

COBOL Teil II  
Übungen zur Schulung

4. Januar 2013

Eine Ausarbeitung von:

**cps4it**

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen  
Fon: +49-6721-992611 • Fax: +49-6721-992613 • Mail: [ralf.seidler@cps4it.de](mailto:ralf.seidler@cps4it.de)  
Internet : <http://www.cps4it.de>  
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

## Inhaltsverzeichnis

<b>1</b>	<b>VORBEREITUNGEN</b>	<b>3</b>
1.1	ANMELDEN UND TEST DER USER-IDENT	3
1.2	BIBLIOTHEK FÜR ÜBUNGSAUFGABEN ERSTELLEN	3
1.3	BIBLIOTHEK FÜR JOBCONTROL ERSTELLEN	3
<b>2</b>	<b>STRUKTUREN UND TABELLEN</b>	<b>4</b>
2.1	KUNDENSTAMMSATZ (COPYBOOK)	4
2.2	INITIALISIEREN VON STRUKTUREN	4
2.3	ERWEITERN VON STRUKTUREN	5
2.4	DEFINIEREN EINER JAHRESTABELLE	5
2.5	JAHRESTABELLE MIT SUBSCRIPTEN	5
2.6	JAHRESTABELLE DURCHSUCHEN	6
2.7	TABELLE MIT VARIABLER SATZANZAHL	6
<b>3</b>	<b>ZEICHENKETTEN</b>	<b>7</b>
3.1	INSPECT	7
3.2	STRING / UNSTRING	7
3.3	STRING / UNSTRING – BESTIMMTE ZAHLENFOLGEN	7
<b>4</b>	<b>COMPILER</b>	<b>8</b>
4.1	COMPILE-OPTIONEN	8
4.2	TRUNC	8
<b>5</b>	<b>UNTERPROGRAMME</b>	<b>9</b>
5.1	STADTVERWALTUNGSDATEI VERARBEITEN – 1	9
5.2	STADTVERWALTUNGSDATEI VERARBEITEN – 2	9
5.3	DATENKAPSELUNG	9
<b>6</b>	<b>PERFORMANCE</b>	<b>10</b>
6.1	NUMMERISCHE DATEN	10
6.2	INITIALISIEREN	10

## 1. Vorbereitungen

---

### 1.1 Anmelden und Test der User-Iden

---

Melden Sie sich nach Vorgabe im TSO an.  
Melden Sie sich wieder ab und erneut wieder an.  
Konfigurieren Sie Ihre TSO-Session, wie Sie am besten arbeiten können.  
Testen Sie, ob Sie auf die Dateien des Referenten lesend zugreifen können.

### 1.2 Bibliothek für Übungsaufgaben erstellen

---

Erstellen Sie eine PO-Datei, um die Übungsaufgaben dieses Seminars aufnehmen zu können.

Name der Bibliothek	userid.KURS.COBOLO
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 2 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis.

### 1.3 Bibliothek für Jobcontrol erstellen

---

Erstellen Sie eine PO-Datei, um die Übungsaufgaben dieses Seminars aufnehmen zu können.

Name der Bibliothek	userid.KURS.JCLCOBOLO
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 2 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis.

## 2. Strukturen und Tabellen

---

### 2.1 Kundenstammsatz (Copybook)

---

Member: COP12x1

Erstellen Sie ein Copybook in SCLM mit folgendem Aufbau:

Feldname	Typ	Größe
Nachname	alphanumerisch	20 Zeichen
Vorname	alphanumerisch	20 Zeichen
Kundennummer-1	numerisch	4 Ziffern
Kundennummer-2	numerisch	2 Ziffern
Kundennummer-3	alphanumerisch	1 Zeichen
Kundennummer-4	numerisch	3 Ziffern
Kunde seit	numerisch	8 Ziffern
Postleitzahl	alphanumerisch	10 Zeichen
Ort	alphanumerisch	20 Zeichen
Straße	alphanumerisch	20 Zeichen
Hausnummer	alphanumerisch	5 Zeichen

Ziehen Sie das Copybook in einem bestehenden Programm an und korrigieren Sie eventuelle Fehler in der Struktur.

