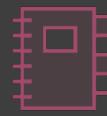
Automotive

**SAP** ERP

+

Automotive

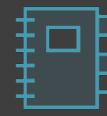




**SAP** **ERP**  
®

+

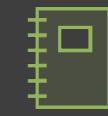
Automotive



**SAP** **ERP**  
®

+

Automotive



**SAP** **ERP**  
®

+

Automotive



# Customizing





## Display IMG

expertise team

[Existing BC Sets](#) [BC Sets for Activity](#) [Activated BC Sets for Activity](#) [Release Notes](#) [Change Log](#) [Where Else Used](#)

### Structure

- ✓ [Financial Accounting](#)
  - > [Financial Accounting Global Settings](#)
  - ✓ [General Ledger Accounting](#)
    - > [G/L Accounts](#)
    - ✓ [Business Transactions](#)
      - ✓ [G/L Account Posting](#)
        - ✓ [Make and Check Document Settings](#)
          - [Define Document Types](#)
          - [Define Posting Keys](#)
          - [Validation in Accounting Documents](#)
          - [Define Texts for Line Items](#)
          - [Define Default Values](#)
          - [Define Field Status Variants](#)
          - [Assign Company Code to Field Status Variants](#)
          - [Screen Variants for Document Entry](#)
          - [Define Subscreens for Coding Blocks](#)
          - [Substitution in Accounting Documents](#)
          - [Define Text IDs for Documents](#)
          - [Define Line Layout for Document Posting Overview](#)
          - [Define Line Layout for Document Change/Display](#)
          - [Select Standard Line Layout for Document Change/Display](#)
          - [Document Change Rules, Document Header](#)
          - [Maintain Fast Entry Screens for G/L Account Items](#)
        - > [Make and Check Settings for Document Parking](#)
        - > [G/L Account Posting - Enquiry](#)



SPRO transaction

document  
transactions



CAGR (04 – 08)  
Revenue 20%  
PBT 15%

H1 09 v H1 08:  
Revenue (2%)  
PBT 4%

### Projected sales of main products in 2013



delivered solid results in a challenging market

	Jan	Feb	Mar
Gold	\$577.00	\$462.20	+22.60%
Patinum	\$277.00	\$366.19	+12.20%
Silver	\$493.00	\$518.24	+5.00%
Copper	\$370.00	\$531.74	+38.80%
Steel	\$491.00	\$644.21	+21.00%
Bauxite	\$571.00	\$455.59	-22.80%
Manganese	\$205.00	\$240.26	+17.20%
Aluminum	\$539.00	\$578.06	+20.80%
Chrome	\$592.00	\$820.51	+38.60%
Nickel	\$574.00	\$697.98	+21.00%
Baute	\$369.00	\$503.74	+17.60%
Cotton	\$190.00	\$192.19	+3.80%
Textiles	\$280.00	\$347.48	+1.60%
Wool	\$156.00	\$192.19	23.20%
Fur	\$280.00	\$316.40	13.00%
Sateen	\$181.00	\$197.65	9.20%
Silk	\$188.00	\$198.53	5.60%
Oil	\$794.00	\$5136.15	31.70%
Gas	\$502.00	\$673.58	31.70%
Electric power	\$599.00	\$700.83	17.00%



Gold	A	Manganese	B	Wool	E	Gold	A	Manganese	B
Patinum	B	Aluminum	V	Fur	GT	Patinum	B	Aluminum	V
Silver	A	Chrome	A	Sateen	A	Silver	A	Chrome	A
Copper	A	Nickel	D	Silk	A	Copper	A	Nickel	D
Steel	D	Bauxite	F	Oil	A	Steel	D	Bauxite	F
Bauxite	A	Cotton	ERT	Gas	GT	Bauxite	A	Cotton	ERT
Manganese	WAX	Flax	A	Electric power	Was	Manganese	WAX	Flax	A

