

COBOL Grundlagen

Übungen zur Schulung

14. August 2019

Eine Ausarbeitung von:

cps4it

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>
Steuernummer: 08/226/4756/9, Finanzamt Bingen, Ust-ID : DE214792185

Inhaltsverzeichnis

1. VORBEREITUNGEN	3
1.1 ANMELDEN UND TEST DER USER-IDENT	3
1.2 BIBLIOTHEK FÜR ÜBUNGSAUFGABEN ERSTELLEN	3
1.3 BIBLIOTHEK FÜR JOBCONTROL ERSTELLEN	3
2. PROGRAMMAUFBAU	4
2.1 DAS ERSTE PROGRAMM	4
3. ARBEITEN MIT NUMMERISCHEN DATEN	4
3.1 ANZEIGE NUMMERISCHE FORMATE	4
3.2 RECHNEN MIT NUMMERISCHEN FORMATEN	5
3.3 DRUCKAUFBEREITUNG VON NUMMERISCHEN FORMATEN	5
4. SCHLEIFEN MIT ZÄHLERN	6
4.1 ABWEISENDE SCHLEIFE	6
4.2 NICHT-ABWEISENDE SCHLEIFE	6
4.3 DEFINITION VON LAUFVARIABLEN	7
4.4 ADDITION / MULTIPLIKATION	7
5. VERZWEIGUNGEN / BEDINGUNGEN	8
5.1 EINFACHER IF	8
5.2 VERSCHACHELTER IF (1)	8
5.3 VERSCHACHELTER IF (2)	8
5.4 BEDINGUNGSNAMEN	9
5.5 EVALUATE	9
5.6 EVALUATE - VOLLSTÄNDIG	9
6. SCHLEIFEN MIT BEDINGUNGEN	10
6.1 LESEN DER SCHULUNGSDATEI	10
7. SECTIONS	11
7.1 LESEN DER SCHULUNGSDATEI	11
7.2 LESEN DER SCHULUNGSDATEI – TABELLE	11
7.3 LESEN DER SCHULUNGSDATEI – TABELLE / DATEI	11
7.4 LESEN DER SCHULUNGSDATEI – GRUPPENWECHSEL	11
8. TABELLEN	12
8.1 DEFINITION EINER TABELLE	12
8.2 INITIALISIEREN EINER TABELLE	12

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.KURSCOB.COBOL
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 50 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis. (Am besten als „Library“ anlegen.)

1.3 Bibliothek für Jobcontrol erstellen

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

Name der Bibliothek	userid.KURSCOB.JCL
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 50 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis. (Am besten als „Library“ anlegen.)

2. Programmaufbau

Hinweis: Die Programme werden in SCLM geschrieben unter einem beliebigen Namen. Das Ergebnis wird in die Datei uid.KURSCOB.COBOL unter dem in den Übungsaufgaben angegebenen Namen kopiert. Ziel ist es, in der Datei alle Programme zu den Aufgaben zu haben.

2.1 Das erste Programm

Member: PGM0201

Schreiben Sie ein Programm, das 80 Stellen aus SYSIN einliest. Der Inhalt soll in ein zweites Feld übertragen werden. Der Inhalt des zweiten Feldes ist auszugeben. Geben Sie auch Informationen aus, wie Ihr Programm heißt und welche Informationen Sie ausgeben.

Benutzen Sie die Befehle ACCEPT, DISPLAY und MOVE.

3. Arbeiten mit numerischen Daten

3.1 Anzeige numerische Formate

Member: PGM0301

Schreiben Sie ein Programm, das ein 8-stelliges numerisches Feld einliest. Diese wird in weitere Felder übertragen mit den Definitionen:

- Display-Feld, 8 Stellen, ohne Vorzeichen
- Display-Feld, 8 Stellen, mit Vorzeichen
- gepacktes Feld, 8 Stellen, ohne Vorzeichen
- gepacktes Feld, 8 Stellen, mit Vorzeichen
- binäres Feld, 8 Stellen, ohne Vorzeichen
- binäres Feld, 8 Stellen, mit Vorzeichen
- Fließkommefeld (kurze Form)

Geben Sie alle Ergebnisfelder aus. Beschreiben Sie, welche Angabe Sie in welcher Zeile drucken.

Prüfen Sie das Ergebnis. Ist das Ergebnis so, wie Sie es erwartet haben? Wird der Speicher (Storage) angezeigt oder wird konvertiert?