### 2.2 Initialisieren von Strukturen

---

Member: PGM1201

Initialisieren Sie das Copybook COP1201 einmal mit MOVE und einmal mit INITIALIZE.

Testen Sie die Laufzeit des Programms, indem Sie es bis zu 10.000.000 Mal laufen lassen.

Welcher Code ist schneller?

### 2.3 Erweitern von Strukturen

---

Member: COPY12x2

Der Kundenstammsatz soll um Kontonummern erweitert werden. Die Kontonummer ist 9 Stellen numerisch. Der Kunde kann bis zu 10 Kontonummern haben, wobei er für jede Kontonummer eine separate Anschrift haben kann.

Definieren Sie die neue Struktur, initialisieren Sie diese mit MOVEs und INITIALIZE und vergleichen Sie die Laufzeiten.

Nutzen Sie den Maschinen-Index.

### 2.4 Definieren einer Jahrestabelle

---

Member: PGM1202

Legen Sie ein neues Programm an. Darin definieren Sie eine Tabelle mit 52 Wochen und 7 Tagen. Die Felder auf der untersten Ebene sollen als Werte den Wochentag, den laufenden Tag der Woche und den laufenden Tag des Jahres aufnehmen.

Initialisieren Sie die Tabelle geschickt für das Jahr 2013.

Nutzen Sie den Maschinen-Index.

### 2.5 Jahrestabelle mit Subscripten

---

Member: PGM1203

Für die vorher genannten Tabellen haben Sie den (Maschinen)Index benutzt. Arbeiten Sie jetzt mit Subscripten. Wählen Sie unterschiedliche Definitionen dafür. Lassen Sie die Verarbeitungen bis zu 10.000.000 Mal laufen und vergleichen Sie die Laufzeiten.

## 2.6 Jahrestabelle durchsuchen

---

Member: PGM1204

Suchen Sie in der Wochen/Tage-Tabelle den Tag des Jahres, den Sie über SYSIN einlesen.

- Nutzen Sie den SEARCH-Befehl.
- Nutzen Sie binäres Suchen.
- Kodieren Sie den Such-Algorithmus per Hand.
- Was ist der schnellste Weg?

## 2.7 Tabelle mit variabler Satzanzahl

---

Member: PGM1205

Legen Sie in dem Programm aus dem Grundlagen-Kurs eine Section an, in der Sie die Verarbeitung mit variabler Satz-Anzahl kodieren.

- Nutzen Sie die Datei der Stadtverwaltung.
- Achten Sie auf die richtige Definition der Zähler.
- Kann die Logik anders als im Beispiel kodiert werden?
- Welche Werte stehen nach der Schleife in den Zählern? Gibt es Unterschiede je nach Art der Kodierung?

### 3. Zeichenketten

---

#### 3.1 Inspect

---

Member: PGM1301 (Basis: PGM1204)

Legen Sie eine Section (bzw. mehrere Sections) an, in der Sie verschiedene Formen des INSPECT ausprobieren. Beispiele:

- Wochentage (MO, DI) in (01, 02) verändern.
- Monatsnamen in 01, 02 verändern. Geht das?
- Suchen Stelle, in der der Monat Mai in der Wochen/Monatstabelle auftritt.
- Datei einlesen. Alle blanks werden in low-value verändert. Gleichzeitig sollen alle low-values in blanks umgesetzt werden. Funktion 2 Mal laufen lassen. Ist der alte Inhalt wieder da?

#### 3.2 String / Unstring

---

Member: PGM1302

Legen Sie in einem neuen Programm eine Section an, in der Sie Ihre Adresse, getrennt durch einen Stern zusammensetzen.

Legen Sie eine weitere Section an, in der Sie die zusammen gesetzte Adresse wieder auseinander nehmen. Schaffen Sie es, den ursprünglichen Zustand wieder herzustellen? Gibt es Fälle, bei denen das nicht erfolgreich ist?