characterized by  
distribution of market  
players





## ABAP Editor: Change Enhancement ZEXP\_MV75AFT0\_COMMENT\_BUILD



← → ⌂ Enhancements | Pattern | Create | Change | Replace | Undo

Include

MV75AFT0\_COMMENT\_BUILD

Active

```
131 ELSE.
132     CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'
133         EXPORTING
134             IT_LIST_COMMENTARY = GS_SD_ALV-GRID_TOP_OF_PAGE.
135 ENDIF.
136
137 ****$--$SE: (1) Form COMMENT_BUILD--$*$*
138 *$--$Start: (1)-----
139 ENHANCEMENT 1  ZEXP_MV75AFT0_COMMENT_BUILD.      "active version
140
141     CLEAR LS_LINE.
142     LS_LINE-TYP  = 'S'.
143     LS_LINE-INFO = ' '.
144     APPEND LS_LINE TO P_GS_SD_ALV-GRID_TOP_OF_PAGE.
145
146     CLEAR LS_LINE.
147     LS_LINE-TYP  = 'S'.
148     LS_LINE-KEY  = 'Course'.
149     LS_LINE-INFO = 'ABAP Workbench: The Big Picture'.
150     APPEND LS_LINE TO P_GS_SD_ALV-GRID_TOP_OF_PAGE.
151
152     CLEAR LS_LINE.
153     LS_LINE-TYP  = 'S'.
154     LS_LINE-KEY  = 'Author'.
155     LS_LINE-INFO = 'Obermaier'.
```

Scope: \FORM COMMENT\_BUILD\ENHANCEMENT 1

| ABAP | Ln 140 Col 1

**Flight Schedule**expertise  
team

Airline	Flight No.	D. Country	Depart. city	Depart	Country	Arrival city	Target	Flight time	Departure	Arrival Time	Distance	Dis.	Charter	Day(s)
= 18:45														
AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00	14:01:00	2.572,000	MI		0
AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:21	09:00:00	17:21:00	2.572,000	MI		0
AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00	21:05:00	845,000	MI		0
AZ	788	IT	ROME	FCO	JP	TOKYO	TYO	12:55	12:00:00	08:55:00	6.130,000	MI		1
AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00	19:25:00	6.130,000	MI		0
AZ	790	IT	ROME	FCO	JP	OSAKA	KIX	13:35	10:35:00	08:10:00	6.030,000	MI	X	1
DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00	09:30:00	3.851,000	MI		1
DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:22	17:15:00	20:37:00	2.572,000	MI		0
DL	1984	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:25	10:00:00	18:25:00	2.572,000	MI		0
JL	407	JP	TOKYO	NRT	DE	FRANKFURT	FRA	12:05	13:30:00	17:35:00	9.100	KM		0
JL	408	DE	FRANKFURT	FRA	JP	TOKYO	NRT	11:15	20:25:00	15:40:00	9.100	KM	X	1
LH	400	DE	FRANKFURT	FRA	US	NEW YORK	JFK	7:24	10:10:00	11:34:00	6.162	KM		0
LH	401	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:15	18:30:00	07:45:00	6.162	KM		1
LH	402	DE	FRANKFURT	FRA	US	NEW YORK	JFK	7:35	13:30:00	15:05:00	6.162	KM	X	0
LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF	1:05	10:30:00	11:35:00	555	KM		0
LH	2407	DE	BERLIN	TXL	DE	FRANKFURT	FRA	1:05	07:10:00	08:15:00	555	KM		0
QF	5	SG	SINGAPORE	SIN	DE	FRANKFURT	FRA	13:45	22:50:00	05:35:00	10.000	KM		1
QF	6	DE	FRANKFURT	FRA	SG	SINGAPORE	SIN	11:10	20:55:00	15:05:00	10.000	KM		1
SQ	2	SG	SINGAPORE	SIN	US	SAN FRANCISCO	SFO	18:25	17:00:00	19:25:00	8.452,000	MI		0
SQ	15	US	SAN FRANCISCO	SFO	SG	SINGAPORE	SIN	18:45	16:00:00	02:45:00	8.452,000	MI		2
SQ	158	SG	SINGAPORE	SIN	ID	JAKARTA	JKT	1:35	15:25:00	16:00:00	560,000	MI		0
SQ	988	SG	SINGAPORE	SIN	JP	TOKYO	TYO	6:40	16:35:00	00:15:00	3.125,000	MI		1
UA	941	DE	FRANKFURT	FRA	US	SAN FRANCISCO	SFO	11:36	14:30:00	17:06:00	5.685,000	MI		0
UA	3504	US	SAN FRANCISCO	SFO	DE	FRANKFURT	FRA	10:30	15:00:00	10:30:00	5.685,000	MI		1
UA	3516	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:25	16:20:00	05:45:00	6.162	KM		1
IA	3517	DF	FRANKFURT	FRA	IIS	NEW YORK	FK	8:15	10:40:00	12:55:00	6.162	KM		0



Screen Painter: Display Screen SAPMV45A 4440

Screen Edit Goto Utilities(M) Environment Help

Flow logic Attributes Element List

Name Text Lg H L Cl

General header data

Sales Doc. Type						
Description						
Req. deliv.date		Deliver. Plant				
Valid from		Valid to				
Valid from		Valid to				
Complete dlv.		Total Weight	.	.	.	.
Delivery block	Volume					
Billing block	Pricing date					
Total amount	V	Doc. Currency	/	.	.	.
Payment card	Exp.date					
Card Verif.Code						
Payment terms	Incoterms					
		Delivery time	▼			
Order reason						
Sales area	/	/	/	/	/	
Promotion						
Catalog						
Catalog code						
Master contract						
Sales Office		Document Date				
Sales Group		Version				
Partner Descr.						
Pricing and statistics						

