Einführung in Programmierung mit ABAP:

Teil 8: Modularisierung mit FORM und Funktionsbaustein

Prof. Dr. Peter Hohmann

Technische Hochschule Mittelhessen

FB MNI

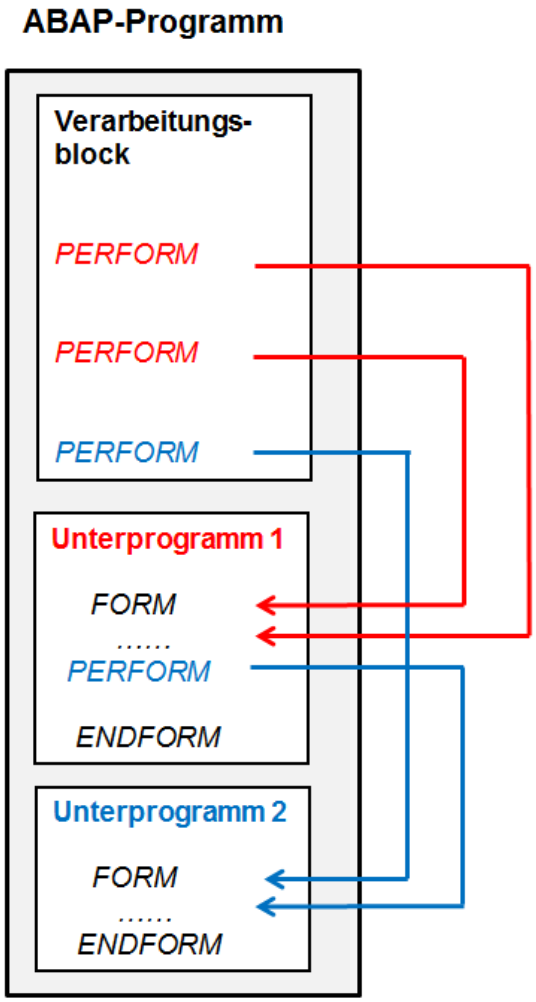
www.prof-dr-hohmann.de

SS 2017

Vorteile der Modularisierung

- Die Funktionalität kann mehrfach im Programm genutzt werden.
- Die Funktionalität kann auch von anderen Objekten genutzt werden.
- Das Programm wird lesbarer/übersichtlicher und ist besser wartbar.
- Änderungen der Funktionalität sind direkt in allen Objekten vorhanden.

Modularisierung mit Unterprogrammen PERFOM/FORM



Modularisierung (Abbildung aus SAP®-Hilfe)

Modularisierung mit Unterprogrammen
PERFOM / FORM

PERFORM = Aufruf von Unterroutinen mit Variablenübergabe mit **USING** oder **CHANGING.**

```
PERFORM  XYZ USING feld1.  "Aktualparameter
```

FORM = Aufruf von Unterroutine mit Variablenübergabe mit **USING** oder **CHANGING**

```
FORM XYZ USING feld_a.   "Formalparameter
```

Modularisierung mit Unterprogrammen PERFOM / FORM

CALL-BY-REFERENZ

Dem Unterprogramm wird die aktuelle Adresse des Aktualparameters übergeben. Das bedeutet, dass sich Wertzuweisungen an den Formalparameter direkt auf den Aktualparameter auswirken. Haupt- und Unterprogramm verwenden den gleichen Speicherbereich. Eignung für den Import und Export von Daten aus dem Hauptprogramm in das Unterprogramm und zurück. Call-by-Referenz wird erzielt, indem man einen Übergabeparameter unter CHANGING oder USING aufführt. CHANGING und USING haben die gleiche Bedeutung.

```
PERFORM unterprogramm CHANGING  a1.
FORM    unterprogramm CHANCING b1.
```

oder mit gleicher Bedeutung

```
PERFORM unterprogramm USING  a1.
FORM    unterprogramm USING b1.
```

Modularisierung mit Unterprogrammen
PERFOM / FORM

CALL-BY-VALUE

Dem Unterprogramm soll eine lokale Kopie des Aktualparameters übergeben werden. Das bedeutet, dass sich die Wertzuweisungen an den Formalparameter überhaupt nicht auf den Aktualparameter auswirken. Eignung nur für den Import von Daten aus dem Hauptprogramm in das Untermodul. Call-by-Value wird erzielt, indem man einen solche Variable mit USING parameter übergibt und im Unterprogramm mit USING VALUE (parameter) empfängt.

PERFORM unterprogramm **USING** a1.
FROM unterprogramm **USING VALUE** (b1).

