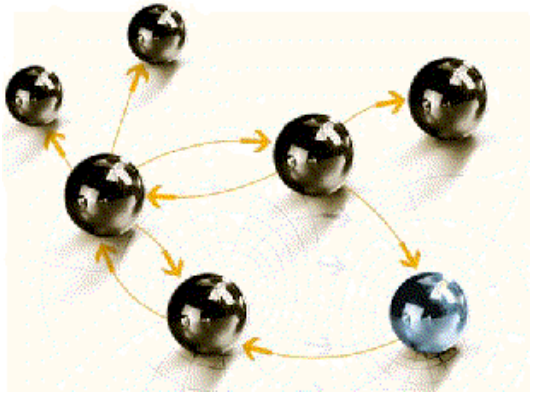


cps4it

COBOL

Dump Analyse im z/OS

cps4it
consulting, projektmanagement und seminare für die informationstechnologie
Ralf Seidler, Stromberger Straße 36A, 55411 Bingen
Fon: +49-6721-992611, Fax: +49-6721-992613, Mail: ralf.seidler@cps4it.de
Internet: http://www.cps4it.de

Ziele des Seminars

- L(anguage) E(nvironment) und seine Komponenten kennen lernen
- Steuerblöcke von COBOL und LE kennen und ihre Bedeutung erkennen
- COBOL-Compileliste verstehen
- Dump-Adressen und Inhalte von Variablen mit Hilfe des CEEDUMPs schnell und sicher erkennen

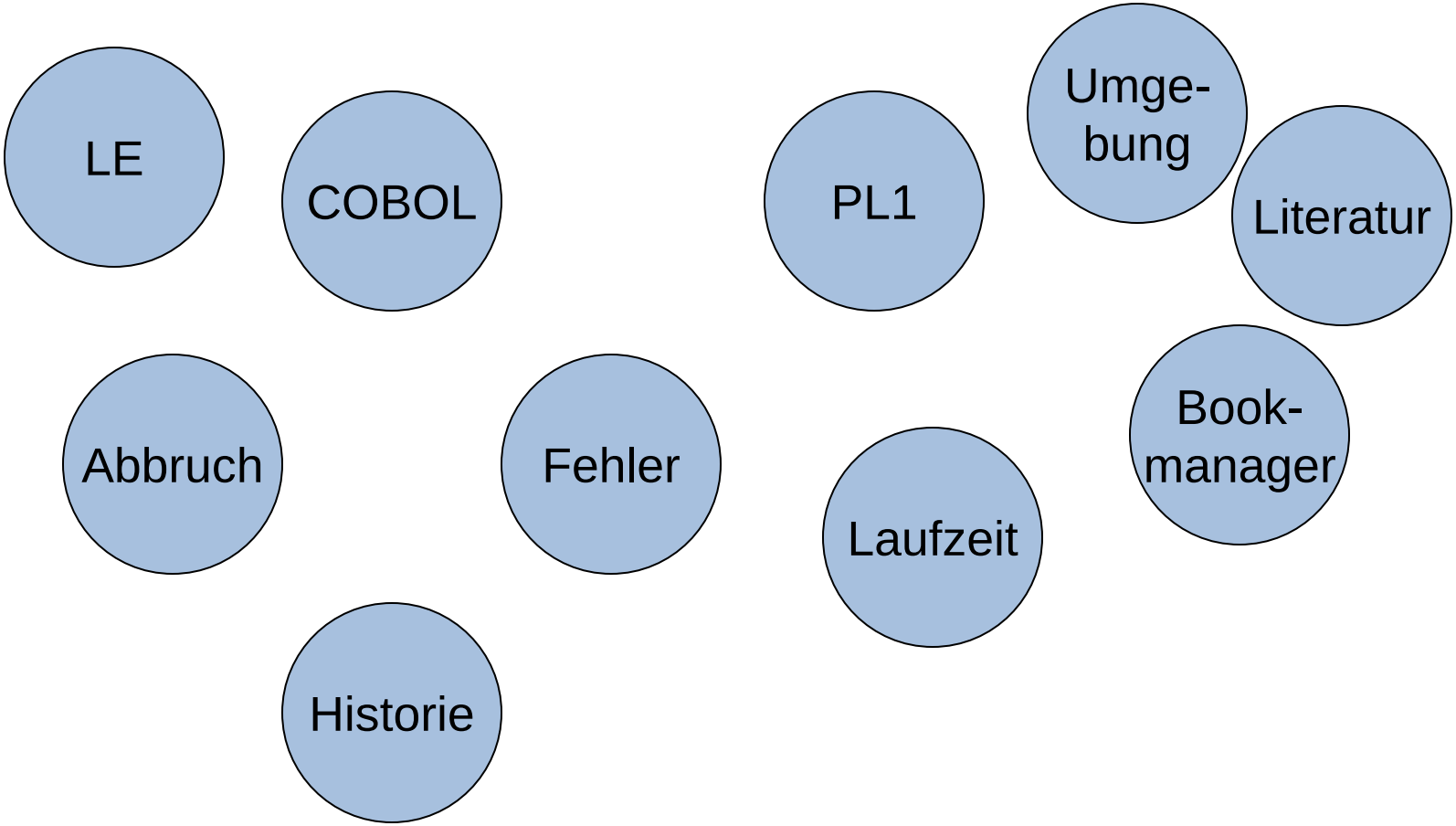
Inhaltsverzeichnis

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

Einführung

Begriffe



Einführung

LE – Historie

- 80-er Jahre kein ILC möglich COBOL-PL1
  - PLI sorgte für den Support: (COBOL kümmerte sich nicht darum!)
- Herausforderung: Investitionssicherheit
- Konzept in Zusammenarbeit mit Kunden
- Ergebnis LE (Language Environment) als / mit
  - Laufzeitumgebung für mehrere Sprachen
  - Routinen liegen 1 Mal optimal vor
  - Ziel Investitionssicherheit erreicht
  - und ...

Einführung

LE – Eigenschaften

- Technische Eigenschaften
  - gemeinsame Initialisierung
  - gemeinsames Beenden
  - gemeinsame Speichernutzung
  - gemeinsames Fehlerhandling
  - Unterstützung allgemeines Debugging Tool
  - gemeinsame Sprachphilosophie
- aktuelle Version für/unter z/OS 2.4

Einführung

LE – Literatur

- Bookmanager im Intranet
- Bookmanager auf IBM-Seiten im Web
- keine(?) Literatur auf dem Markt
- LE-Seite der IBM
  http://www.ibm.com/servers/eserver/zseries/zos/le/
- verschiedene Share-Vorträge z.B.
  - Allgemeine Informationen über Abend im LE:
    ~~http://www-03.ibm.com/servers/eserver/zseries/zos/le/conference/pdf/swa8208.pdf~~
  - Abbruch im Bereich Heap oder Stack
    ftp://ftp.software.ibm.com/eserver/zseries/zos/le/an8209.pdf

Einführung

DUMP – was ist das?

- to dump:
  abladen; schütten; auskippen; fallen lassen;
  abziehen; lagern; stapeln; verklappen;
- the dump
  Abzug; Dump; Auflistung; Depot; Kaff; Dreckloch;
  Sauladen; Schutthaufen, Abfallhaufen;
- to dump s.b.
  jdm. abschieben, jdm. loswerden
- to dump memory
  Speicherinhalt anzeigen

Einführung

DUMP – Haltung bei einem Dump

Beispiel

Einführung

Übung(en)

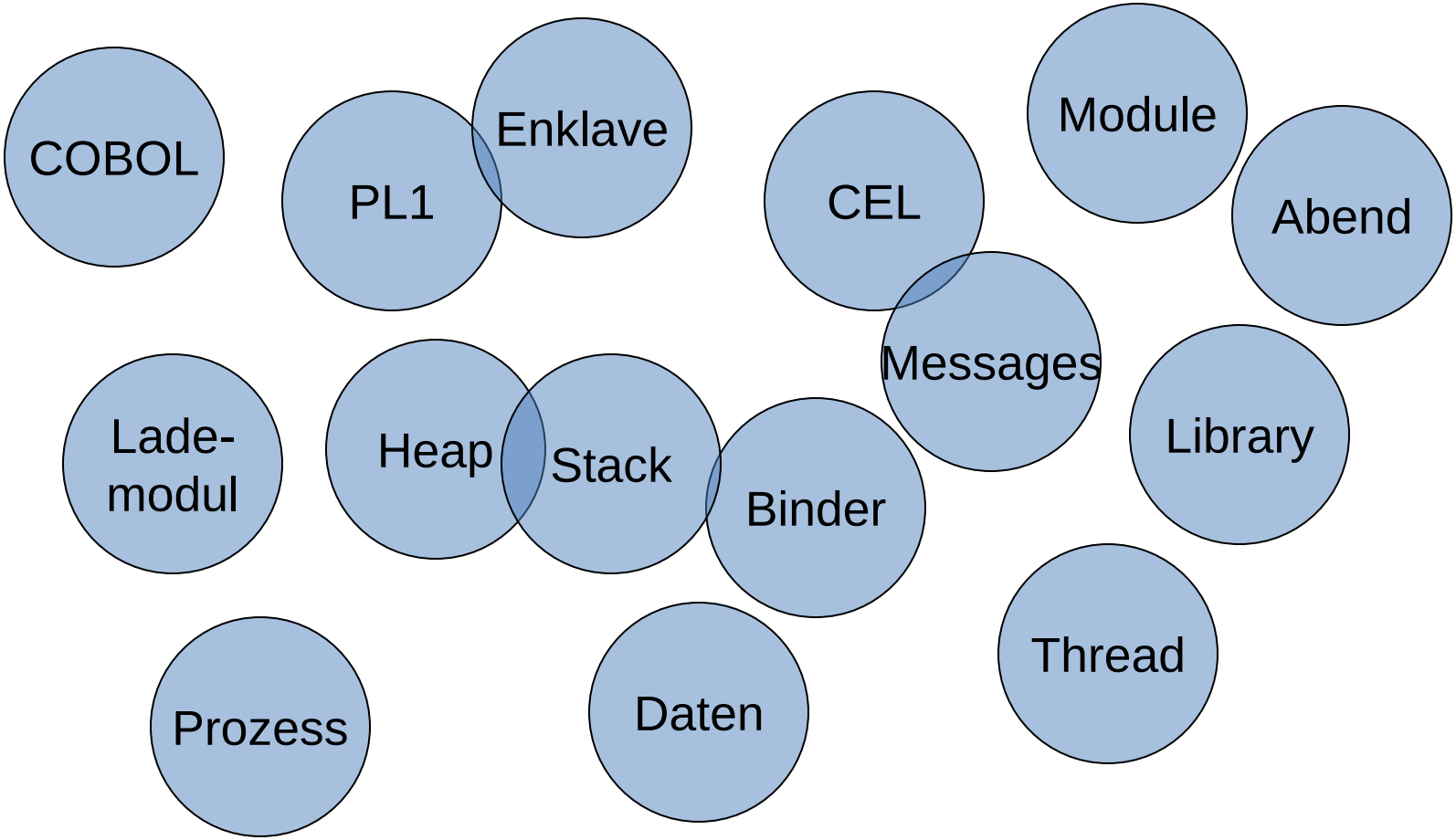
- 01-01 – Anmelden am System
- 01-02 – Zugang zu Bookmanger Intranet
- 01-03 – Zugang zu Bookmanager Internet
- 01-04 – wichtige Bücher in COBOL
- 01-05 – wichtige Bücher in LE
- 01-06 – Shelf Messages and Codes