Active 79 Cl x 94 L INS EA4

From To Search

City: FRANKFURT	Country:
-----------------	----------

City: NEW YORK	Country:
----------------	----------

## Flights

Airline	Airline	Flight Number	Depart.city	Dest. airport	Dep. airport	Arrival city	Departure	Arrival Time	Arrival date	Airfare
LH	Lufthansa	0400	FRANKFURT	JFK	FRA	NEW YORK	10:10:00	11:34:00	14.02.2009	
LH	Lufthansa	0400	FRANKFURT	JFK	FRA	NEW YORK	10:10:00	11:34:00	14.03.2009	
LH	Lufthansa	0400	FRANKFURT	JFK	FRA	NEW YORK	10:10:00	11:34:00	11.04.2009	
LH	Lufthansa	0400	FRANKFURT	JFK	FRA	NEW YORK	10:10:00	11:34:00	09.05.2009	
LH	Lufthansa	0400	FRANKFURT	JFK	FRA	NEW YORK	10:10:00	11:34:00	06.06.2009	

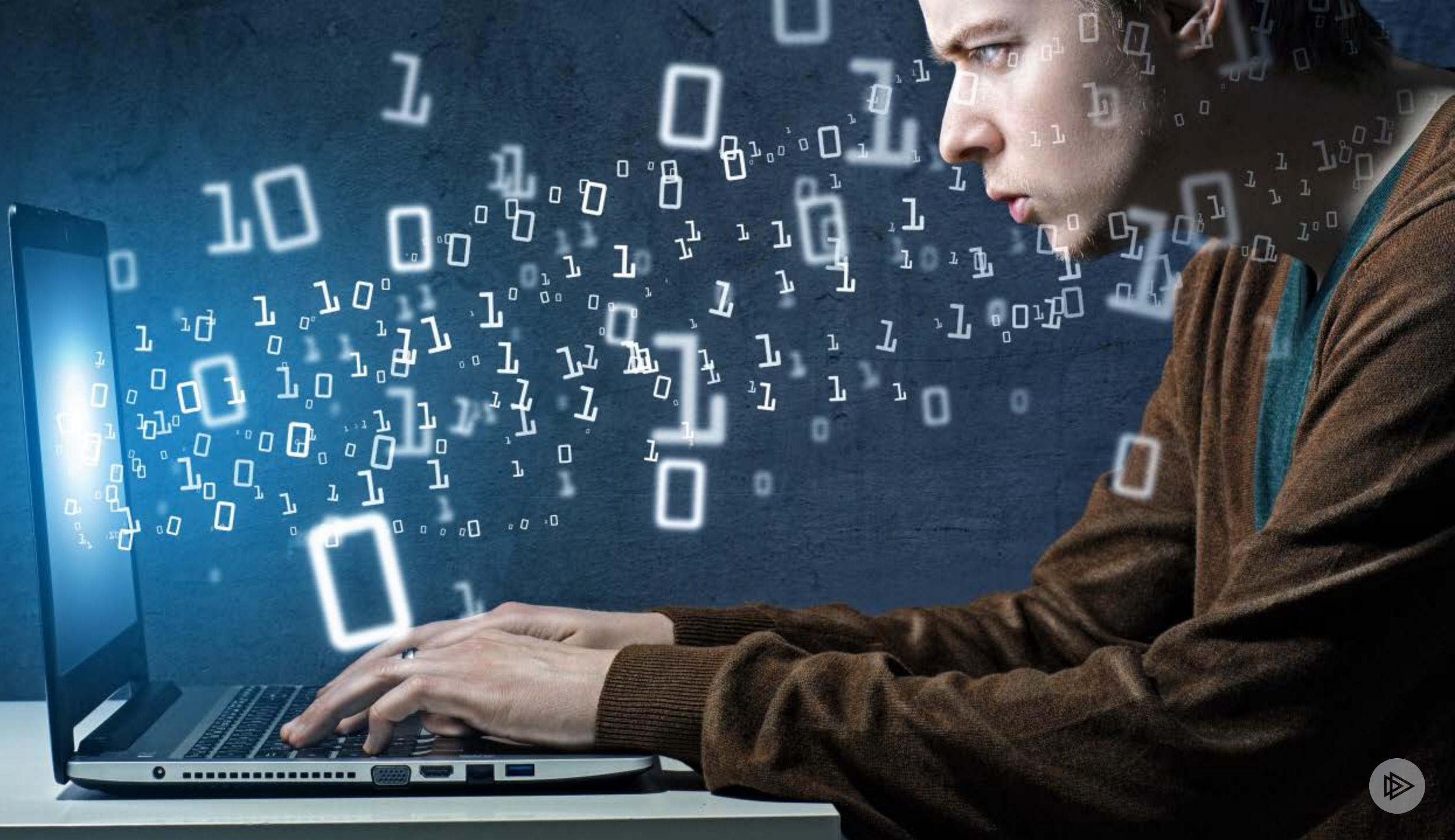
## Flight Bookings

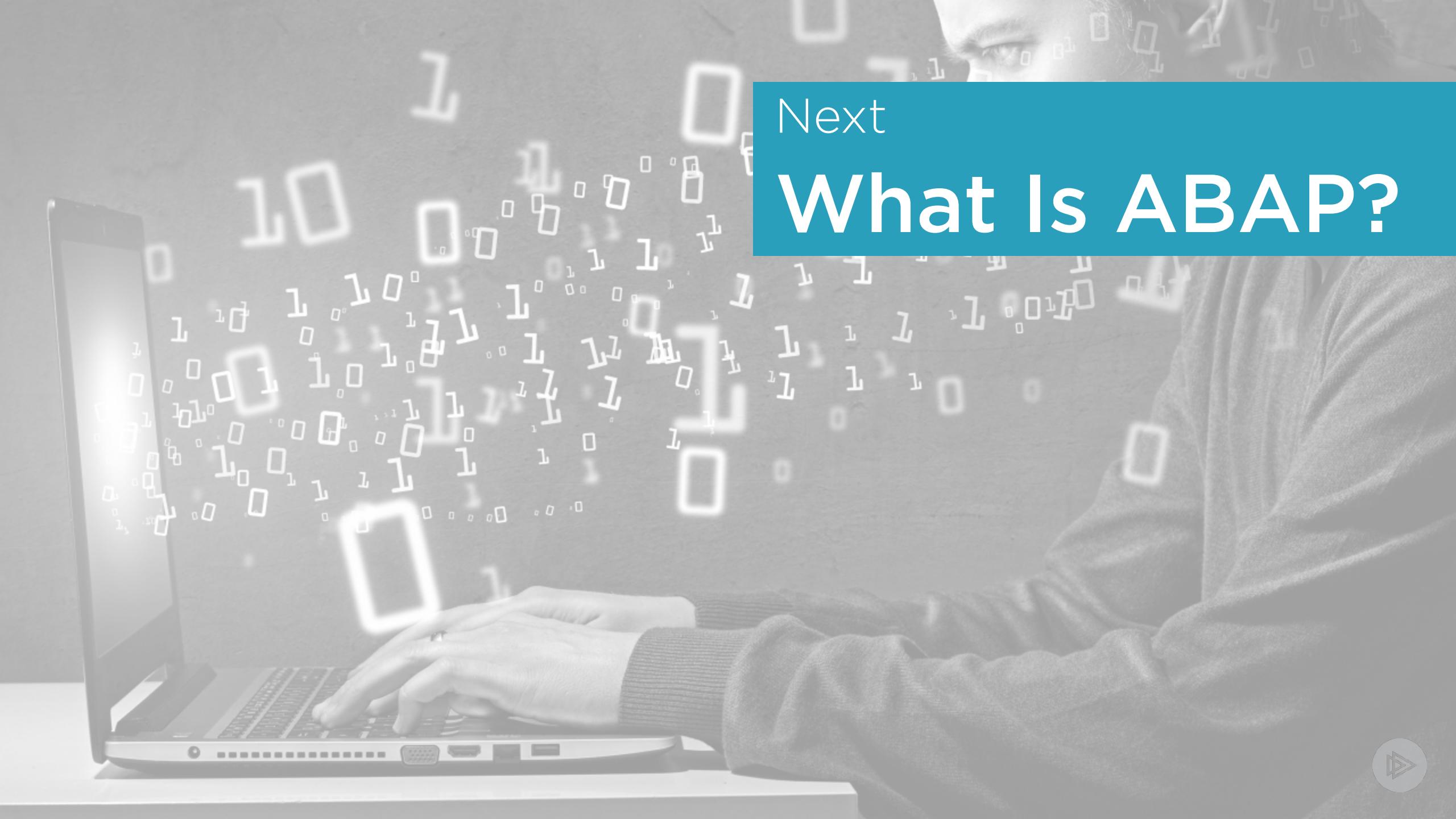
Airline	Flight Number	Date	Booking num...	Customer Num...	Passenger Name
LH	0400	11.04.2009	00000620	00001112	Salvador Fischer
LH	0400	11.04.2009	00000621	00002994	Roland Sessler
LH	0400	11.04.2009	00000622	00004437	Peter Dumbach
LH	0400	11.04.2009	00000623	00000491	Ruth Henry