#### 3.3 String / Unstring – bestimmte Zahlenfolgen

---

Member: PGM1303

Für eine bestimmte Anwendung müssen beginnend ab 1 bis zu einer Zahl x, die durch 4 teilbar sein muss, zwei Zeilen ausgegeben werden:

1;2;5;6;9;10;13;14; etc.

3;4;7;8;11;12;15;16 etc.

Wie groß darf x maximal sein, wenn die Zeilen kleiner als 80 sein müssen?

Wenn die Zeilen größer als 80 werden, geben Sie die Ausgabe mehrzeilig aus.

## 4. Compiler

---

### 4.1 Compile-Optionen

---

Sehen Sie sich das Compile-Menü / die SCLM-Methoden näher an.  
Nehmen Sie ein beliebiges eigenes Programm und compilieren (und testen) Sie es mit unterschiedlichen Optionen.  
Nehmen Sie ein Programm, das mit einem Subscript fehlerhaft arbeitet und testen Sie dieses mit SSRANGE. (PGM0703 / P96N701)  
Fehlermeldung beachten.

### 4.2 Trunc

---

Member: PGM1401

Testen Sie die maximalen Werte eines binären Feldes mit den Einstellungen TRUNC(STD) und TRUNC(BIN).

Wandeln Sie Ihr Programm mit der Option LIST um. Schauen Sie sich das Ergebnis an.

Wandeln Sie Ihr Programm mit der Option LIST und achten Sie auf die Unterschiede

NUMPROC(PFD)

NUMPROC(NOPFD)



## 5. Unterprogramme

---

### 5.1 Stadtverwaltungsdatei verarbeiten – 1

---

Member: PGM1501 / PGM15011

Schreiben Sie ein Programm, das ein Unterprogramm aufruft. Im Hauptprogramm lesen Sie die Daten der Stadtverwaltung mit READ ein. Das Unterprogramm wird für Satzart-1 = 1 aufgerufen. Dieses soll die übergebenen Daten in einer "lesbaren" Form ausgeben.

- Nur für R+V: Denken Sie daran, das Unterprogramm im SCLM korrekt zu definieren.
- Nur für Commerzbank: Denken Sie daran, das Unterprogramm im Datenhaushalt bekannt zu geben.

### 5.2 Stadtverwaltungsdatei verarbeiten – 2

---

Member: PGM1501/ PGM15011 / PGM15012 / PGM15013

Erweitern Sie das Hauptprogramm, indem es eine Verarbeitungsart über ACCEPT einliest:

1 = Standesamt

2 = Einwohnermeldeamt

3 = Polizeiamt

Für jede Verarbeitungsart soll es ein separates Unterprogramm geben, das die Daten ausgibt.

### 5.3 Datenkapselung

---

Member: PGM1502/ PGM15021 / PGM15022 / PGM15023

Testen Sie ein Programm mit einem Unterprogramm, das ein Feld "by content" und eines "by reference" übergibt. Im Hauptprogramm und im Unterprogramm werden die Felder jeweils unterschiedlich gesetzt. Drucken Sie sich die Feldinhalte jeweils vor und nach der Modifikation und vor und nach dem UP-Aufruf an.

## 6. Performance

---

### 6.1 numerische Daten

---

Member: PGM1601

Schreiben Sie ein kleines Programm, in dem 2 mal 3 Felder mit den Definitionen PIC S9(09), PIC S9(09) PACKED-DECIMAL, PIC S9(08) BINARY enthalten sind. Übertragen Sie die Inhalte des ersten Blocks in jedes Feld des zweiten Blocks.

Lassen Sie das jeweils 10.000.000 Mal laufen und vergleichen Sie. Prüfen Sie eventuell den ASM Code.

### 6.2 Initialisieren

---

Member: PGM1602

Definieren Sie eine Tabelle, die Sie initialisieren. Tun Sie dieses unterschiedlich und schauen Sie sich bei 10.000.000 Aufrufen die Unterschiede in der CPU-Zeit an.