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

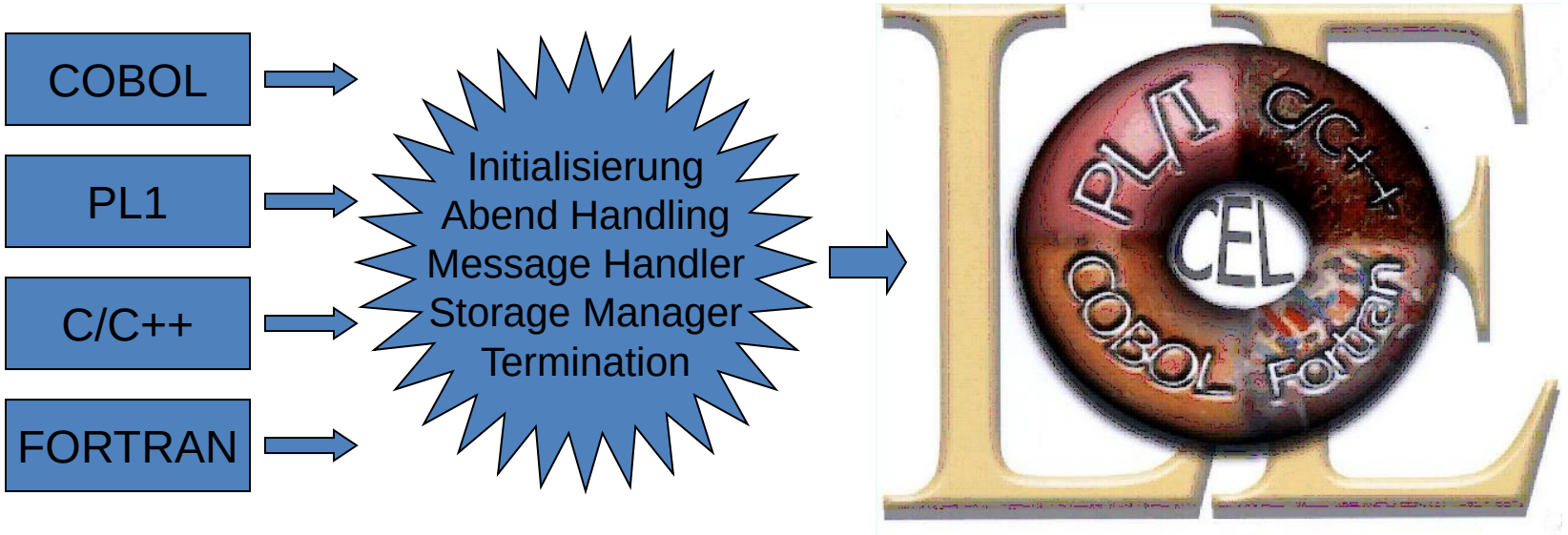
LE – Program Management

Begriffe



LE – Program Management

Philosophie

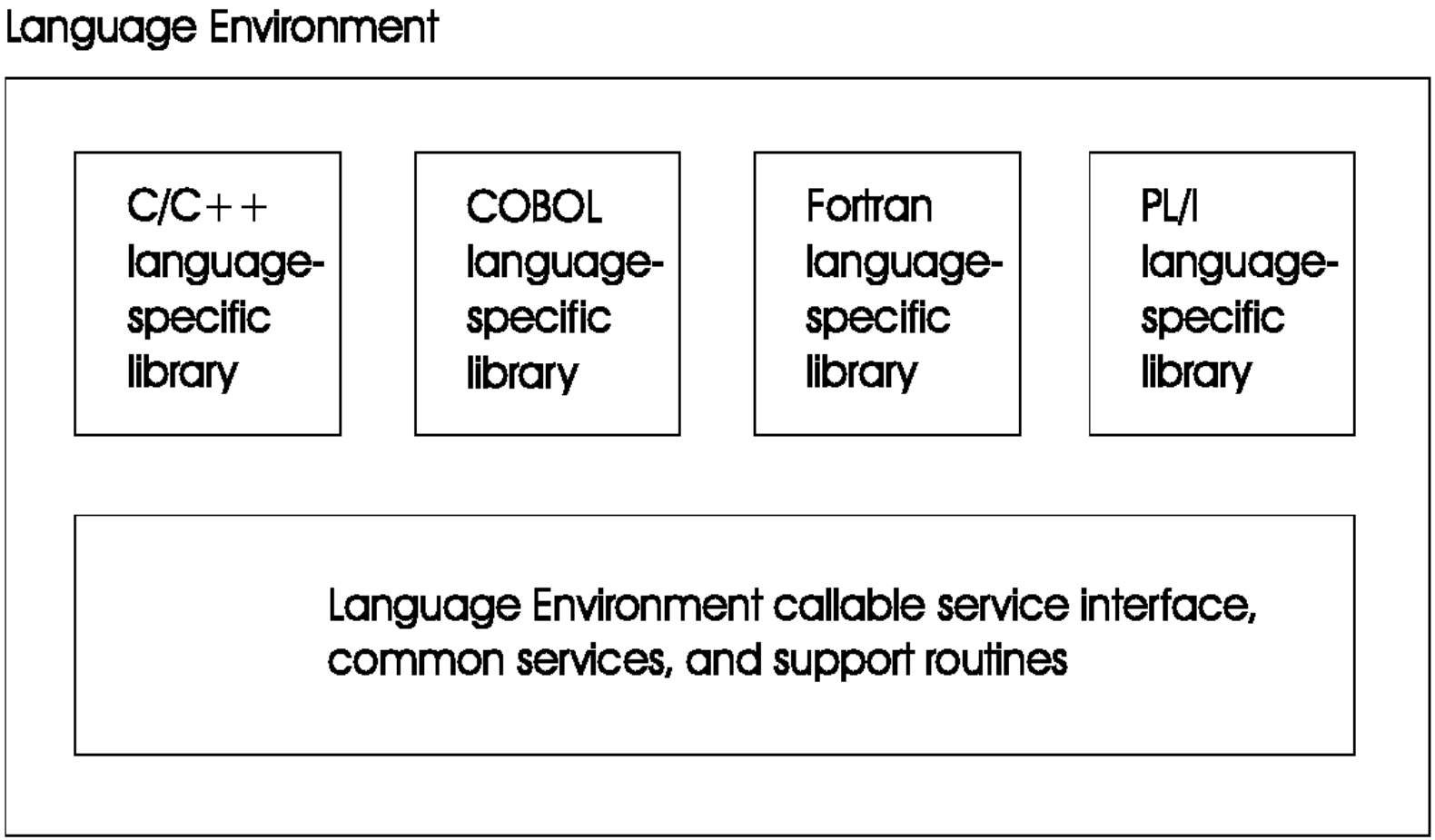


- 4 unabhängige Produkte
- jede Technik zu codieren
- ohne richtige Zukunft

- 1 Produkt
- kompatibel zu Technik
- zukunftsorientiert

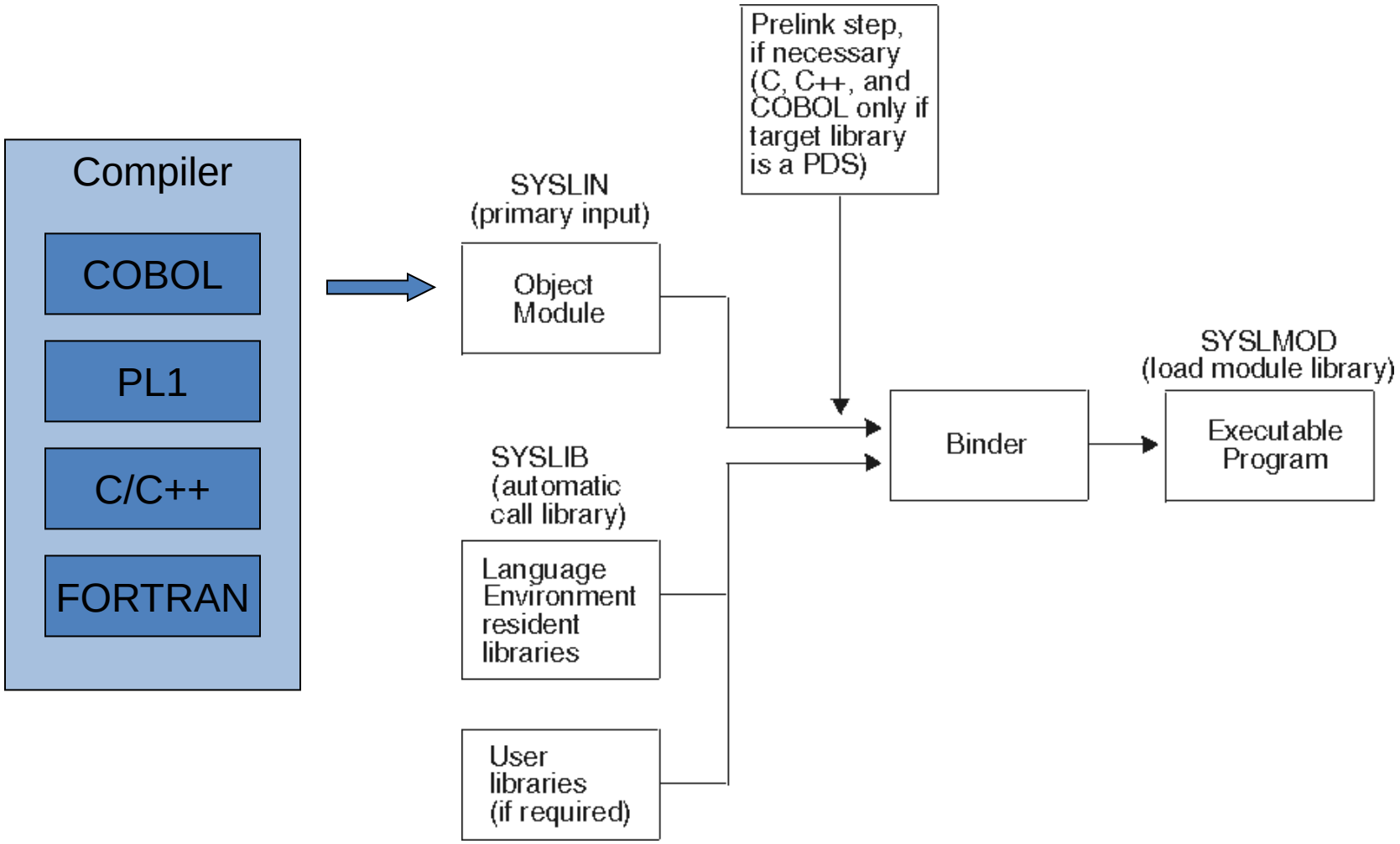
LE – Program Management

Komponenten



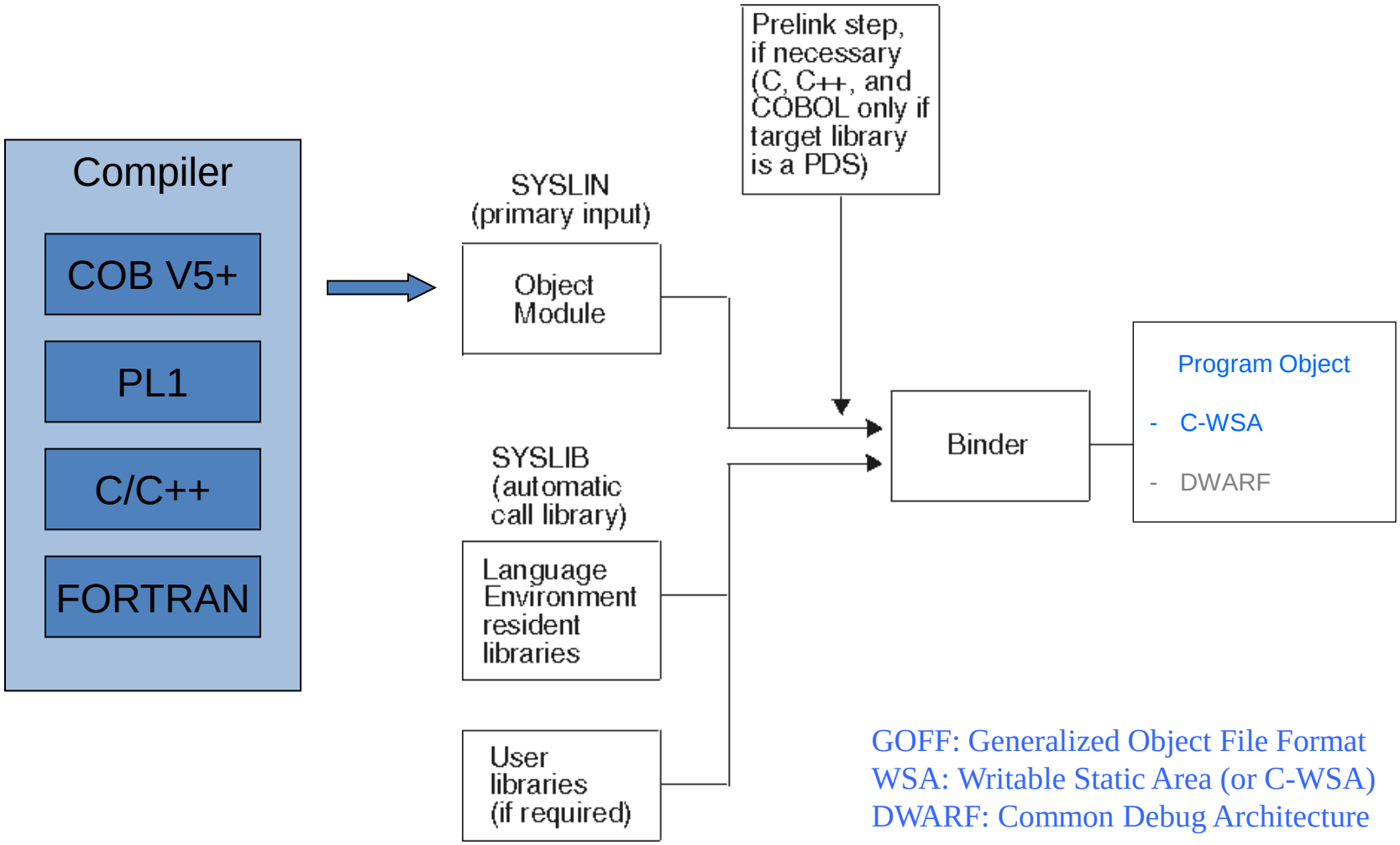
LE – Program Management

der Weg zum Lademodul, so war es mal



LE – Program Management

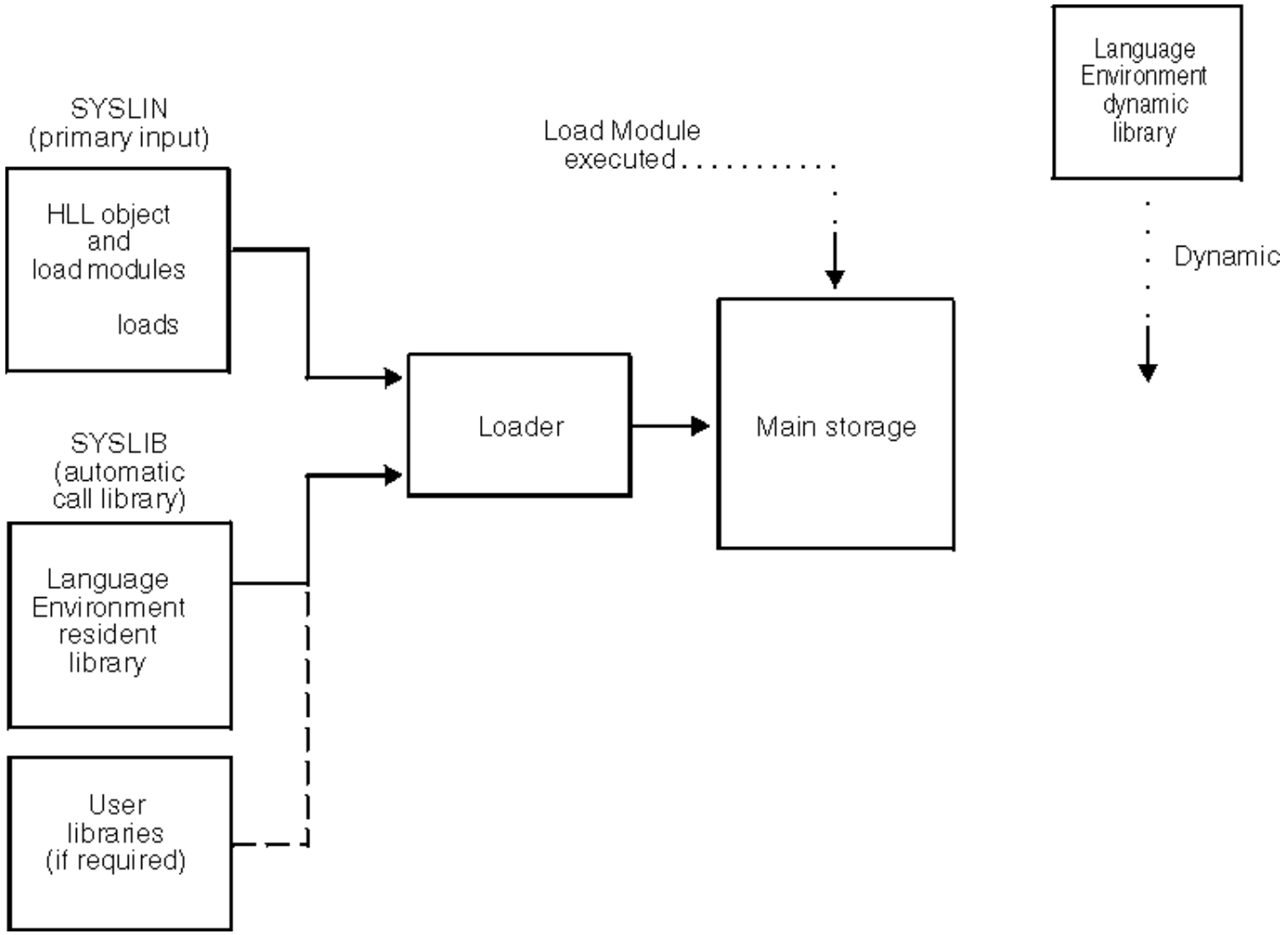
der Weg zum Program Object



GOFF: Generalized Object File Format
WSA: Writable Static Area (or C-WSA)
DWARF: Common Debug Architecture