## Flight Booking Details

Airline:	LH	Paymnt currency:	EUR
Flight Number:	0400	Amount:	1.198,80
Date:	11.04.2009	Airline Currency:	EUR
Booking number:	00000621	Booking date:	26.01.2009
Customer Number:	00002994	Sales office:	00000000
B/P customer:	<input checked="" type="checkbox"/>	Agency No.:	00000119





A grayscale photograph of a person wearing a hoodie, sitting at a desk and typing on a laptop keyboard. Binary code (0s and 1s) is overlaid on the image, appearing as if it's floating in the air around the person's head and hands.

Next

# What Is ABAP?



# What Is ABAP?

---



```
SELECT * FROM sbook INTO TABLE @DATA(flight_bookings)
WHERE fldate = '20091109'.
```

```
DATA(simple_alv) = NEW zcl_simple_alv(
    title = 'Flight Bookings'
).
```

```
simple_alv->display( flight_bookings ).
```

ABAP



CLI	TYPE,C'A'	TYPE = 'A' ?
BE	TYPEA	YES => JUMP TO TYPEA
CLI	TYPE,C'B'	TYPE = 'B' ?
BE	TYPEB	YES => JUMP TO TYPEB

Allgemeiner Berichts Aufbereitungs Prozessor



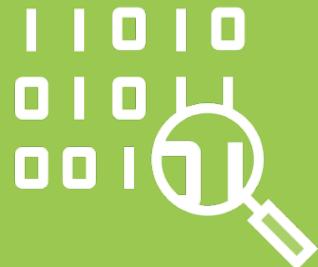
Generic Report Preparation Processor



# Advanced Business Application Programming

**ABAP**

Business Applications  
Large Part of The Basis  
Development Environment



**C / C++**

Kernel



**IF** xcode = new.

- \* E-Meldung, wenn Programm schon existiert  
**PERFORM** check\_directory. "prüft ob Programm existiert
- \* reservierte Zeichen/Namen, bei neuem Programm alle Meldungen  
**CALL FUNCTION** 'RS\_PROGRAM\_CHECK\_NAME'  
**EXPORTING**  
  progname = rs38m-programm.
- \* existierendes Programm soll geändert werden

Code Snippet of the **ABAP Editor**





## Create Standard Order: Overview

expertise team

Standard Order  Net value  0,00 Sold-To Party Ship-To Party PO Number PO date 

Sales Item overview Item detail Ordering party Procurement Shipping Reason for rejection

Req. deliv.date	<input type="text"/> D 07.12.2015	Deliver.Plant	<input type="text"/>
<input type="checkbox"/> Complete dlv.		Total Weight	<input type="text"/> 0,000 <input type="button"/>
Delivery block	<input type="text"/>	Volume	<input type="text"/> 0,000 <input type="button"/>
Billing block	<input type="text"/>	Pricing date	<input type="text"/> 26.11.2015
Payment card	<input type="text"/>	Exp.date	<input type="text"/>
Card Verif.Code	<input type="text"/>	Incoterms	<input type="text"/>
Payment terms	<input type="text"/>		
Order reason	<input type="text"/>		

All items

Item	Material	Order Quantity	Un	Description	S	Customer Material Numb	ItCa	DGIP	HL Itm	D First date	Plnt	Batch	CnTy	Amou	
					<input type="checkbox"/>					<input type="text"/> D 07.12.2015					
					<input type="checkbox"/>					<input type="text"/> D 07.12.2015					

VA01 - Create Sales Order



Screen Painter: Display Screen SAPMV45A 4440

Screen Edit Goto Utilities(M) Environment Help

Flow logic Attributes Element List

Name Text Lg H L Cl

General header data

Sales Doc. Type						
Description						
Req. deliv.date		Deliver.Plant				
Valid from		Valid to				
Valid from		Valid to				

<input type="checkbox"/> Complete dlv.		Total Weight	.	.	.	.
Delivery block		Volume	.	.	.	.
Billing block		Pricing date				
Total amount	V	Doc. Currency	/	.	.	.
Payment card		Exp.date				

Card Verif.Code						
Payment terms		Incoterms				
		Delivery time				

Order reason						
Sales area	/	/	/			
Promotion						
Catalog						
Catalog code						
Master contract						
Sales Office		Document Date				