Versuchen Sie einen Abbruch zu erzeugen. Der Referent wird den Abbruch erklären.

3.2 Rechnen mit numerischen Formaten

Member: PGM0302

Schreiben Sie ein Programm, das die folgenden Aufgaben erfüllt:

- (F1) gepacktes Feld, 9 Stellen, davon 2 Nachkommastellen; Wert: 1000 bzw. 225,75
- (F2) gepacktes Feld, 9 Stellen, davon 2 Nachkommastellen; Wert: 75 bzw. 7,5
- (F3) gepacktes Feld, 7 Stellen, davon 0 Nachkommastellen; Wert: 225,75 bzw. 1000
- (F4) gepacktes Feld, 7 Stellen, davon 0 Nachkommastellen; Wert: 5 bzw. 5
- Ein einziges gepacktes Feld als Ergebnisfeld, das in der Lage ist, alle zu berechnenden Werte aufzunehmen.

Die folgenden Rechenoperationen sind durchzuführen und zu prüfen. Die Berechnungen sollen jeweils mit und ohne Rundung erfolgen. Als Ergebnisfeld soll nur ein einziges Feld definiert werden.

- $F1 + F2$
 - $F1 * F2$
 - $F4 / F1$
 - $F3 - F1$
 - $F2 ** F4$
 - $(F1 + F2) * F2 + F3 - F4 ** F4$
- Was passiert, wenn Sie anders klammern?

3.3 Druckaufbereitung von numerischen Formaten

Member: PGM0303

Schreiben Sie ein Programm mit verschiedenen Druckaufbereitungsarten. Probieren Sie Währung, Nullen unterdrücken, Dezimalzahlen, Stern etc. aus.

4. Schleifen mit Zählern

4.1 abweisende Schleife

Member: PGM0401

Schreiben Sie ein Programm mit einer abweisenden Schleife, indem Sie eine Variable von 1 bis 10 laufen lassen. Setzen Sie den Anfangswert der Laufvariablen so, dass Sie die Schleife „sauber“ durchlaufen wird und in einem weiteren Fall so, dass Sie nicht durchlaufen wird.

Innerhalb der Schleife zeigen Sie den Wert der Laufvariablen an. Zusätzlich soll eine Variable die Summe der Laufvariablen beinhalten.

4.2 nicht-abweisende Schleife

Member: PGM0402

Schreiben Sie ein Programm mit einer nicht-abweisenden Schleife, indem Sie eine Variable von 1 bis 10 laufen lassen. Setzen Sie den Anfangswert der Laufvariablen so, dass Sie die Schleife „sauber“ durchlaufen wird und in einem weiteren Fall so, dass Sie nicht bzw. nur einmal durchlaufen wird. Innerhalb der Schleife zeigen Sie den Wert der Laufvariablen an. Zusätzlich soll eine Variable die Summe der Laufvariablen beinhalten.

Wie kann man erreichen, dass die Schleife nie durchlaufen wird?

4.3 Definition von Laufvariablen

Member: PGM0403

Benutzen Sie das vorherige Programm. Definieren Sie die Laufvariablen sowie die Begrenzer in unterschiedlichen Programmversionen jeweils wie folgt:

- PIC S9(08).
- PIC S9(09) COMP-3.
- PIC S9(08) COMP.

Prüfen Sie die Laufzeiten der Jobs für die unterschiedlichen Programmversionen. Ab welchem Wert der Ende-Bedingung ist ein erkennbarer Laufzeitunterschied zu finden? Kann hier CPU gespart werden? Machen Sie eine grobe Hochrechnung, ob es Sinn macht, in Ihrer Firma „optimal“ zu programmieren.

4.4 Addition / Multiplikation

Member: PGM0404

Schreiben Sie ein Programm mit den folgenden Funktionen:

- Einlesen einer Zahl n
- Multiplikation der Zahl n mit sich selbst
- Potenzieren der Zahl mit 2
- n -fache Addition der Zahl mit sich selbst.
- Multiplikation mit der Zahl 2

Vergleichen Sie die Ergebnisse. Gibt es unterschiedliche Ergebnisse bei unterschiedlicher Anzahl der Nachkommastellen?