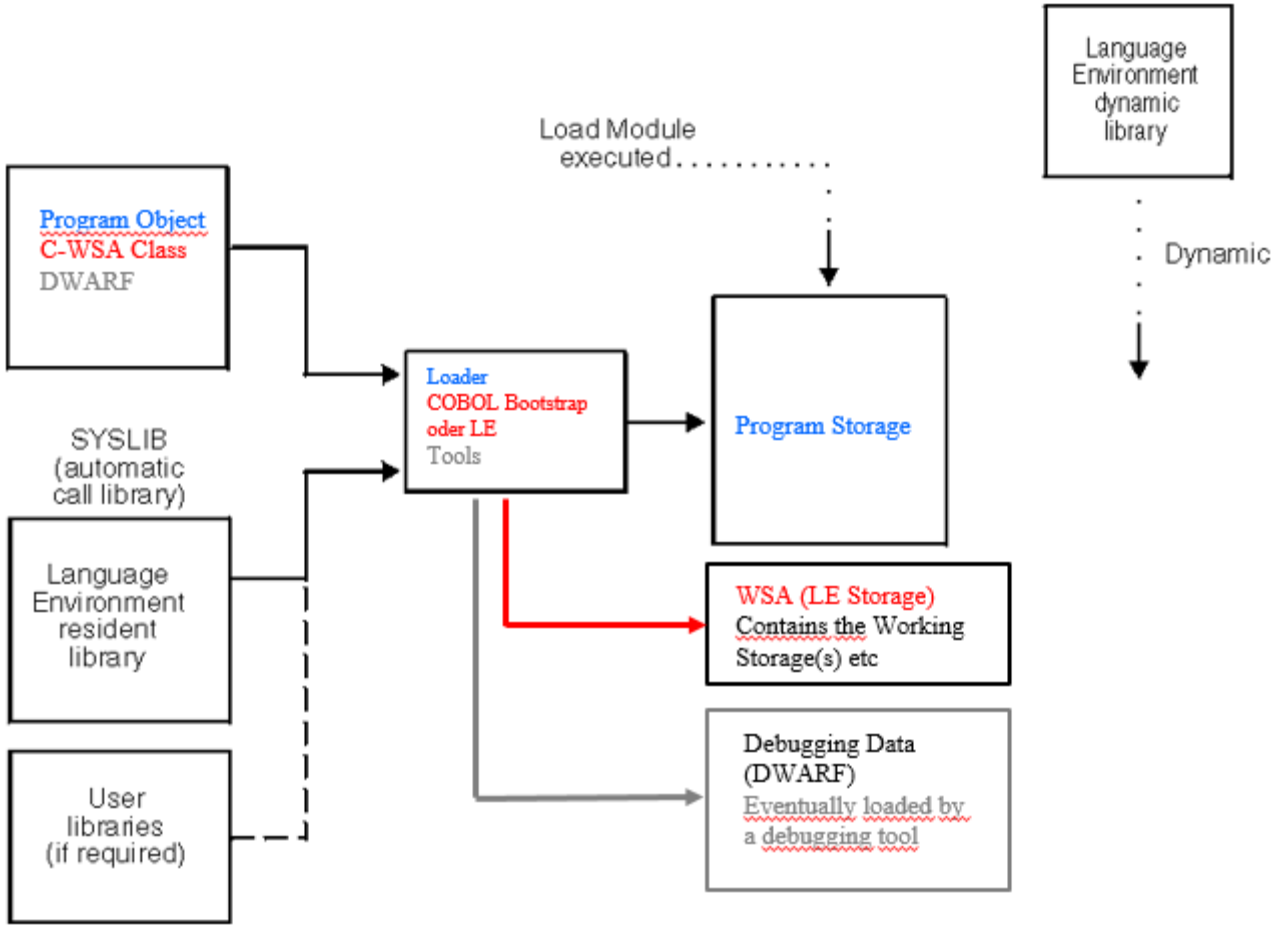
LE – Program Management

Laden des Programms – bisher



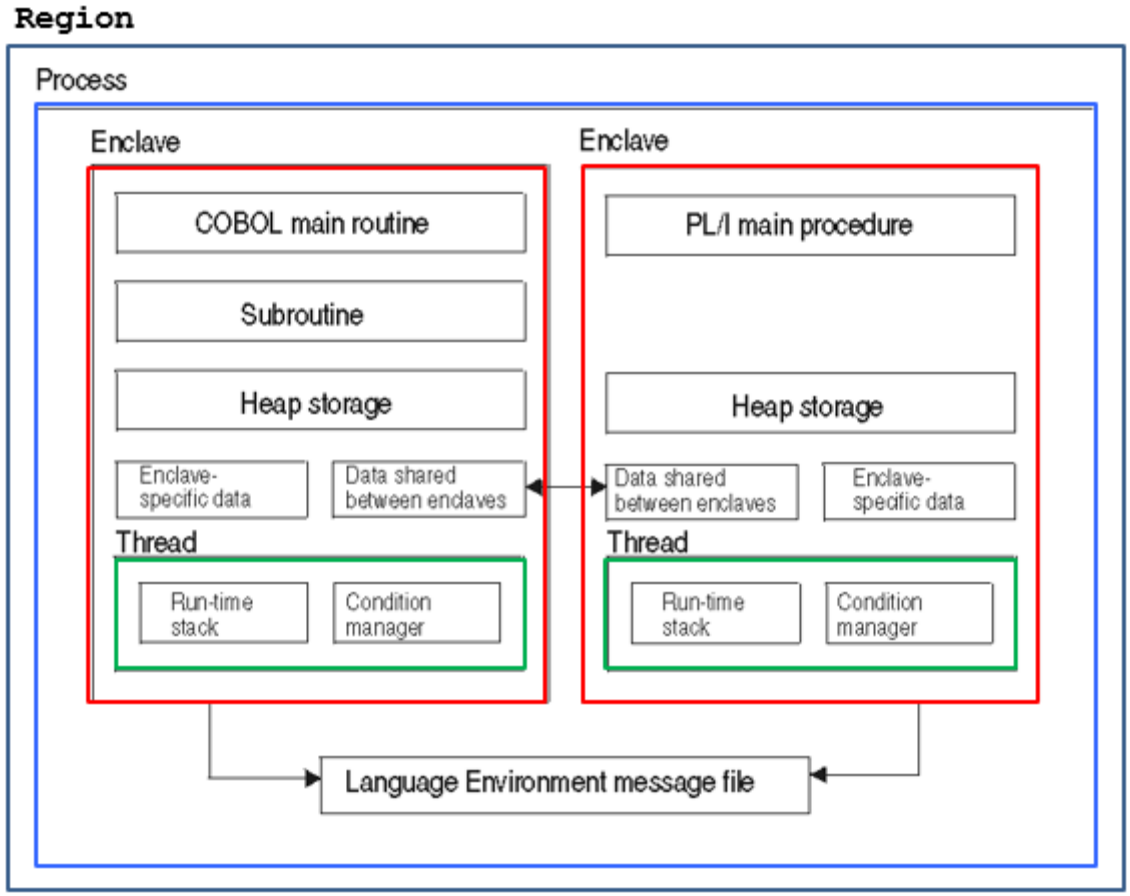
LE – Program Management

Laden des Program Objects (Programms)



LE – Program Management

Programm Management



Details

LE – Program Management

Terminologie – Programme

- Anwendungsprogramm
  - Ein Programm oder eine Sammlung von mehreren Programmen, die eine bestimmte Aufgabe erfüllen.
- Environment
  - Innerhalb von LE normalerweise die Referenz zur Laufzeitumgebung.
    - Region,
    - Process,
    - Enclave, eine oder mehrere (z.B. CICS) / Process
    - Thread, eine oder mehrere (z.B. PLI, C, POSIX) / Enclave

LE – Program Management

Terminologie – LE ↔ Sprache (1)

- Routine : Prozedur, Routine, Funktion
  - COBOL: Programm
  - PL1: procedure, begin-Block
  - C/C++: function
- Enclave: Routine(n) mit 1 Hauptprogramm
  - COBOL: Run-unit
  - PL1: main mit subroutines
  - C/C++: main function mit subfunctions
  - FORTRAN: program mit subroutines

LE – Program Management

Terminologie – LE ↔ Sprache (2)

- Region:
  - Beinhaltet LE Laufzeit Module und deren Adressierung
- Process:
  - die oberste Hierarchie im LE-Programm-Management. Er beinhaltet u.a. Synchronisierung von Standard input/output. Enthält zumindest eine Enclave
- Enclave:
  - die oberste Hierarchie im Program Management. Er beinhaltet ein Main Program und managed den HEAP storage
- Thread:
  - Ein Konstrukt zur Ausführungszeit, das synchrone Aufrufe und Beendigungen von Routinen beinhaltet. Der Thread ist der “Anfang” im LE-Modell und wird durch das System geladen mit eigenem Stack und PSW sowie eigenen Registern. Es kann mehrere Threads parallel geben, Sehr häufig in POSIX verwendet (z.B. Java)

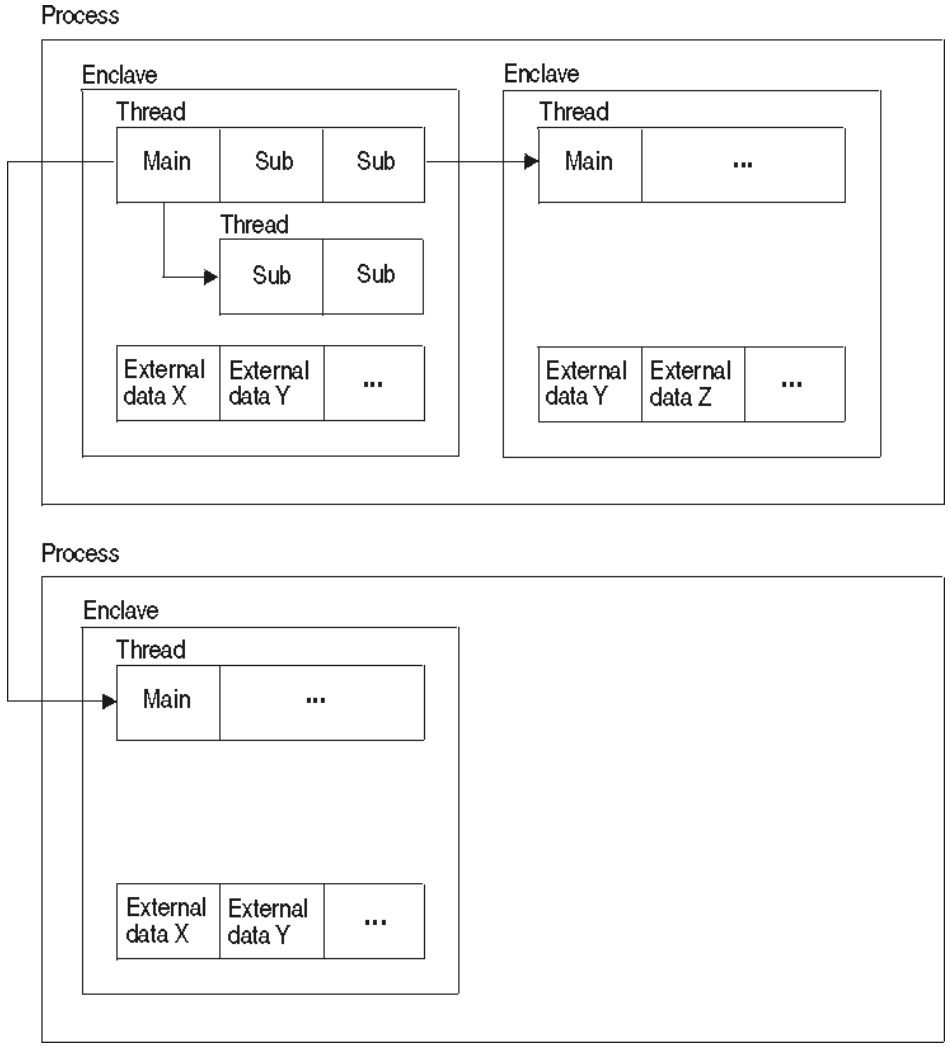
LE – Program Management

Terminologie – LE ↔ Daten

- Automatic Data:
  - Die Daten gehören zu einer Routine und sind nicht persistent. Sie werden zu Beginn einer Routine immer mit den gleichen Werten bereit gestellt (Beispiel: "Local Storage" in DSA)
- External Data:
  - Die Daten können durch mehrere Routinen benutzt werden. Sie sind innerhalb einer Enclave jederzeit bekannt.
- Local Data:
  - Die Daten sind nur innerhalb einer Routine bekannt.
  - Die Daten sind Persistent (Beispiel: "Working Storage" in WSA)

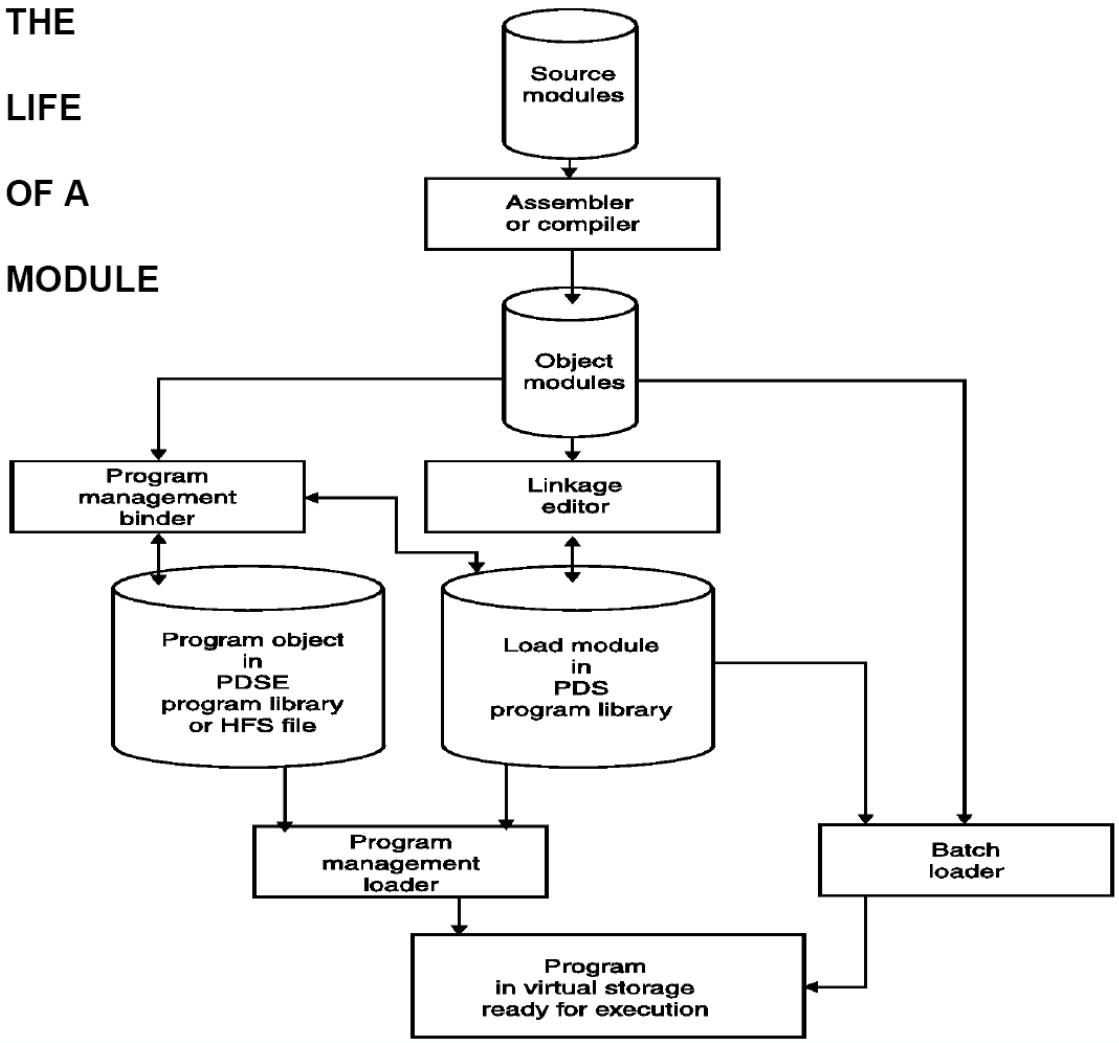
LE – Program Management

Terminologie



LE – Program Management

Das Leben eines Moduls

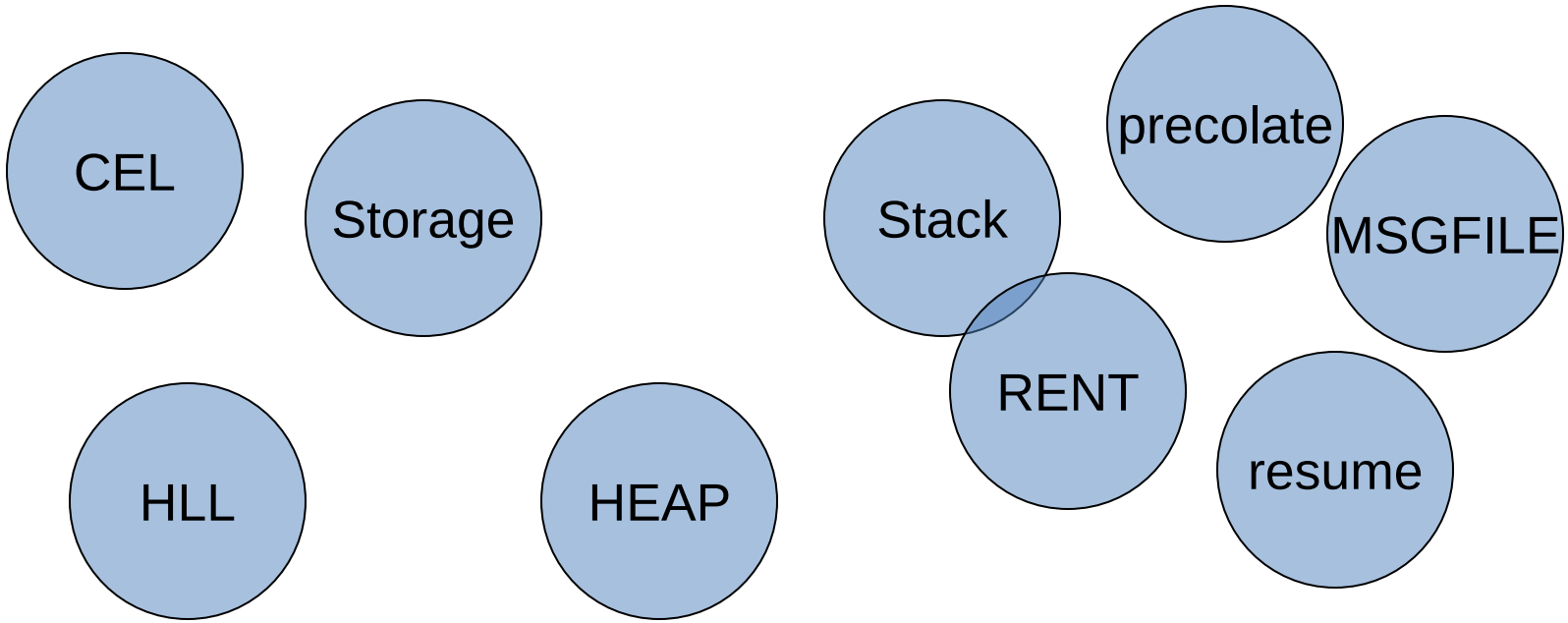


Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

LE – Condition Handler

Begriffe



LE – Condition Handler

CEL – Common Execution Library – Funktionen

- Initialisierung
- Storage Management
- Condition Handling
- Message Services
- Date/Time Services
- Math Functions
- Termination
- alles durch Options beeinflussbar