Screen definition behind VAO1



## Screen Painter: Display Screen for SAPMV45A



Screen number 4440 Active

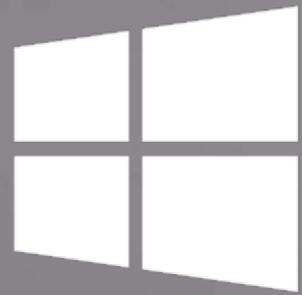
Attributes Element list Flow logic

```
52           " Not used because Listbox
53   * Text zu Bestellart des Kunden ermitteln
54   MODULE T176T_SELECT.
55   MODULE T176T_SELECT_E.
56
57   CALL SUBSCREEN HEAD_USER INCLUDING 'SAPMV45A' '8310'.
58
59   * Vertragsdaten für Subscreen aufbauen.
60   CALL SUBSCREEN SUBSCREEN_VERTRAG
61     INCLUDING 'SAPLV45W' '0400'.
62
63   * PAT *****
64 PROCESS AFTER INPUT.
65   CALL SUBSCREEN HEAD_USER .
66
67   MODULE VBAK_BEARBEITEN_VOR.          " Bach
68
69   * Füllen des Auftragskopfes mit Vorschlagswerten
70   MODULE VBAK_FUELLEN.
71
72   * Prüfen der Gruppenkontaktnummer
```

Source code behind **V0A1**

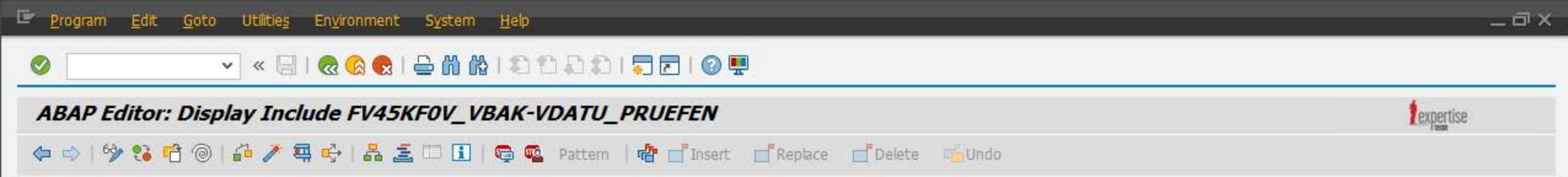
| ABAP | Ln 11 Col 1





Windows® 8





Include FV45KF0V\_VBAK-VDATU\_PRUEFEN Active

```
37         t180-trtyp ne chara and
38         checked_vdatu is initial .
39
40
41   if ( t180-trtyp = charv and checked_vdatu is initial ).
42     da_warning = space.
43   else.
44     da_warning = 'X'.
45   endif.
46
47   clear: checked_vdatu.
48
49   ----- check the requested delivery date -----
50
51   * 1. check for date in the past
52   *      and convert date format (screen/external into req.del.date)
53   call function 'PERIOD_AND_DATE_CONVERT_INPUT'
54     exporting
55       dialog_date_is_in_the_past = da_warning
56       external_date              = us_ketdat
57       external_period             = us_kprgbz
58     importing
59       internal_date              = vbak-vdatu
60       ev_date_in_past            = lv_date_in_past
61       internal_period             = vbak-vprgr.
```

Scope: \FORM vbak-vdatu\_pruefen

| ABAP | Ln 50 Col 1



*ABAP Editor: Display Include FV45KF0V\_VBAK-VDATU\_PRUEFEN*



```
Include FV45KF0V_VBAK-VDATU_PRUEFEN Active

37          t180-trtyp ne chara and
38          checked_vdatu is initial ).
39
40
41 if ( t180-trtyp = charv and checked_vdatu is initial
42     da_warning = space.
43 else.
44     da_warning = 'X'.
45 endif.
46
47 clear: checked vdatu.
```

```
49 *---- check the requested delivery date ----
50 |
51 * 1. check if the requested delivery date is in the past
52 *      (and convert it to the correct format (screen/external into req.del.date))
53 call 'SAPL_DC_DATE_IN_PAST' 'X_DC_DATE_IN_PAST_X'
54   expecting
55     da_logon           = us_da_logon
56     external_date       = us_external_date
57     external_period     = us_external_period
58     internal_date       = us_internal_date
59     internal_period     = us_internal_period
60     imp_starting         = us_imp_starting
61     imp_stopping        = us_imp_stopping
62     k-vdatu             = us_k-vdatu
63     date_in_past        = us_date_in_past
64     k-vprqr              = us_k-vprqr.
```