5. Verzweigungen / Bedingungen

5.1 einfacher IF

Member: PGM0501

Schreiben Sie ein Programm das eine Zahl mit 8 Stellen einliest. Ist diese Zahl größer als 5000, geben Sie eine Information aus. Die eingelesene Zahl geben Sie immer aus.

5.2 verschachtelter IF (1)

Member: PGM0502

Schreiben Sie ein Programm das eine Zahl mit 8 Stellen einliest. Ist diese Zahl größer als 5.000, geben Sie eine Information aus; sollte sie aber zusätzlich > 10.000 sein, geben Sie diese Information ebenfalls aus. Lösen Sie diese Aufgabe mit einem verschachtelten IF. Die eingelesene Zahl geben Sie immer aus.

5.3 verschachtelter IF (2)

Member: PGM0503

Schreiben Sie ein Programm das eine Zahl mit 8 Stellen einliest. Ist diese Zahl größer als 5.000, geben Sie eine Information aus; sollte sie aber zusätzlich > 10.000 sein, geben Sie diese Information ebenfalls aus. Sollte die Zahl aber kleiner oder gleich 5.000 sein, geben Sie dies als Information aus. Lösen Sie diese Aufgabe mit einem verschachtelten IF. Die eingelesene Zahl geben Sie immer aus.

5.4 Bedingungsnamen

Member: PGM0504

Lösen Sie die vorige Aufgabe mit Bedingungsnamen. Wählen Sie diese geschickt, so dass die Abfragen eindeutig sind.

5.5 EVALUATE

Member: PGM0505

Lösen Sie die vorige Aufgabe mit einem EVALUATE.

5.6 EVALUATE - vollständig

Member: PGM0506

Erweitern Sie den EVALUATE so, dass eine vollständige Prüfung auf alle Feldinhalte durchgeführt wird. Der OTHER-Zweig soll einen Fehlerhinweis ausgeben.

Achten Sie darauf, dass die Bedingungen des EVALUATE überschneidungsfrei formuliert sind, aber trotzdem die Anforderung erfüllt wird.

6. Schleifen mit Bedingungen

6.1 Lesen der Schulungsdatei

Member: PGM0601

Lesen Sie die Datei, die der Referent Ihnen zur Verfügung gestellt hat. Geben Sie die einzelnen Zeilen aus und zusätzlich die Anzahl der Zeilen. Die letzte Zeile hat in den Spalten 1-3 die Kennung '999'.

- Nutzen Sie den Befehl ACCEPT (SYSIN).
- Die Zeilen sind 80 Zeichen lang.

7. Sections

7.1 Lesen der Schulungsdatei

Member: PGM0701

Lesen Sie die Datei, die der Referent Ihnen zur Verfügung gestellt hat. Geben Sie die einzelnen Zeilen aus und zusätzlich die Anzahl der Zeilen. Den Lesevorgang lagern Sie in eine separate Section aus.

7.2 Lesen der Schulungsdatei – Tabelle

Member: PGM0702

Lesen Sie die Datei, die der Referent Ihnen zur Verfügung gestellt hat. Lesen Sie alle Zeilen in eine Tabelle und geben Sie diese Tabelle und zusätzlich die Anzahl der Zeilen am Ende aus.

7.3 Lesen der Schulungsdatei – Tabelle / Datei

Member: PGM0703

Umbau Accept auf Read / Display auf Write
DDSE01 ist Eingabe
DDSA01 ist Ausgabe (SYSOUT=*)

7.4 Lesen der Schulungsdatei – Gruppenwechsel

Member: PGM0704

Lesen Sie die Datei, die der Referent Ihnen zur Verfügung gestellt hat. Geben Sie die einzelnen Zeilen aus und zusätzlich die Anzahl der Zeilen. Prüfen Sie jeweils die erste Stelle der Eingabe-Datei; wenn sich die Spalte von Inhalt ändert, geben Sie '---' (80 Stellen) aus.

8. Tabellen

8.1 Definition einer Tabelle

Member: PGM0801

Definieren Sie eine Tabelle, die als Initialwerte die Monatsnamen enthält.

8.2 Initialisieren einer Tabelle

Member: PGM0802

Definieren Sie eine 2-dimensionale Tabelle aus 52 Wochen und je 7 Tagen. Die Tage haben 4 Vorkomma- und 2 Nachkommastellen. Initialisieren Sie diese Tabelle auf verschiedene Wege auf 0.