COBOL-Beispiel

COBOL-Beispiel

LE – Condition Handler

CEL – Common Execution Library – Initialisierung

- Code, der zum Programm gelinkt wurde, beginnt einen Bootstrap-Prozess zum Initialisieren von LE
  - Initial Storage besorgen
  - Initialisieren Condition Handler
  - sprachspezifische Laufzeitumgebung initialisieren
- Kontrolle an Anwendungsprogramm

LE – Condition Handler

CEL – Common Execution Library – Storage Manager

- HLL Run-time oder CEL ruft auf
  - HEAP-Storage
    - COBOL V4 Working Storage, C malloc, PL1 allocate
    - COBOL V5+ WSA
  - STACK-Storage (Thread)
    - verwendet für Linkage (save area), Automatische Variablen und mehr! In LE auch (z.B. im CEEDUMP DSA's genannt)
    - C/C++,
    - PL1 automatic variables,
    - COBOL Local-Storage-Section

Note: "DSA" = Dynamic Save Area

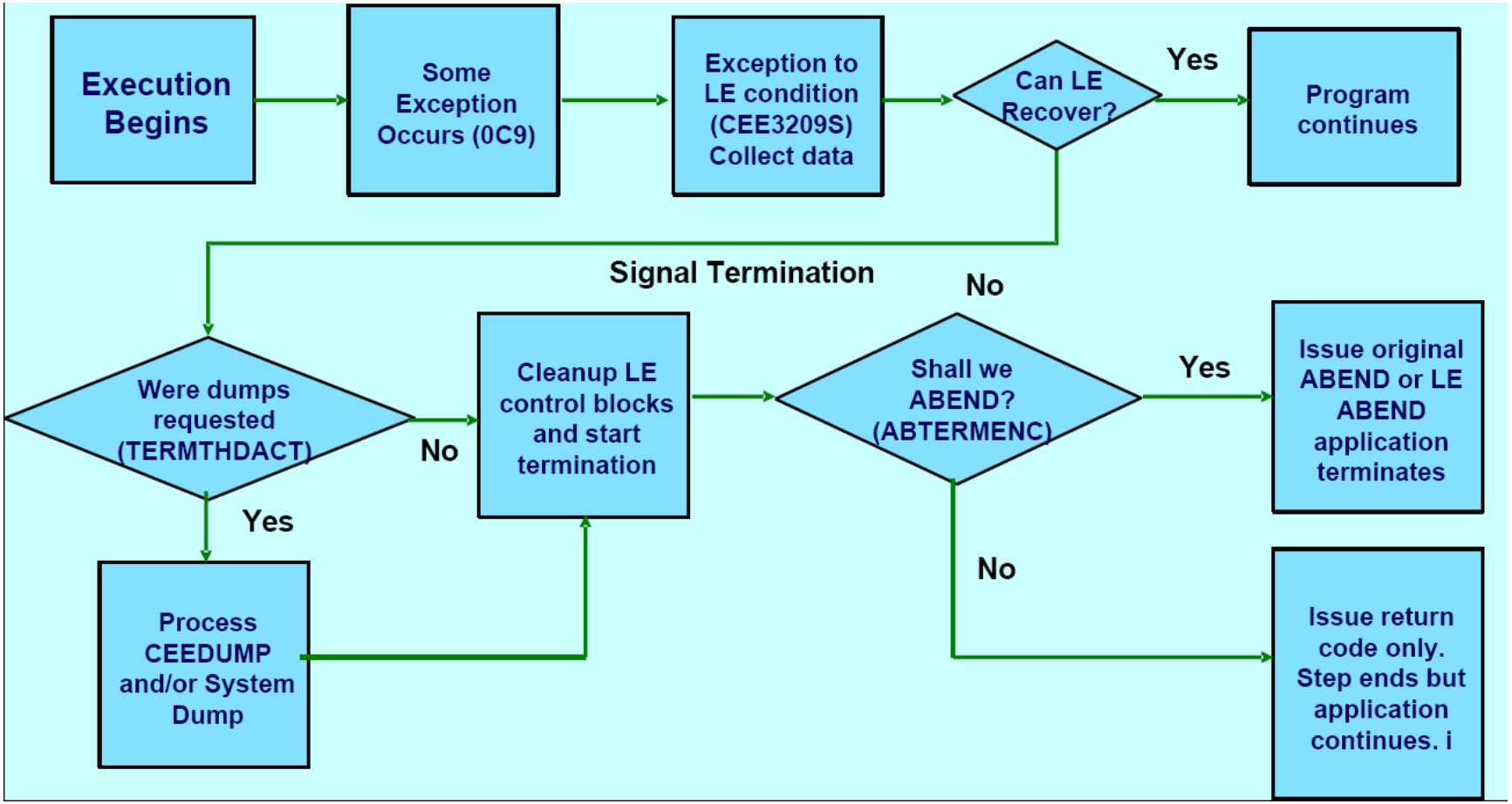
LE – Condition Handler

CEL – Common Execution Library – Condition Handler

- Conditions werden abgefangen
  - *handled* – PL1 on units, C/C++ signal/raise, LE User Handler (letzteren können auch von COBOL verwendet werden
  - *unhandled* – hardware abend / software abend
- Steuerung geht an condition handler (LE/USER)
- mögliche Aktionen
  - *resume* – Kontrolle geht an einen “resume cursor”
  - *percolate* – condition handling wird abgelehnt
  - *promote* – Bedeutung der condition verändern
  - *fix-up and resume* – Korrektur und weiter (nur LE)

LE – Condition Handler

Ablauf



LE – Condition Handler

Informationsausgabe

- messages
- CEEDUMP
- system Dump (SYSUDUMP, SYSMDUMP…)
- Run-time Options Report
- Run-time Storage Report

LE – Condition Handler

CEL – Common Execution Library – Message Services

- Möglichkeit für allgemeine Meldungen
- Meldungen werden an 1 Ort geschrieben
  - MSGFILE
- irgendwelche “komischen” Abbrüche können formatiert werden
- Unterdrückung von Dumps bei bestimmten Arten von Abbrüchen ist möglich
- und so weiter

LE – Condition Handler

CEL – Common Execution Library – Termination

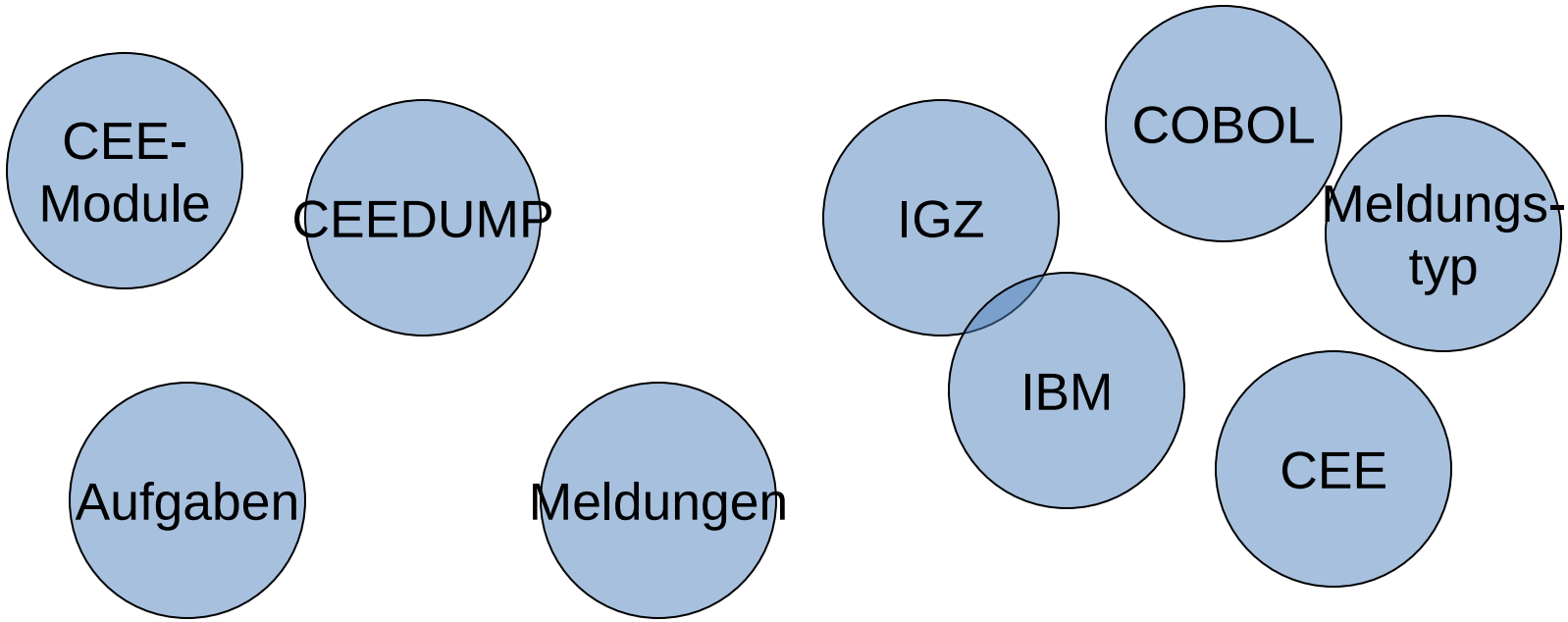
- Nach Ende der Anwendung ist die LE-Umgebung “weg”.
- vollständiges garbage collection von allen Ressourcen
  - bei Initialisierung
  - während Laufzeit
- Bedingung: Es darf nur mit “sauberen” LE-Mitteln gearbeitet werden.
  - Hinweis: CICS benutzt „saubere“ Mittel.

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

LE – Abbruchinformationen

Begriffe



LE – Abbruchinformationen

wichtige Laufzeit-Module (1)

- CEEHDSP
  - immer die TOP CSECT in CEE-Dumps
  - Aufgaben
    - Fehleranalyse und Behandlung, Fehler Infos sammeln,
    - Dumps und Fehlernachrichten aufbereiten / schreiben
    - Eventuell Fehler ignorieren und Dumps unterdrücken
  - ignorieren bei Analyse
- CEEPLPKA
  - Hauptprogramm des LE
  - beinhaltet u.a. CEEHDSP
  - Wenn hier ein Abbruch auftritt, kann das im LE sein, aber auch in der Anwendung.

LE – Abbruchinformationen

wichtige Laufzeit-Module (2)

- CEEBINIT
  - Aufgabe: Initialisierung
  - Wenn hier ein Abbruch auftritt, hat das was mit dem Setup der Umgebung zu tun.
- CEECCICS
  - Aufgabe: Initialisierung im CICS
  - Wenn hier ein Abbruch auftritt, hat das was mit dem Setup der Umgebung zu tun.

LE – Abbruchinformationen

wichtige Laufzeit-Module (3)

- CEEHSGLT
  - Aufgabe: signal handler
  - Zeigt den Verursacher an.
- CEEV#GH/CEEV#FH
  - Aufgabe: Get/Free Heap-Speicher
  - Wenn hier ein Abbruch auftritt, ist wahrscheinlich der Heap-Speicher defekt.

LE – Abbruchinformationen

wichtige Laufzeit-Module (4)

- CEEEVxxx
  - Aufgabe: Event Handler
  - xxx zeigt die Sprache an

    003 – C/C++
    004 – COBOL V5+
    005 – COBOL
    007 – FORTRAN
    010 – PL1
    011 – VA PL1
    012 – Debug Tool

LE – Abbruchinformationen

Meldungen – Verursacher

- CEE CEL (aber könnte woanders hin zeigen)
- IGZ COBOL
- IBM PL1
- AFH FORTRAN
- EDC C/C++

- Details zu COBOL siehe zum Beispiel:
  z/OS V2R1.0 Language Environment Run-Time Messages,
  Kapitel 7.0 COBOL Run-Time Messages

LE – Abbruchinformationen

Meldungen – Aufbau und Typen von Meldungen (COBOL)

- IGZnnnnx mit x=
  - I Informational message
  - W Warning message
  - E Error message
  - S Severe error message
  - C Critical error message

LE – Abbruchinformationen

Meldungen – Beispiel 1 (COBOL)

```
IGZ0006S The reference to table ??? by verb number ???
         on line ??? addressed an area outside
         the region of the table.
```

Explanation: When the SSRANGE option is in effect, this message is issued to indicate that a fixed-length table has been subscripted in a way that exceeds the defined size of the table, or, for variable-length tables, the maximum size of the table.

The range check was performed on the composite of the subscripts and resulted in an address outside the region of the table. For variable-length tables, the address is outside the region of the table defined when all OCCURS DEPENDING ON objects are at their maximum values; the ODO object's current value is not considered. The check was not performed on individual subscripts.

Programmer Response: Ensure that the value of literal subscripts and/or the value of variable subscripts as evaluated at run-time do not exceed the subscripted dimensions for subscripted data in the failing statement.

System Action: The application was terminated.

Symbolic Feedback Code: IGZ006

LE – Abbruchinformationen

Meldungen – Beispiel 2 (COBOL)

```
IGZ0011C ??? was not a proper module for this system
         environment.
```

Explanation: A library subroutine that is system sensitive is inappropriate for the current system environment. For example, an OS environment specific module has been loaded under CICS. The likely causes are:

- Improper concatenation sequence of partitioned data sets that contain the subroutine library, either during run-time or during link-edit of the COBPAC.
- An attempt to use a function unsupported on the current system (for example, ACCEPT on CICS).

Programmer Response: Check for the conditions stated above, and modify the environment or the application as needed.

System Action: The application was terminated.

Symbolic Feedback Code: IGZ00B

LE – Abbruchinformationen

Meldungen – Beispiel 3 (COBOL)

```
IGZ0100S Argument-1 for function ??? in program ??? at
         displacement ??? was less than or equal to -1.
```

Explanation: An illegal value was used for Argument-1.

Programmer Response: Ensure that argument-1 is greater than -1.

System Action: The application was terminated.

Symbolic Feedback Code: IGZ034

LE – Abbruchinformationen

Meldungen – Beispiel 4 (COBOL)

```
IGZ0017S The open of DISPLAY or ACCEPT file with
         environment name ??? was unsuccessful.
```

Explanation: An error occurred while opening the DISPLAY/ACCEPT file.

Programmer Response: Check to make sure a ddname has been defined for the file.

System Action: The application was terminated.