Scope: (F) M vbak-vdatu\_pruefen

| ABAP | Ln 50 Col

# ABAP Evolves

... ← 3.0 ← ... ← 7.0 ← ... ← 7.4 ← 7.5 ← ...  
..... →

Backward Compatible



```
" before 7.4 release
```

```
DATA flight_schedule TYPE spfli.
```

```
READ TABLE flight_schedules INTO flight_schedule
```

```
WITH KEY carrid = 'AA'
```

```
connid = '0017'.
```

```
" after 7.4 release
```

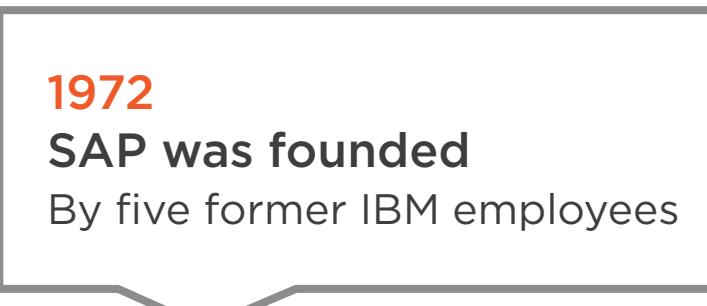
```
DATA(flight_schedule) = flight_schedules[ carrid = 'AA' connid = '0017' ].
```

# ABAP is Backward Compatible

I will demonstrate the examples in 7.4



# ABAP Objects



# Procedural ABAP

```
DATA schedules TYPE TABLE OF spfli.
```

```
SELECT * FROM spfli  
INTO TABLE schedules  
UP TO 100 ROWS.
```

```
SORT schedules BY carrid connid.
```

```
LOOP AT schedules INTO schedule.
```

```
...
```

```
ENDLOOP.
```

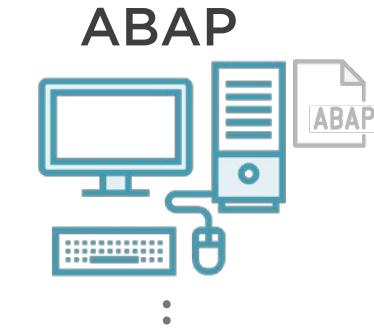
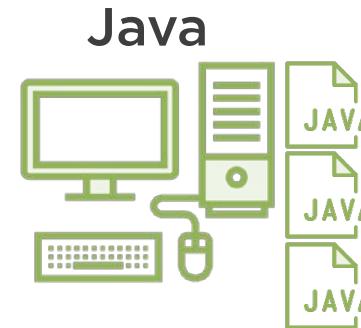
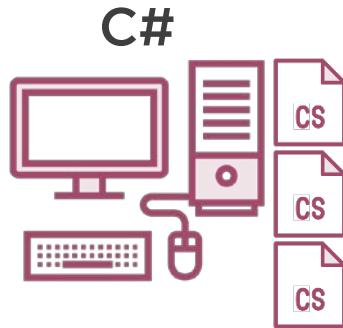
# OO ABAP

```
CLASS zcl_simple_alv DEFINITION.  
  PUBLIC SECTION.  
    METHODS constructor.  
    METHODS initialize_alv.  
  ...  
ENDCLASS.
```

```
CLASS zcl_simple_alv IMPLEMENTATION.  
  ...  
  METHOD initialize_alv.  
    cl_salv_table=>factory(  
      IMPORTING  
        r_salv_table = _alv  
      CHANGING  
        t_table      = schedules).  
    ENDMETHOD.  
  ...  
ENDCLASS.
```



# Developing on the Application Server



# Cross-Platform ABAP Development

## ABAP

```
SELECT * FROM spfli INTO TABLE schedules  
UP TO 100 ROWS.
```

```
LOOP AT schedules INTO schedule.
```

```
...
```

```
ENDLOOP.
```

High-level abstraction

## Database

Oracle  
MS-SQL  
IBM DB2  
MAXDB

## Operating System

Windows  
Linux  
IBM AIX  
Oracle Solaris



# Cross-Platform ABAP

Next

## ABAP Syntax

### ABAP

```
SELECT * FROM spfli INTO TABLE schedules  
UP TO 100 ROWS.
```

```
LOOP AT schedules INTO schedule.
```

```
...
```

```
ENDLOOP.
```

High-level abstraction

### Database

Oracle  
MS-SQL  
IBM DB2  
MAXDB

### Operating System

Windows  
Linux  
IBM AIX  
Oracle Solaris



# ABAP Syntax

---



# ABAP Syntax

## ◀ ABAP statements

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

◀ ABAP statements

◀ Keyword + Operand + Operator

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction
```

```
    day_of_the_week = day_of_the_week - 1.
```

```
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.
```

```
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

◀ ABAP statements

◀ Keyword + Operand + Operator

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction
```

```
    day_of_the_week = day_of_the_week - 1.
```

```
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.
```

```
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

◀ ABAP statements

◀ Keyword + Operand + Operator

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto   = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

◀ ABAP statements

◀ Keyword + Operand + Operator

◀ Separated by **spaces**

```
DATA(day_of_the_week)•=•selected_date•MOD•7.
```

```
IF•day_of_the_week•>•1." Correction  
    day_of_the_week•=•day_of_the_week•-•1.  
ENDIF.
```

```
flight_schedule-cityfrom•=•'New York'.  
flight_schedule-cityto••=•'Budapest'.
```

\* Simplifies the usage of CL\_SALV\_TABLE

```
CLASS•zcl_simple_alv•DEFINITION.
```

...

```
ENDCLASS.
```

```
simple_alv->display(•).
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is **not** case sensitive

~~dayOfTheWeek~~ → Camel Case

~~DayOfTheWeek~~ → Pascal Case

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.
```

...

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is **not** case sensitive

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE
```

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows **30 characters**

<del>value</del>	→ meaningless
<del>day_week</del>	→ still not enough
<del>day_of_the_week</del>	→ meaningful

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

\* Simplifies the usage of CL\_SALV\_TABLE

```
CLASS zcl_simple_alv DEFINITION.
```

```
...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
  - \_day\_of\_the\_week
  - 2nd\_day\_of\_the\_week
  - day\_of\_2nd\_week

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
...  
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
- ◀ Comment: full line

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

\* Simplifies the usage of CL\_SALV\_TABLE

```
CLASS zcl_simple_alv DEFINITION.  
  ...
```

```
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
- ◀ Comment: full line
- ◀ Comment: partial line

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
  ...  
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

- ◀ ABAP statements
- ◀ Keyword + Operand + Operator
- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
- ◀ Comment: full line
- ◀ Comment: partial line
- ◀ Back quotes for strings

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
  ...  
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
  ...  
ENDCLASS.
```

```
simple_alv->display().
```

- ◀ **Keyword + Operand + Operator**
- ◀ **Separated by spaces**
- ◀ **Close statements with dots**
- ◀ **Syntax is not case sensitive**
- ◀ **ABAP allows 30 characters**
- ◀ **[ a-z 0-9 \_ ]**
- ◀ **Comment: full line**
- ◀ **Comment: partial line**
- ◀ **Back quotes for strings**
- ◀ **Component selector: **structure****



# ABAP Syntax

- ◀ Separated by spaces
- ◀ Close statements with dots
- ◀ Syntax is not case sensitive
- ◀ ABAP allows 30 characters
- ◀ [ a-z 0-9 \_ ]
- ◀ Comment: full line
- ◀ Comment: partial line
- ◀ Back quotes for strings
- ◀ Component selector: structure
- ◀ Component selector: object

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
  ...  
ENDCLASS.
```

```
simple_alv->display().
```



# ABAP Syntax

◀ Separation

◀ Close statement

◀ Syntax

◀ ABAP allows 30 characters

◀ [ a-z 0-9 \_ ]

◀ Comment: full line

◀ Comment: partial line

◀ Back quotes for strings

◀ Component selector: structure

◀ Component selector: object

```
DATA(day_of_the_week) = selected_date MOD 7.
```

```
IF day_of_the_week > 1. " Correction  
  day_of_the_week = day_of_the_week - 1.  
ENDIF.
```

```
flight_schedule-cityfrom = 'New York'.  
flight_schedule-cityto  = 'Budapest'.
```

```
* Simplifies the usage of CL_SALV_TABLE  
CLASS zcl_simple_alv DEFINITION.  
  ...  
ENDCLASS.
```

```
simple_alv->display().
```

Next

# Summary



# Summary



## SAP ERP as a standard solution

- SAP Industry Solutions
- Customizing
- ABAP

## Primary programming language for SAP

- The basis of the entire system
- Except the kernel
- The source code is available to you
- ABAP is backward compatible
- Hybrid language
- Development on the server
- Cross-platform language

## ABAP syntax: basic language elements



Summ

Next Module

SAP ERP is a generic solution

SAP Industry Solutions

# ABAP: Behind the Scenes



- The basis of the entire system
- Except the kernel
- The source code is available to you
- ABAP is backward compatible
- Hybrid language
- Development on the server
- Cross-platform language

ABAP syntax: basic language elements