Symbolic Feedback Code: IGZ00H

LE – Abbruchinformationen

ein einfaches Beispiel – Hauptprogramm

```
CBL NOLIB,APOST,NODYNAM,NOOPT,TEST
    PROCESS QUOTE,MAP

    IDENTIFICATION DIVISION.
    PROGRAM-ID. COBOLED1.
    ENVIRONMENT DIVISION.
    DATA DIVISION.
    WORKING-STORAGE SECTION.
    01  WS-VARS.
        05 WS-COMP1 PIC S9(4) BINARY VALUE ZEROES.
    PROCEDURE DIVISION.
        CALL "COBOLED2".
        STOP RUN.
    END PROGRAM COBOLED1.
```

LE – Abbruchinformationen

ein einfaches Beispiel – Unterprogramm

```
 IDENTIFICATION DIVISION.
    PROGRAM-ID. COBOLED2.
...
    01  WS-VARS.
        05 WS-COMP1 PIC S9(4) BINARY VALUE ZEROES.
        05 WS-COMP2 PIC S9(4) BINARY VALUE ZEROES.
        05 WS-COMP3 PIC S9(4) BINARY VALUE ZEROES.
    PROCEDURE DIVISION.
        MOVE 32 TO WS-COMP3.
        MOVE 10 TO WS-COMP1.
        DIVIDE WS-COMP1 BY WS-COMP2 GIVING WS-COMP3.
        STOP RUN.
    END PROGRAM COBOLED2.
```

LE – Abbruchinformationen

ein einfaches Beispiel – Auszug aus dem Joblog

- Ein CEEDUMP ist formatiert und kann mit einfachem Browse angezeigt werden
  - ISPF browse
  - USS obrowse
  - CICS Teil vom CICS Transaction Dump oder CESE Queue
    - » **LE Runtime Option "TERMTHDACT (CICSDDS | CESE) "**
  - Der CEEDUMP ist Teil von Language Environment und somit kostenlos.
  - Nachteile?

```
CEE3209S The system detected a fixed-point divide exception.
         From compile unit COBOLED2 at entry point COBOLED2 at
         statement 13 at compile unit offset +00000308 at
         address 23E029E0.
```

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (1)

- Call-Hierarchie
  - von unten nach oben
  - mit Einstiegsadressen
  - für alle (offenen) Threads

```
1CEE3DMP V2 R3.0: Condition processing resulted in the unhandled condition.          06/23/19 1:47:26 PM   Page:    1
 ASID: 00E2   Job ID: JOB08948   Job name: DUMP0C71   Step name: P96NDP1    UserID: XV8822D

 CEE3845I CEEDUMP Processing started.

 Information for enclave P96NDP1

   Information for thread 8000000000000000

   Traceback:
     DSA    Entry       E  Offset  Statement   Load Mod            Program Unit                  Service  Status
     1      CEEHDSP     +00004A4C               CEEPLPKA            CEEHDSP                       HLE77B0  Call
     2      P96NDP2     +00000298  105          P96NDP2             P96NDP2                                Exception
     3      IGZXFCA1    +00002F58               IGZXLPKA            IGZXFCA1                               Call
     4      P96NDP1     +00000336  77           P96NDP1             P96NDP1                                Call

     DSA    DSA Addr    E  Addr     PU Addr     PU Offset  Comp Date  Compile Attributes
     1      000237C8    0CFAEB30    0CFAEB30    +00004A4C  20170307   CEL
     2      00023548    23B7F000    23B7F000    +00000298  20190623   COBOLV5+ EBCDIC  HFP
     3      00023298    23967540    23967540    +00002F58  20190211   LIBRARY
     4      00023030    23900000    23900000    +00000336  20190623   COBOLV5+ EBCDIC  HFP
```

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (2)

- Alle Informationen rund um den Abbruch
  - Fehlerart / Programm / PSW / Register / Speicher
- ED08 ED,R0,R8

```
Condition Information for Active Routines
    Condition Information for P96NDP2 (DSA address 00023548)
      CIB Address: 00023E98
      Current Condition:
        CEE0198S The termination of a thread was signaled due to an unhandled condition.
      Original Condition:
        CEE3207S The system detected a data exception (System Completion Code=0C7).
      Location:
        Program Unit: P96NDP2 Entry: P96NDP2 Statement: 105 Offset: +00000298
      Machine State:
        ILC..... 0006    Interruption Code..... 0007
        PSW..... 078D0000 A3B7F29E
        GPR0..... 00000000_00000026  GPR1..... 0000000C_00023688  GPR2..... 00000000_23B62200  GPR3..... 00000000_23B7F798
        GPR4..... 00000000_00023688  GPR5..... 00000000_23B7FF70  GPR6..... 00000000_23B6248E  GPR7..... 00000000_00000000
        GPR8..... 00000000_23B622B0  GPR9..... 00000000_23B623A0  GPR10.... 00000000_23B7F024  GPR11.... 00000000_00023170
        GPR12.... 00000000_00020C30  GPR13.... 00000000_00023548  GPR14.... 00000000_A3B7F294  GPR15.... 7F71E000_23B7FF70
        FPC...... 40000000
        FPR0..... 22080000  00000000           FPR1..... 22080000  00000000
        FPR2..... 00000000  0000004E           FPR3..... 00000000  00000002
        FPR4..... 22080000  00000000           FPR5..... 00000000  00000000
        FPR6..... 00000000  00000044           FPR7..... 00000000  00000000
        FPR8..... 00000000  00000000           FPR9..... 00000000  00000000
        FPR10.... 00000000  00000000           FPR11.... 00000000  00000000
        FPR12.... 00000000  00000000           FPR13.... 00000000  00000000
        FPR14.... 00000000  00000000           FPR15.... 00000000  00000000

    Storage dump near condition, beginning at location: 23B7F288
      +000000 23B7F288  90EE1814 50604018 18F50DEF 18F51814  ED082032 00AAED08 203C10AA B3D60011
```

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (3)

- Register für jede savearea auf Hierarchie
- Speicher rund um jedes Register

```
Parameters, Registers, and Variables for Active Routines:
    CEEHDSP (DSA address 000237C8):
      UPSTACK DSA
      Saved Registers:
        GPR0..... 00023B38  GPR1..... 00023848  GPR2..... 00023A34  GPR3..... 000238F0
        GPR4..... 0CFB4290  GPR5..... 00016038  GPR6..... 2390C4C0  GPR7..... 00023E98
        GPR8..... 8CFB3062  GPR9..... 000257C6  GPR10.... 000247C7  GPR11.... 0CFAEB30
        GPR12.... 00020C30  GPR13.... 000237C8  GPR14.... 800170EE  GPR15.... 8CF61A08
    GPREG STORAGE:
      Storage around GPR0 (00023B38)
        -0020 00023B18  00000000 00000000 8D073D48 00023378  0002390C 0001F828 80011198 00020C30  !.....................8....q....!
        +0000 00023B38  C3969584 89A38996 95409799 968385A2  A2899587 409985A2 A493A385 84408995  !Condition processing resulted in!
        +0020 00023B58  40A38885 40A49588 81958493 85844083  96958489 A3899695 4B404040 40404040  ! the unhandled condition.       !
      Storage around GPR1 (00023848)
        -0020 00023828  A3B80480 000237F0 00000001 8CFB357E  00000000 A3B7F000 A3B7F000 00000000  !t......0.......=....t.0.t.0.....!
        +0000 00023848  00023B38 00023C98 00023A34 00023A34  000238E4 00024CC0 000254C0 00000005  !.......q...........U..<.........!
        +0020 00023868  000300C6 59C3C5C5 00000000 00000000  00000000 00000001 000000CB 00CDCECF  !...F.CEE.......................!
```

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (4)

- Options Report

```
Run-Time Options Report:

   LAST WHERE SET                  OPTION
   ------------------------------------------------------------------------------
   IBM-supplied default              ABPERC(NONE)
   IBM-supplied default              ABTERMENC(ABEND)
   IBM-supplied default            NOAIXBLD
   PARMLIB(CEEPRM00)                 ALL31(OFF)
   PARMLIB(CEEPRM00)                 ANYHEAP(40960,40960,ANYWHERE,FREE)
   IBM-supplied default            NOAUTOTASK
   IBM-supplied default              BELOWHEAP(8192,4096,FREE)
   IBM-supplied default              CBLOPTS(ON)
   IBM-supplied default              CBLPSHPOP(ON)
   IBM-supplied default              CBLQDA(OFF)
   IBM-supplied default              CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALLOC)
   IBM-supplied default              CHECK(ON)
   IBM-supplied default              COUNTRY(US)
   IBM-supplied default            NODEBUG
   IBM-supplied default              DEPTHCONDLMT(10)
   PARMLIB(CEEPRM00)                 DYNDUMP(*USERID,NODYNAMIC,NOTDUMP)
   IBM-supplied default              ENVAR("")
   IBM-supplied default              ERRCOUNT(0)
   IBM-supplied default              ERRUNIT(6)
   IBM-supplied default              FILEHIST
   IBM-supplied default              FILETAG(NOAUTOCVT,NOAUTOTAG)
   Default setting                 NOFLOW
   PARMLIB(CEEPRM00)                 HEAP(102400,40960,ANYWHERE,KEEP,8192,4096)
   IBM-supplied default              HEAPCHK(OFF,1,0,0,0,1024,0,1024,0)
```

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (5)

- Inhalte der Variablen (wenn mit TEST(SYM) compiliert)

```
Local Variables:
              18 77 LEVEL            X(30) DISP       'P96NDP2 28/04/09 LV034        '
              45 IDX-2           IX                3
              48 IDX-1           IX                4
              19 01 HILFSFELDER      AN-GR
              20  05 PGM-NAME          X(008) DISP      'P96NDP2 '
              21  05 P96NDP4           X(008) DISP      'P96NDP4 '
              22  05 I1                S9(009) COMP     +000000000
              23  05 I1-MAX            S9(009) COMP     +000000000
              24  05 BIN-ZAHL          S9(008) COMP     +00000000


Local Variables:
              18 77 LEVEL            X(30) DISP       'P96NDP1 28/04/09 LV075        '
              19 01 I1               S9(009) COMP     +000000000
              20 01 I1-MAX           S9(009) CMP3     +000000001
              21 01 I1-MAX-BIN       S9(009) COMP     +000000001
```

- Programminformationen (Compile-Datum/-Zeit)

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (6)

- Programminformationen
  - Compile-Datum/-Zeit/-Options

```
Program P96NDP2 was compiled 06/23/19 1:43:40 PM
    COBOL Version = 06 Release = 01 Modification = 00     User Level = 'P190213 '

    Compile Options for P96NDP2:
      NOADV, AFP(VOLATILE), ARCH(11), ARITH(COMPAT), AWO, NOCICS, CODEPAGE(01141),
      NOCOPYRIGHT, DATA(31), NODBCS, DISPSIGN(COMPAT), NODLL, DYNAM, NOEXPORTALL,
      NOFASTSRT, HGPR(PRESERVE), INTDATE(LILIAN), NUMPROC(NOPFD), OPT(2),
      OUTDD(SYSOUT), PGMNAME(COMPAT), QUALIFY(COMPAT), RENT, RMODE(ANY), NOSERVICE,
      NOSQL, SQLCCSID, NOSSRANGE, NOSTGOPT, TEST(NOEJPD,NOSOURCE,NOSEPARATE),
      NOTHREAD, TRUNC(OPT), VLR(STANDARD), XMLPARSE(XMLSS), ZWB
```

LE – Abbruchinformationen

cps4it

Zusammenfassung für die Fehlersuche mittels CEEDUMP

- File, Index und lokale Daten werden alle unter „Local variables“ formatiert.
- Working Storage wird nicht mehr über BLW‘s in TGT (Reg9) adressiert. Beide sind obsolet geworden.
- Working Storage wird ab COBOL V5+ in eine WSA (Writable Static Area) aufgebaut.

Writable Static Area oder...
Wie finde ich meine Working Storage ohne formatierte Debugging Daten im CEEDUMP (DWARF) ?

LE – Abbruchinformationen

CEEDUMP: weitere Details ... (1)

- File Data

```
COB6VSAM (DSA address 19A4D030):
  UPSTACK DSA
  Saved Registers:
    GPR0..... 00000000  GPR1..... 19A4D130  GPR2..... 196001C0  GPR3..... 19600AF8
    GPR4..... 00000000  GPR5..... 19A4D030  GPR6..... 00000000  GPR7..... 99A76FD3
    GPR8..... 19A6D070  GPR9..... 19A6D560  GPR10.... 19600024  GPR11.... 19611070
    GPR12.... 19612378  GPR13.... 19A4D030  GPR14.... 99600506  GPR15.... 198829D8
Local Variables:
       34 OUT-FILE                          FILE SPECIFIED AS: ORGANIZATION=QSAM SEQUENTIAL, ACCESS
                                            MODE=SEQUENTIAL, RECFM=FIXED. CURRENT STATUS OF FILE IS:
                                            OPEN STATUS=OUTPUT, FILE STATUS CODE=00.

       38 01 OUT-RECORD       AN-GR
       39   05 NAME            X(15) DISP       '  VERBESTEL    '
       40   05 FILLER          X(2) DISP        '"'
       41   05 VORNAME         X(10) DISP       '01NORBERT '
       42   05 FILLER          X(2) DISP        '"'
       43   05 PERS-KEY        X(4) DISP        '0016'
       44   05 FILLER          X(2) DISP        '"'
       45   05 SALARY-A        9(5) DISP       X'0000000000'
       46   05 FILLER          X(40) DISP       '                                        '
       47 VSAM-FILE                         FILE SPECIFIED AS: ORGANIZATION=VSAM INDEXED, ACCESS
                                            MODE=DYNAMIC, RECFM=FIXED. CURRENT STATUS OF FILE IS: OPEN
                                            STATUS=INPUT, A SUCCESSFUL ACTION SINCE OPEN, LAST
                                            REQUEST=READ, FILE STATUS CODE=00, VSAM FEEDBACK=000, VSAM
                                            RET CODE=000, VSAM FUNCTION CODE=000.

       49 01 VSAM-SATZ        AN-GR
       50   05 VSAM-KEY        X(4) DISP        '0016'
       51   05 VORNAME         X(10) DISP       '01NORBERT '
       52   05 NAME            X(15) DISP       '  VERBESTEL    '
       53   05 SALARY          X(6) DISP        '  0885'
       54   05 FILLER          X(4) DISP        '00  '
       55   05 MONATE          XX DISP          '  '
       56   . . . . . .
```

LE – Abbruchinformationen

CEEDUMP: weitere Details ... (2)

- Indexes

```
P96NDP2 (DSA address 19A4C488):
     UPSTACK DSA
     Saved Registers:
       GPR0..... E65B3534  GPR1..... 998AAF72  GPR2..... 19A6C220  GPR3..... 19A79B90
       GPR4..... 00000010  GPR5..... 19A4C488  GPR6..... 00000000  GPR7..... 00000000
       GPR8..... 19A6C2D0  GPR9..... 19A6C3C0  GPR10.... 19A79024  GPR11.... 19A4C128
       GPR12.... 19611378  GPR13.... 19A4C488  GPR14.... 99A7946A  GPR15.... 997284B0
     Local Variables:
             18 77 LEVEL            X(30) DISP       'P96NDP2 28/04/09 LV034        '
             45 IDX-2            IX               3
             48 IDX-1            IX               4
     . . . . . .
```

LE – Abbruchinformationen

CEEDUMP: weitere Details ... (3)

- Register für jede save area (DSA) auf Hierarchie
- Speicher rund um jedes Register (GPREG)

```
Parameters, Registers, and Variables for Active Routines:
   CEEHDSP (DSA address 19A4C568):
 COBOLED2 (DSA address 19A4C400):
   UPSTACK DSA
   Saved Registers:
     GPR0..... 00000000  GPR1..... 00000000  GPR2..... 00000000  GPR3..... 19A754D8
     GPR4..... 19A481C8  GPR5..... 19A4C400  GPR6..... 00000000  GPR7..... 0000000A
     GPR8..... 19A6C220  GPR9..... 19A6C2E0  GPR10.... 19A75024  GPR11.... 00000000
     GPR12.... 19611378  GPR13.... 19A4C400  GPR14.... 99A75156  GPR15.... 19A75328
 GPREG STORAGE:
 Storage around GPR0 (00000000)
     +0000 00000000    Inaccessible storage.
     +0020 00000020    Inaccessible storage.
     +0040 00000040    Inaccessible storage.
 Storage around GPR1 (00000000)
     ...
     ...
 Storage around GPR13(19A4C400)
    -0020 19A4C3E0  000000D0 000000D0 00000000 19A6C060  19A6C060 00000000 00000000 00000000  !.............w.-.w.-............!
    +0000 19A4C400  00110301 19A4C1A8 19A4C4F8 99A75156  19A75328 00000000 00000000 00000001  !.....uAy.uD8rx...x.............!
    +0020 19A4C420  19A754D8 19A481C8 19A4C400 00000000  00000000 19A6C220 19A6C2E0 19A75024  !.x.Q.uaH.uD.........wB..wB..x&.!
 Storage around GPR14, etc
    ...
    ...
```

- LE Runtime Option TERMTHDACT(DUMP,,32) or higher needed. 96 would mean 3 lines of storage.
- TERMTHDACT(DUMP,,0) will suppress the GPREG storage print

DSA: Dynamic Save Area

LE – Abbruchinformationen

CEEDUMP: weitere Details ... (4)

- Options Report

```
Run-Time Options Report:

  LAST WHERE SET                 OPTION
  ------------------------------------------------------------------------------
  PARMLIB(CEEPRM01)                ABPERC(NONE)
  PARMLIB(CEEPRM01)                ABTERMENC(ABEND)
  PARMLIB(CEEPRM01)              NOAIXBLD
  PARMLIB(CEEPRM01)                ALL31(OFF)
  PARMLIB(CEEPRM01)                ANYHEAP(16384,8192,BELOW,FREE)
  PARMLIB(CEEPRM01)              NOAUTOTASK
  PARMLIB(CEEPRM01)                BELOWHEAP(8192,4096,FREE)
  PARMLIB(CEEPRM01)                CBLOPTS(ON)
  PARMLIB(CEEPRM01)                CBLPSHPOP(ON)
  PARMLIB(CEEPRM01)                CBLQDA(OFF)
  PARMLIB(CEEPRM01)                CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALLOC)
  PARMLIB(CEEPRM01)                CHECK(ON)
  PARMLIB(CEEPRM01)                COUNTRY(US)
  PARMLIB(CEEPRM01)              NODEBUG
  PARMLIB(CEEPRM01)                DEPTHCONDLMT(10)
  PARMLIB(CEEPRM01)                DYNDUMP(*USERID,NODYNAMIC,NOTDUMP)
  Installation default             ENVAR("")
  PARMLIB(CEEPRM01)                ERRCOUNT(0)
  PARMLIB(CEEPRM01)                ERRUNIT(6)
  . . .
  PARMLIB(CEEPRM01)                NOTEST(DWARF)        -> is needed as a minimum to format local variables in the COBOL V5+ CEEDUMP
  . . .
  PARMLIB(CEEPRM01)                TERMTHDACT(DUMP,,0)  -> controls the amount of storage to be dumped around registers.
  . . .
  INVOCATION COMMAND               RPTOPTS(ON)          -> will print all LE Run Time Options
```

LE – Abbruchinformationen

aber: Sieht das „Live“ auch so aus?

- verschiedene Links

Sourcecode TES39 / P96NDP1
Sourcecode TES47 / P96NDP2
Jobcontrol zu TES39 / P96NDP1
Joblog zu TES39 / P96NDP1

LE – Abbruchinformationen

Übung(en)

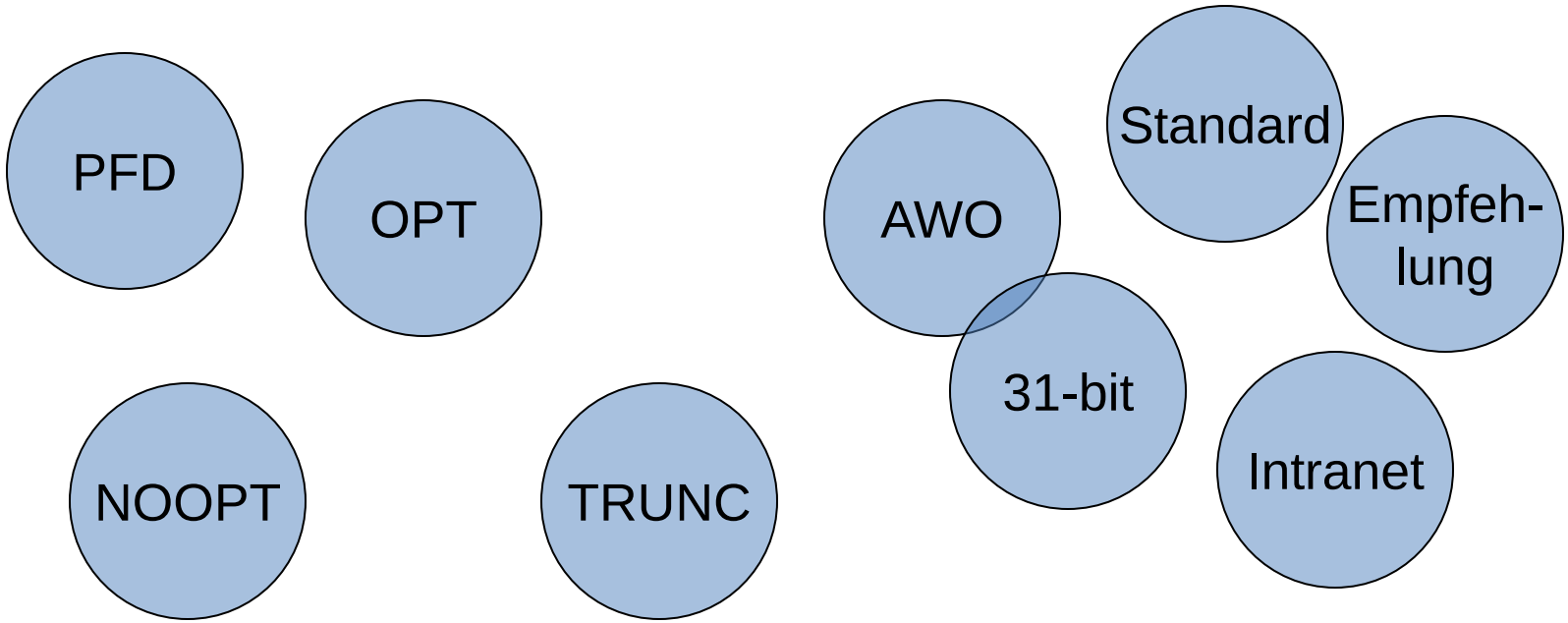
- 04-01 – Wichtige Module im Dump
- 04-02 – Aufbau der COBOL-Meldungen
- 04-03 – Dumpinfos finden in Beispiel
- 04-04 – Dumpinfos in Produktion finden

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

LE – Abbruchinformationen

Begriffe



Linkage Convention und Optionen

Konventionen – Hintergrund

- Innerhalb z/OS gibt es eindeutige Konventionen, wie Register benutzt werden.
  - ASM-HLL-Konvention
  - XPLINK-Konvention
- Grund: einheitliche Beschreibung für
  - Parameterübergabe
  - Sprung hin
  - Sprung zurück
  - Variablenadressierung

Linkage Convention und Optionen

Konventionen – Inhalte

- R1 Parameterliste
- R12 Adresse der CAA (common anchor area)
- R13 Adresse der save area
- R14 Rücksprungadresse
- R15 Adresse Entrypoint
- Rn frei verfügbar

- Konsequenz: alle Inhalte aller Programme zu finden!

LE – Abbruchinformationen

ein einfaches Beispiel – und weiter ... (1)

Diese Seite ist eine Kopie aus einem früherem Kapitel.

```
1CEE3DMP V2 R3.0: Condition processing resulted in the unhandled condition.          06/23/19 1:47:26 PM   Page:    1
 ASID: 00E2   Job ID: JOB08948   Job name: DUMP0C71   Step name: P96NDP1    UserID: XV8822D

 CEE3845I CEEDUMP Processing started.

 Information for enclave P96NDP1

   Information for thread 8000000000000000

   Traceback:
     DSA    Entry       E  Offset  Statement   Load Mod            Program Unit                  Service  Status
     1      CEEHDSP     +00004A4C               CEEPLPKA            CEEHDSP                       HLE77B0  Call
     2      P96NDP2     +00000298  105          P96NDP2             P96NDP2                                Exception
     3      IGZXFCA1    +00002F58               IGZXLPKA            IGZXFCA1                               Call
     4      P96NDP1     +00000336  77           P96NDP1             P96NDP1                                Call

     DSA    DSA Addr    E  Addr     PU Addr     PU Offset  Comp Date  Compile Attributes
     1      000237C8    0CFAEB30    0CFAEB30    +00004A4C  20170307   CEL
     2      00023548    23B7F000    23B7F000    +00000298  20190623   COBOLV5+ EBCDIC  HFP
     3      00023298    23967540    23967540    +00002F58  20190211   LIBRARY
     4      00023030    23900000    23900000    +00000336  20190623   COBOLV5+ EBCDIC  HFP
```

Linkage Convention und Optionen

COBOL-Optionen in Auswahl

- (NO)LIST
- (NO)MAP
- DATA(24/31)
- (NO)DYNAM
- OPT(0,1,2)
- (NO)RENT
- RMODE(24/ANY,AUTO)
- NOTEST(DWARF)
- (NO)SSRANGE
- (NO)OFFSET

Umwandlung P96NDP1 /TES39 LIST

Umwandlung P96NDP2/TES47 LIST

Umwandlung P96NPD2/ TES47 NOLIST

Linkage Convention und Optionen

LINK-Optionen in Auswahl

- AMODE(24,31,64,ANY) - Default is the ESD AMODE
- RMODE(24/ANY) - Default is the ESD AMODE
- MAP=YES/NO
- (NO)REUSE
- (NO)XPLINK

Linkliste P96NDP1/TES39

Linkage Convention und Optionen

LE-Optionen in Auswahl (1)

- ALL31(OFF/ON)
- ANYHEAP(16384,8192,BELOW,FREE)
- BELOWHEAP(8192,4096,FREE)
- HEAP(32768,32768,ANYWHERE,KEEP, 8192,4096)
- . . .

Linkage Convention und Optionen

cps4it

LE-Optionen in Auswahl (2)

- LIBSTACK(4096,4096,FREE)
- STACK(131072,131072,BELOW,KEEP,524288,131072)
- STORAGE(NONE,NONE,NONE,32768)
- NOTEST(DWARF)
- THREADHEAP(4096,4096,ANYWHERE,KEEP)
- THREADSTACK(OFF,4096,4096,BELOW,KEEP,131072,131072)
- XPLINK(OFF)

Linkage Convention und Optionen

Übung(en)

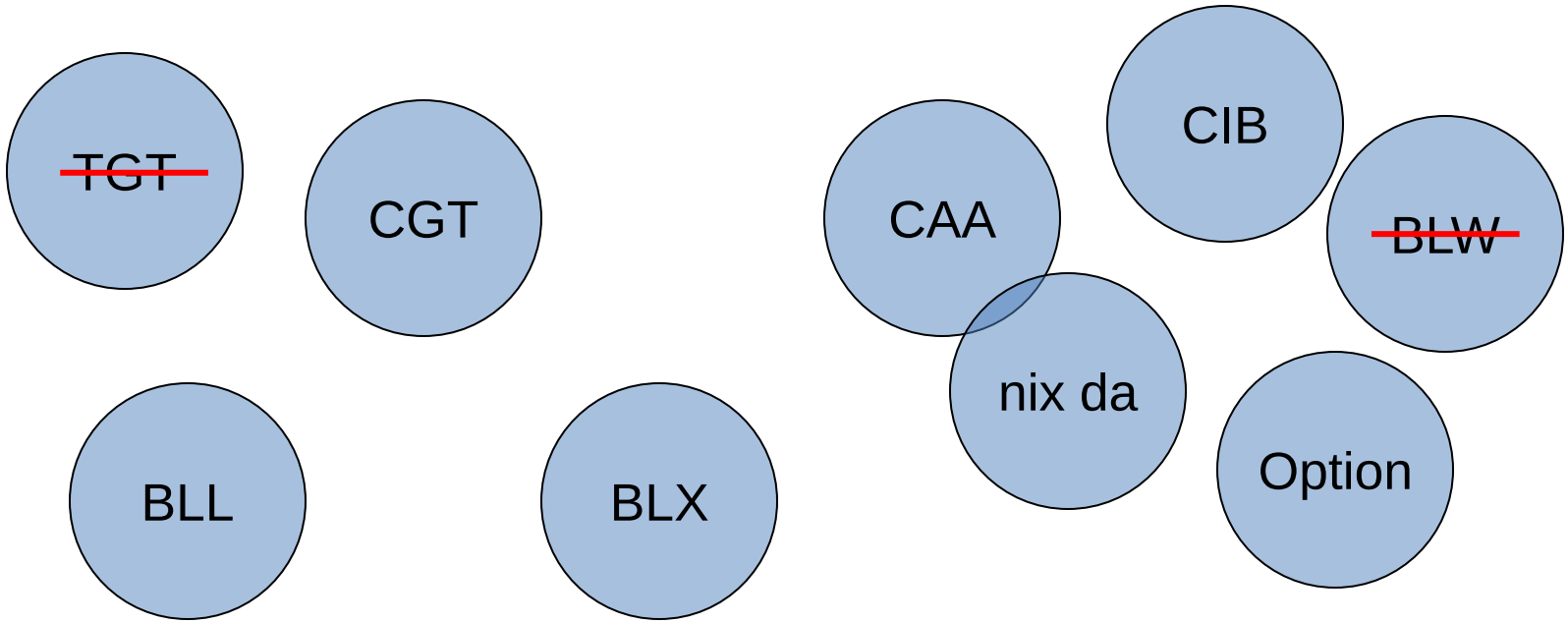
- 05-01 – Programm erstellen
- 05-02 – Umwandlungsoptionen einstellen
- 05-03 – JCL zu Programm erstellen

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

Steuerblöcke in COBOL und LE

Begriffe



Steuerblöcke in COBOL und LE

COBOL – Baselocator

- ~~BLW Working-Storage~~
- BLL Linkage Section
- BLF Files
- BLS Sort Items
- BLX external Data
- BLT Base Locator Table
- BLV Variable Located Data
- IX Indizes

- **Symbols used in LIST and MAP output Programming Guide**

Steuerblöcke in COBOL und LE

COBOL – Systembereiche in Auswahl

- PPA (programm prolog area)
  - reservierter Bereich für Standardinformationen
  - PPA 1, 2, 3, 4
- Constant Area (CGT)
  - Literale und Konstanten
- COBDSACB (COB V5+, TGT Ersatz)
  - Aber ohne Working Storage Adressen
- CLLE (cobol load list entry address)
  - für dynamisch aufgerufene programe
- FCB (file control block)
  - für alle Files

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Assembly

- wichtige Optionen: NOTEST(DWARF),LIST,MAP
- Assembly: Anfang ist Offset 000000 (Init-Code)

```
0 000002:          PROGRAM-ID.    P96NDP2.
      000000                      000002              PROC    P96NDP2
      000000  47F0 F014           000002              BC      R15,20(,R15)          #  Skip over constant area
      000004  01C3 C5C5           000002              DC      X'01C3C5C5'           #  Eyecatcher: CEE
      000008  0000 0280           000002              DC      X'00000280'           #  Stack Size
      00000C  0000 0D38           000002              DC      X'00000D38'           #  Offset to PPA1
      000010  47F0 F001           000002              BC      R15,1(,R15)           #  Wrong Entry Point: cause exception
. . .
      0001E8                      000002  USER-ENTRY: EQU     *
      0001E8                      000002              SNAPSHOT ENTRY
      0001E8                      000074  SCHEIN      EQU     *
      0001E8                      000074              SNAPSHOT PATHLABEL
   000076:             DISPLAY '                     '.
      0001E8                      000076              SNAPSHOT STMT
      0001E8  D217 D140 31E4      000076              MVC     320(24,R13),484(R3)   #
      0001EE  4140 D140           000076              LA      R4,320(,R13)          #  _ArgumentList
      0001F2  5850 31B4           000076              L       R5,436(,R3)           #
      0001F6  58C0 D084           000076              L       R12,132(,R13)         #
      0001FA  1814                000076              LR      R1,R4
      0001FC  18F5                000076              LR      R15,R5
      0001FE  0DEF                000076              BASR    R14,R15               #  Call "IGZXDSP"
```

Offset

Source Line#

Assembler Mnemonics

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Constant Area

- ...beinhaltet die Konstanten die z.B auch als VALUEs definiert wurden

```
CONSTANT AREA

0004F0 (+0)      00CCDDFF 00000000 D7F9F6D5 C4D7F100 00000000 40404040 40404040 40404040      !........P96NDP1.....                !
000510 (+32)     40404040 40000000 00000000 00000000 5C40C1D5 C64B40D7 F9F6D5C4 D7F1405C      !     ...........* ANF. P96NDP1 *!
000530 (+64)     5C406060 60606060 60606060 405CD7F9 F6D5C4D7 F3000000 00000000 001C0C5C      !* ---------- *P96NDP3..........*!
000550 (+96)     40C9F160 D4C1E740 6140C6E4 D5D2E3C9 D6D540E6 C1D94000 00000000 00000000      ! I1-MAX / FUNKTION WAR .........!
000570 (+128)    5C40C5D5 C4C540D7 F9F6D5C4 D7F1405C C9C7E9E2 D9E3C3C4 40000AE2 E8E2D6E4      !* ENDE P96NDP1 *IGZSRTCD ..SYSOU!
000590 (+160)    E3404000 180E0FD7 F9F6D5C4 D7F140F2 F861F0F4 61F0F940 D3E5F0F7 F5D7F9F6      !T  ....P96NDP1 28/04/09 LV075P96!
0005B0 (+192)    D5C4D7F2 00000000 00000000 00000000 00000000 00000000 00000000 00000000      !NDP2............................!
0005D0 (+224)    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000      !................................!
0005F0 (+256)    00000000 00000000 D7F9F6D5 C4D7F340 40000000 00000000 00000000 00000000      !........P96NDP3  ...............!
000610 (+288)    00000000 00000000 00000000 00000040 00000000 00000002 0000000B 00000007      !............... ................!
000630 (+320)    000004F8 0000000B 00000007 000004F8 00000001 01000475 00000504 0000000B      !...8...........8................!
000650 (+352)    00000011 00000000 00000001 01000475 00000520 0000000B 00000010 00000000      !................................!
000670 (+384)    00000001 01000475 00000530 0000000B 0000000E 00000000 00000002 01000475      !................................!
000690 (+416)    0000054F 0000000B 00000018 00000000 00000000 00000011 00000005 00000009      !...!............................!
0006B0 (+448)    00000001 01000475 00000570 0000000B 00000010 00000000 00000006 00000000      !................................!
0006D0 (+480)    D7F9F6D5 C4D7F240 00000008 00000009 00000008 00000001 40404040 40404040      !P96NDP2 ................       !
0006F0 (+512)    40404040 40404040 40404040 40404040 40404040 4040D7F9 F6D5C4D7 F140F2F8      !                      P96NDP1 28!
000710 (+544)    61F0F461 F0F940D3 E5F0F7F5 40404040 40404040 C9C7E9E2 D9E3C3C4 00000000      !/04/09 LV075        IGZSRTCD....!
000730 (+576)    E2E8E2D6 E4E34040 00000000 00000004 00000000 00000050 F0F0F0F0 F0F0F0F0      !SYSOUT  ...............&00000000!
000750 (+608)    F6F0F0F0 F0F0F0F0 F0F1F0F0 F0F0F0F0 F0F0F000 00000000 F1F9F0F4 F4F6F1F0      !6000000001000000000.....19044610!
000770 (+640)    E2E8E2D6 E4E34040 00000000 F1F9F0F4 F4F6F1F0 00000000 000004F8 00000007      !SYSOUT  ....19044610.......8....!
000790 (+672)    04000000 00000000 00000000 00000000 E2E8E2D6 E4E34040 00000000 00000000      !................SYSOUT  ........!
0007B0 (+704)    00000000                                                                     !....                           !
```

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Working Storage Map

- zeigt die Offsets und Längen der einzelnen 01 Level Variablen

```
 * * * * *   W O R K I N G - S T O R A G E   M A P   * * * * *

  OFFSET (HEX)   LENGTH (HEX)   NAME

            0              4       JNIENVPTR
1PP 5655-EC6 IBM Enterprise COBOL for z/OS  6.1.0 P190213       P96NDP1   Date 06/23/2019  Time 13:43:31   Page    24
0           8              2       RETURN-CODE
           10              2       SORT-RETURN
           18              8       SORT-CONTROL
           20              4       SORT-CORE-SIZE
           28              4       SORT-FILE-SIZE
           30              4       SORT-MODE-SIZE
           38              8       SORT-MESSAGE
           40              4       TALLY
           48              1       SHIFT-OUT
           50              1       SHIFT-IN
           58              4       XML-CODE
           60             1E       XML-EVENT
           80              4       XML-INFORMATION
           88              4       JSON-CODE
           90             1E       LEVEL
           B0              4       I1
           B8              5       I1-MAX
           C0              4       I1-MAX-BIN
           C8             77       HILFSFELDER


                 * * * * *   E N D   O F   W - S T O R .   M A P   * * * * *
```

von Compiler generiert

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Local Storage Map

- Zeigt die Offsets und Längen der einzelnen "Local Storage" Variablen

```
* * * * *   A U T O M A T I C     M A P   * * * * *

 OFFSET (HEX)    LENGTH (HEX)    NAME

 Block name: P96NDP1

          80               4       _§CAA
         120               4       TS2=14
         124               4       TS2=13
         128               4       _CACHED_$STATICWSA
         12C               4       _CACHED_$STATIC
         130               4       _CACHED_CRENT
         138               8       _BEtemp312
         140              28       _ArgumentList
         168              50       TS2=8
         1B8              44       COBDSACB
         200              18       VLC_cells
         218               9       _VTS_2
         228              10       _VTS_1
```

von Compiler generiert

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Working Hierarchy

- Zeigt die Offsets und Längen der einzelnen Variablen

```
0Source   Hierarchy and                                  Base      Displacement  Asmblr Data
Data Def
 LineID   Data Name                                      Locator     Structure   Definition        Data Type
Attributes
      2  PROGRAM-ID P96NDP2-------------------------------------------------------------------------------*
     18   77 LEVEL . . . . . . . . . . . . . . . . . .              000000000   DS 30C            Display
     19   1  HILFSFELDER . . . . . . . . . . . . . . .              000000000   DS 0CL198         Group
     20     2  PGM-NAME. . . . . . . . . . . . . . . .              000000000   DS 8C             Display
     21     2  P96NDP4 . . . . . . . . . . . . . . . .              000000008   DS 8C             Display
     22     2  I1. . . . . . . . . . . . . . . . . . .              000000010   DS 4C             Binary
     23     2  I1-MAX. . . . . . . . . . . . . . . . .              000000014   DS 4C             Binary
     24     2  BIN-ZAHL. . . . . . . . . . . . . . . .              000000018   DS 4C             Binary
     25     2  BIN-CHAR. . . . . . . . . . . . . . . .              000000018   DS 4C             Display        R
     27     2  PAC-ZAHL. . . . . . . . . . . . . . . .              00000001C   DS 5P             Packed-Dec
     28     2  PAC-CHAR. . . . . . . . . . . . . . . .              00000001C   DS 5C             Display        R
     30     2  DIS-ZAHL. . . . . . . . . . . . . . . .              000000021   DS 9C             Disp-Num
     31     2  DIS-CHAR. . . . . . . . . . . . . . . .              000000021   DS 9C             Display        R
     33     2  AUSGABE-ZEILE . . . . . . . . . . . . .              00000002A   DS 0CL36          Group
     34       3  FELD-X009 . . . . . . . . . . . . . .              00000002A   DS 9C             Display
     35       3  FILLER. . . . . . . . . . . . . . . .              000000033   DS 1C             Display
```

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Linkage Hierarchy

- Zeigt die Offsets und Längen der einzelnen Variablen

```
0Source   Hierarchy and                                  Base      Displacement  Asmblr Data
Data Def
 LineID   Data Name                                      Locator     Structure   Definition        Data Type
Attributes
      2  PROGRAM-ID P96NDP2-------------------------------------------------------------------------------------*
. . .
51    1  EINGABE-ZEILE . . . . . . . . . . . . . . . . BLL=00001   000000000   DS 0CL79          Group
     52     2  I1-MAX-EINGABE. . . . . . . . . . . . . . BLL=00001   000000000   DS 9C             Disp-Num
     53     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   000000009   DS 1C             Display
     54     2  FELD-1. . . . . . . . . . . . . . . . . . BLL=00001   00000000A   DS 9C             Disp-Num
     55     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   000000013   DS 1C             Display
     56     2  FELD-2. . . . . . . . . . . . . . . . . . BLL=00001   000000014   DS 9C             Disp-Num
     57     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   00000001D   DS 1C             Display
     58     2  FELD-3. . . . . . . . . . . . . . . . . . BLL=00001   00000001E   DS 9C             Disp-Num
     59     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   000000027   DS 1C             Display
     60     2  FELD-4. . . . . . . . . . . . . . . . . . BLL=00001   000000028   DS 9C             Disp-Num
     61     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   000000031   DS 1C             Display
     62     2  FELD-5. . . . . . . . . . . . . . . . . . BLL=00001   000000032   DS 9C             Disp-Num
     63     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   00000003B   DS 1C             Display
     64     2  FELD-6. . . . . . . . . . . . . . . . . . BLL=00001   00000003C   DS 9C             Disp-Num
     65     2  FILLER. . . . . . . . . . . . . . . . . . BLL=00001   000000045   DS 1C             Display
     66     2  FELD-7. . . . . . . . . . . . . . . . . . BLL=00001   000000046   DS 9C             Disp-Num
     68   1  LINKAGE-ZUSATZ. . . . . . . . . . . . . . . BLL=00002   000000000   DS 5000C          Display
```

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – File Section (Fremdpgm)

- Zeigt die Offsets und Längen der einzelnen Variablen

```
****************************************                00268604
 DATA DIVISION.                                         00268704
****************************************                00268804
*                                                       00268904
 FILE SECTION.                                          00269004
 FD  OUT-FILE                                           00269104
     RECORDING MODE IS F                                00269204
     RECORD 80 CHARACTERS                               00269304
     LABEL RECORD STANDARD.                             00269404
 01  OUT-RECORD.                                        00269504 BLF=00001,000000000 0CL80
      05  NAME              PIC X(15).                  00269604 BLF=00001,000000000 15C
      05  FILLER            PIC X(2).                   00269704 BLF=00001,00000000F 2C
      05  VORNAME           PIC X(10).                  00269804 BLF=00001,000000011 10C
      05  FILLER            PIC X(2).                   00269904 BLF=00001,00000001B 2C
      05  PERS-KEY          PIC X(4).                   00270004 BLF=00001,00000001D 4C
      05  FILLER            PIC X(2).                   00270104 BLF=00001,000000021 2C
      05  SALARY-A          PIC 9(5).                   00270204 BLF=00001,000000023 5C
      05  FILLER            PIC X(40).                  00270304 BLF=00001,000000028 40C
 FD  VSAM-FILE                                          00270404
     RECORD 80 CHARACTERS.                              00269304
 01  VSAM-SATZ.                                         00270504 BLF=00002,000000000 0CL80
      05  VSAM-KEY          PIC X(4).                   00270604 BLF=00002,000000000 4C
      05  VORNAME           PIC X(10).                  00270704 BLF=00002,000000004 10C
      05  NAME              PIC X(15).                  00270804 BLF=00002,00000000E 15C
      05  SALARY            PIC X(6).                   00270904 BLF=00002,00000001D 6C
      05  FILLER            PIC X(4).                   00271004 BLF=00002,000000023 4C
      05  MONATE            PIC XX.                     00271104 BLF=00002,000000027 2C
      05  FILLER            etc
```

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – File Record Hierarchy (Fremdpgm)

- Zeigt die Offsets und Längen der einzelnen Variablen

```
Source   Hierarchy and                    Base       Displacement  Asmblr Data              Data Def
LineID   Data Name                       Locator    Structure     Definition    Data Type  Attributes
 5  PROGRAM-ID COB6VSAM----------------------------------------------------------------------------------------
    34   FD OUT-FILE. . . . . . . . . . .BLF=00001                               QSAM          F
    38   1  OUT-RECORD. . . . . . . . . .BLF=00001   000000000     DS 0CL80        Group
    39     2  NAME. . . . . . . . . . . .BLF=00001   000000000     DS 15C          Display
    40     2  FILLER. . . . . . . . . . .BLF=00001   00000000F     DS 2C           Display
    41     2  VORNAME . . . . . . . . . .BLF=00001   000000011     DS 10C          Display
    42     2  FILLER. . . . . . . . . . .BLF=00001   00000001B     DS 2C           Display
    43     2  PERS-KEY. . . . . . . . . .BLF=00001   00000001D     DS 4C           Display
    44     2  FILLER. . . . . . . . . . .BLF=00001   000000021     DS 2C           Display
    45     2  SALARY-A. . . . . . . . . .BLF=00001   000000023     DS 5C           Disp-Num
    46     2  FILLER. . . . . . . . . . .BLF=00001   000000028     DS 40C          Display
    47   FD VSAM-FILE . . . . . . . . . .BLF=00002                               VSAM          F
    49   1  VSAM-SATZ . . . . . . . . . .BLF=00002   000000000     DS 0CL80        Group
    50     2  VSAM-KEY. . . . . . . . . .BLF=00002   000000000     DS 4C           Display
    51     2  VORNAME . . . . . . . . . .BLF=00002   000000004     DS 10C          Display
    52     2  NAME. . . . . . . . . . . .BLF=00002   00000000E     DS 15C          Display
    53     2  SALARY. . . . . . . . . . .BLF=00002   00000001D     DS 6C           Display
    54     2  FILLER. . . . . . . . . . .BLF=00002   000000023     DS 4C           Display
    55     2  MONATE. . . . .  etc….
```

Steuerblöcke in COBOL und LE

COBOL – Compiler Listing – Static Map

- Zeigt die Offsets der Indexe
- Pointer zu Linkage-Parametern
- . . .

```
                * * * * *  S T A T I C    M A P  * * * * *

OFFSET (HEX)   LENGTH (HEX)   NAME

          0              4       IDX-2
          4              4       IDX-1
          8              8       BLL_Ptrs
         10              C       BLT_Ptrs
         1C             18       VNI_cells
         34             60       GPCB
         94              4       WS-BASE-ADDRESS
         98              8       TS2=45


                * * * * *  E N D    O F    S T A T I C    M A P  * * * * *
```

Steuerblöcke in COBOL und LE

LE – Steuerblöcke

- CIB (condition information block)
  - Informationen rund um den Abbruch
    - Abbruchadresse
    - Art des Abbruchs
    - PSW
    - Registerinhalte
- CAA (common anchor area)
  - wichtige Adressen wie CIB, PCB, DSA
  - existiert pro Thread
- DSA (dynamic save area)
  - mit allen Basisregistern für Kommunikation
  - verwendet in allen LE Komponenten und Sprachen

Steuerblöcke in COBOL und LE

cps4it

Übung(en)

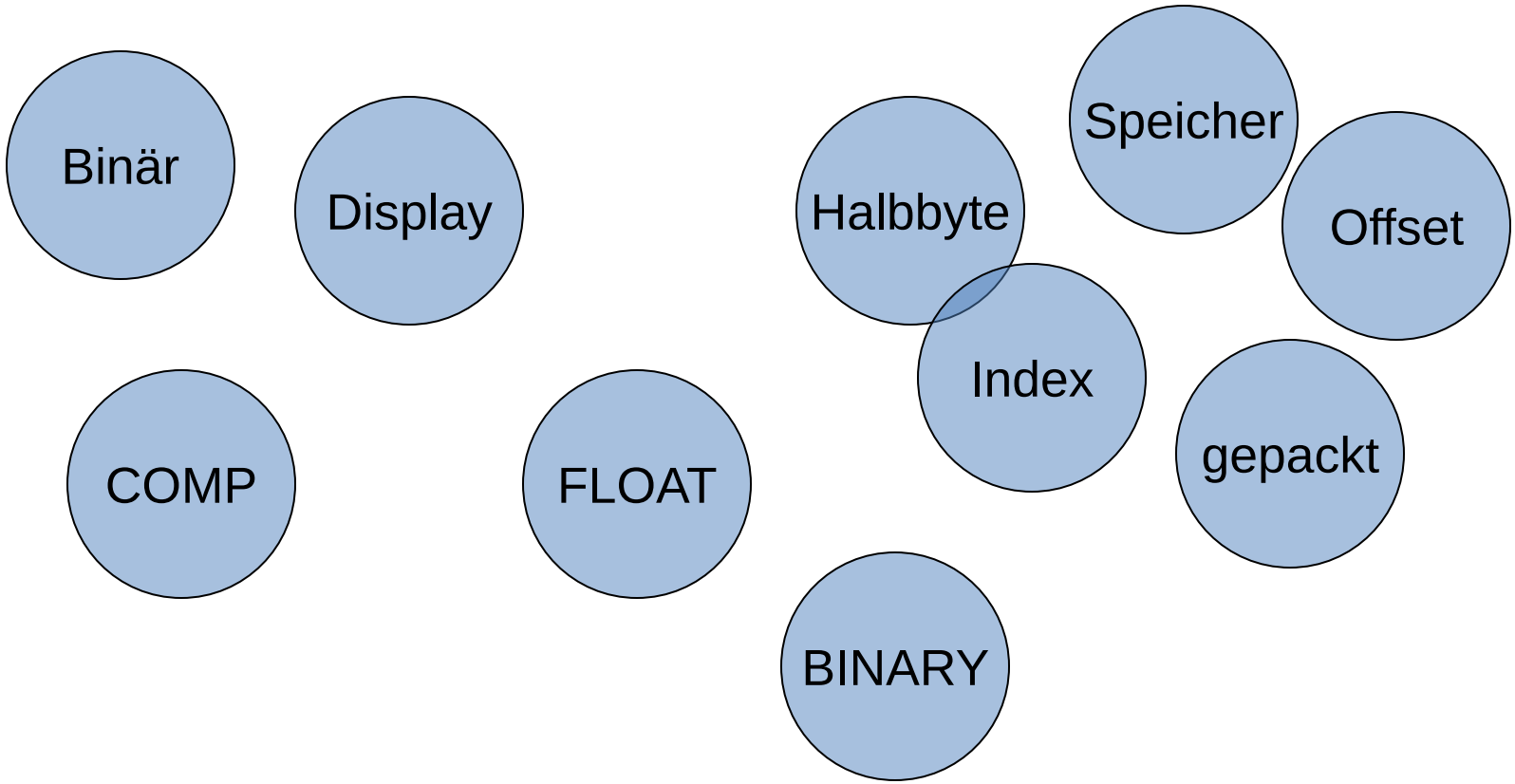
- 06-01 – Steuerblöcke in Umwandlungsliste
- 06-02 – Tests des Programms
  - Abbruch 0C7 auf verschiedenen Feldern
  - Abbruch 0CB auf einem Feld
  - Abbruch 0C4
  - Abbruch U1066
- 06-03 – Besonderes im IMS

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

Numerische Daten

Begriffe



Numerische Daten

External decimal

- PIC S9999 [DISPLAY]

  +1234 F1 F2 F3 C4

  -1234 F1 F2 F3 D4

  1234 F1 F2 F3 C4

- PIC 9999 [DISPLAY]

  1234 F1 F2 F3 F4

```
*
 01  WERT-OHNE-VZ       PIC  9999.
 77  WERT-MIT-VZ        PIC S9(4).
 01  WERT-MIT-VZ        PIC S9(04) DISPLAY.
```

Numerische Daten

Internal decimal

- PIC S9(5) PACKED DECIMAL oder COMP-3

  +1234 01 23 4C

  -1234 01 23 4D

- PIC 9(5) PACKED DECIMAL oder COMP-3

  +1234 01 23 4F

  -1234 01 23 4F

```
*
 01  WERT-MIT-VZ        PIC S99999 PACKED DECIMAL.
```

Numerische Daten

binär

- PIC S9(4) BINARY oder COMP oder COMP-4

  +1234 04 D2

  -1234 FB 2E

- PIC 9(4) BINARY oder COMP oder COMP-4

  +1234 04 D2

*

```
01  WERT-MIT-VZ       PIC S9999 BINARY.
```

Link binäre Zahlen und Calc

Numerische Daten

Index – Beispiel

- Inhalt im Dump für IDX-1: B0
- Inhalt im Dump für IDX-2: 6C

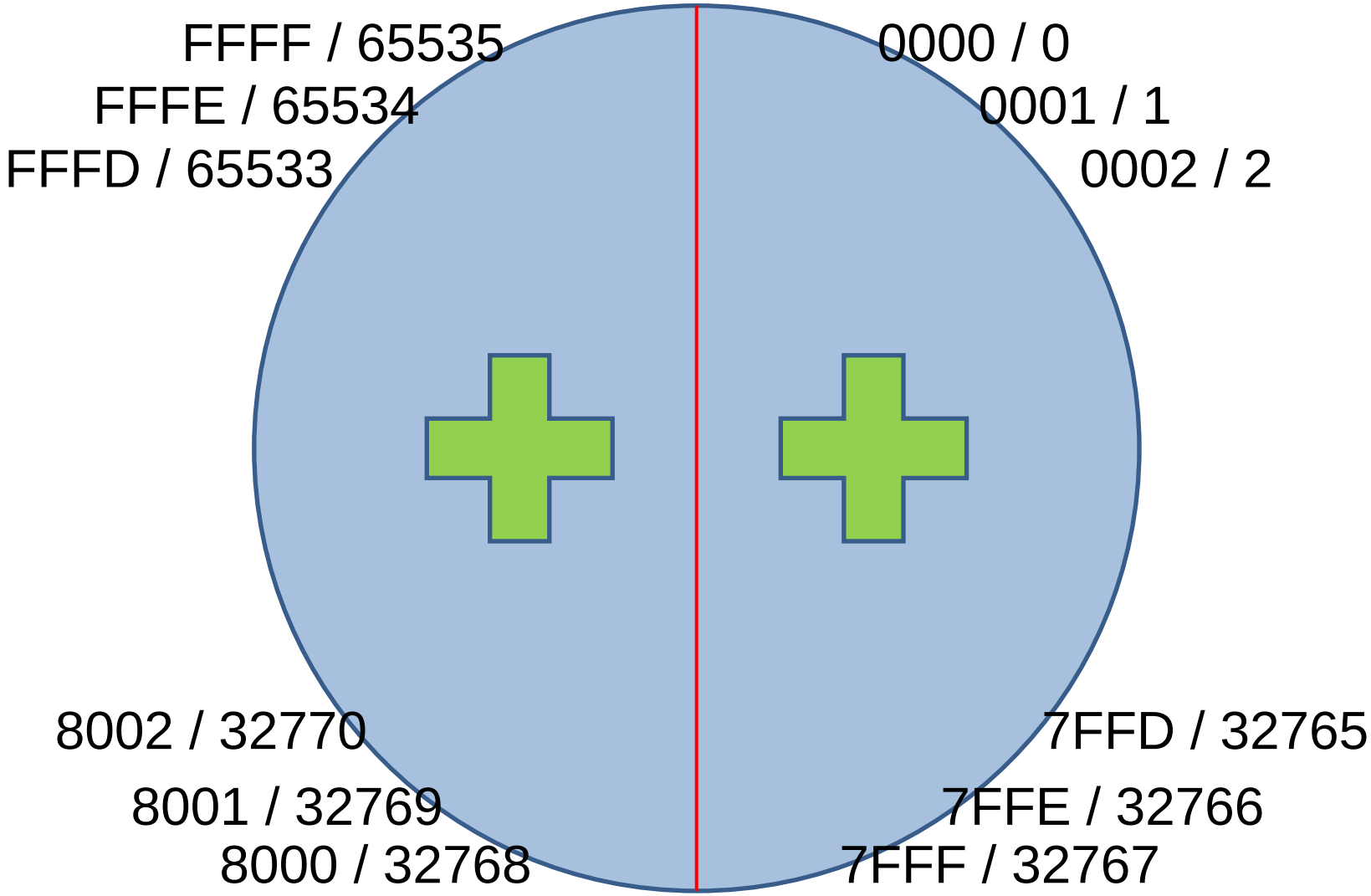
*

```
01   TAB1     OCCURS 5 PIC X(088) INDEXED BY IDX-1.
01   TAB2     OCCURS 7 PIC X(027) INDEXED BY IDX-2.
```

- Berechnung des Subscripts:
  - IDX-1 (x'B0' = 176): (176/88) + 1 = 3
  - IDX-2 (x'6C' = 108): (108/27) + 1 = 5

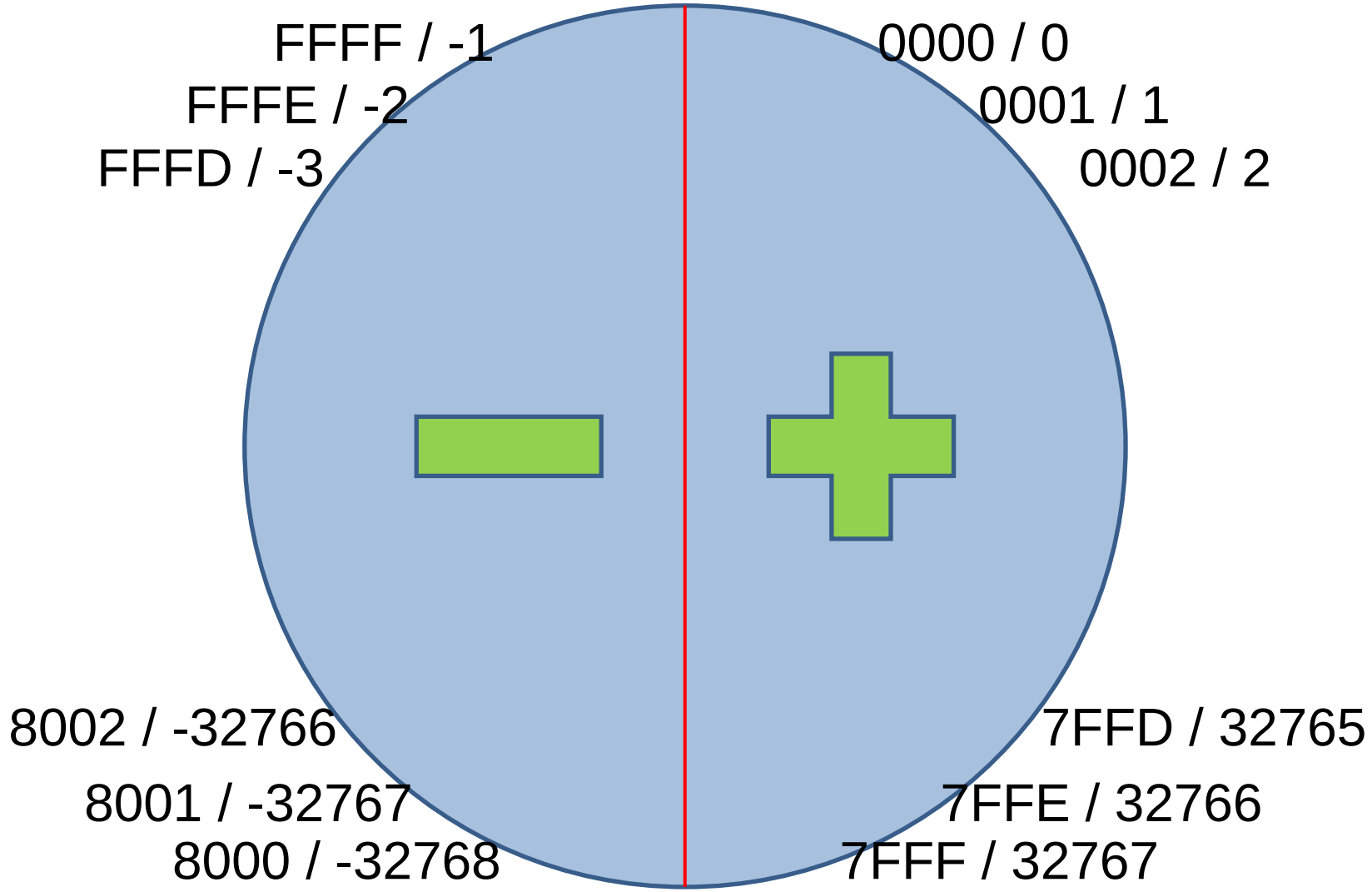
der binäre Kreis

ohne Vorzeichen – 2 Byte Feld (PIC 9(4) binary / – / –)



der binäre Kreis

mit VZ – 2 Byte Feld (PIC S9(4) binary / fixed bin(15) / smallint)



Numerische Daten

Internal Floating Point

- COMP-1

  +1234 43 4D 20 00

- COMP-2

  +1234 43 4D 20 00 00 00 00 00

  -1234 C3 4D 20 00 00 00 00 00

```
*
 01   WERT-MIT-FP        COMP-1.
```

- Logik:
  - The leftmost bit contains the sign and the next 7 bits contain the exponent; the remaining 3 or 7 bytes contain the mantissa.

Numerische Daten

External Floating Point

- PIC +9(2).9(2)E+99 [DISPLAY]

  +1234 4E F1 F2 4B F3 F4 C5 4E F0 F2

  -1234 60 F1 F2 4B F3 F4 C5 4E F0 F2

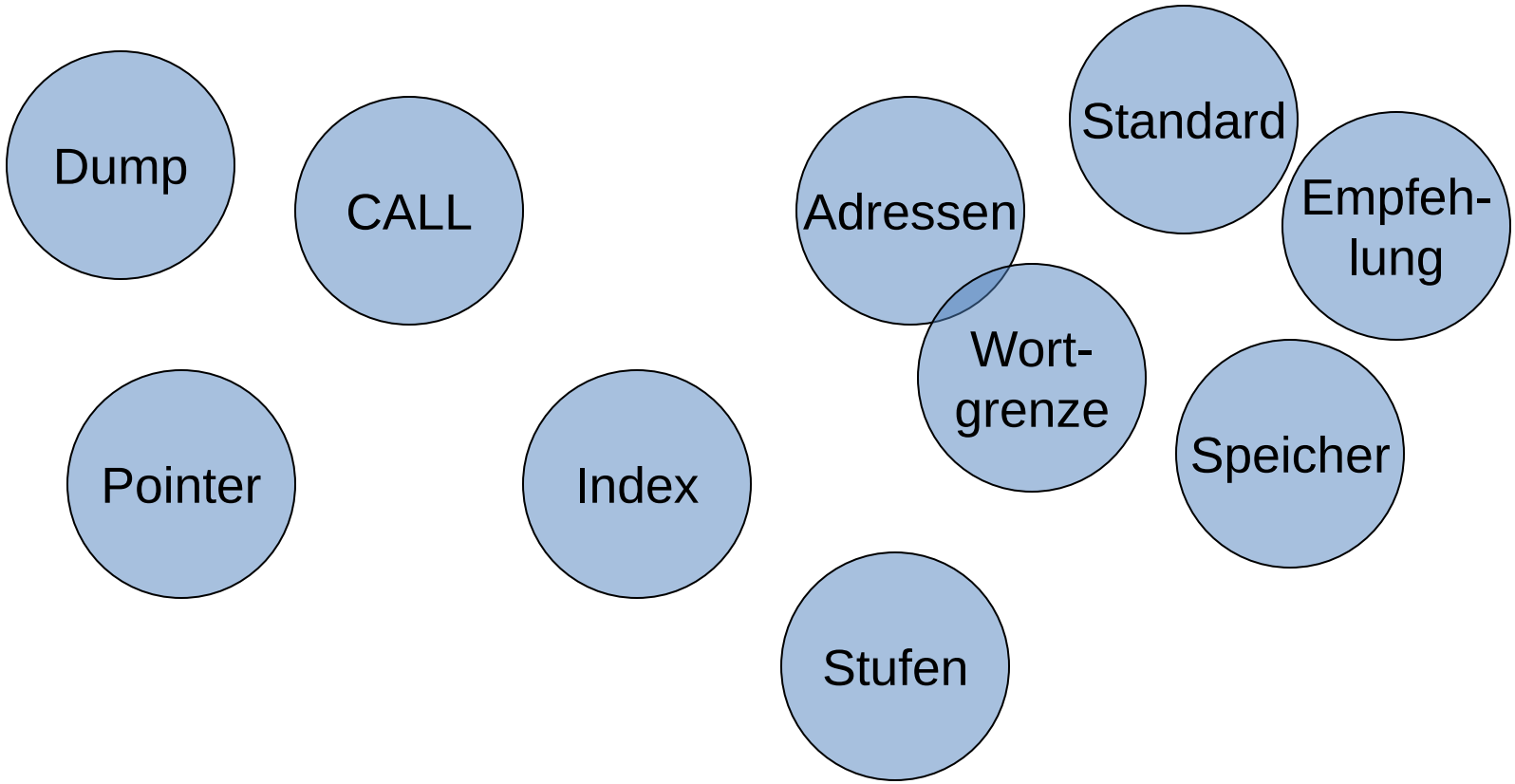
```
*
 01  WERT-MIT-EXP       PIC +99.99E+99.
```

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch

Programmiertechniken

Begriffe



CALL

- Parameterübergabe
  - CALL pgm-name USING BY CONTENT var-name
  - CALL pgm-name USING BY REFERENCE var-name
- statischer CALL / dynamischer CALL
  - CALL ‘TES47’ USING HUGO
  - CALL TES47 USING HUGO

Variablen und ihr Speicher

- SET data-name TO pointer
- RENAMES
- REDEFINES
- Bedingungsnamen
- nicht benutzte Variablen + OPT(FULL)
- Indices
- SYNCHRONIZED
- JUSTIFIED

Programmiertechniken

Übung(en)

- 07-01 – Index statt Subscript
- 07-02 – Variablendefinitionen / Konsequenzen
  - REDEFINES
  - 88-er Stufe
  - nicht benutzte Variablen
  - SYNCHRONIZED

Inhalt

- Vorstellung und Einführung
- Language Environment – Program Management
- Language Environment – Condition Handling
- Language Environment – Abbruchinformationen
- Linkage Convention und Optionen
- Steuerblöcke in COBOL und LE
- Numerische Daten
- Programmiertechniken
- Zusammenfassung – Diskussion – Austausch