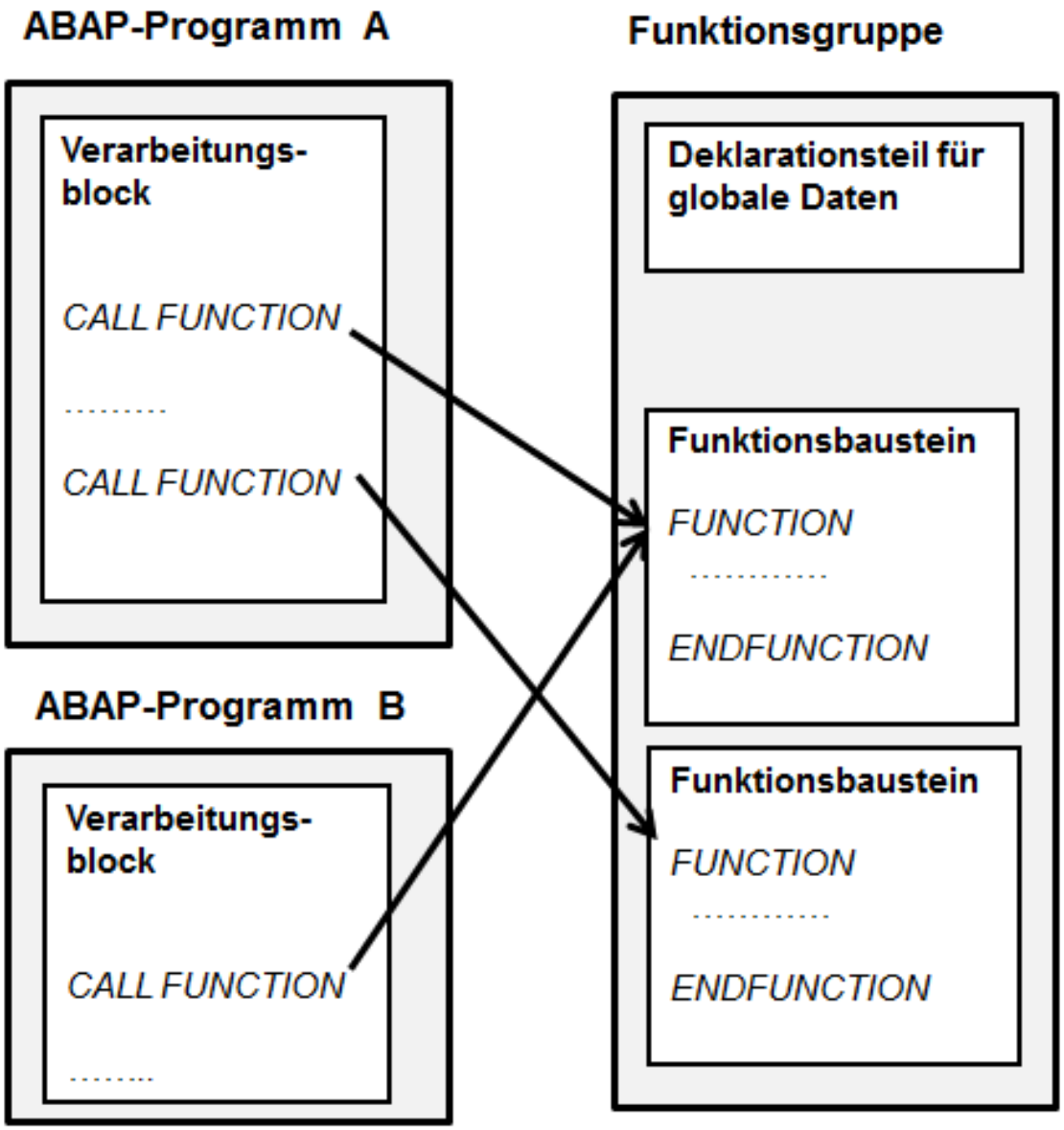
Modularisierung mit Unterprogrammen
PERFOM / FORM

CALL-BY-VALUE-AND-RESULT

Dem Unterprogramm soll eine lokale Kopie des Aktualparameters mit Wertrückgabe übergeben werden. Das bedeutet, dass sich Wertzuweisungen an den Formalparameter erst beim Verlassen des Unterprogramms auf den Aktualparameter auswirken. Diese Übergabe sollten Sie wählen, wenn Sie sicherstellen wollen, dass bei einem vorzeitigen Abbruch des Unterprogramms die Aktualparameter noch nicht verändert worden sind.

```
PERFORM unterprogramm CHANGING a1.
FORM    unterprogramm CHANGING VALUE (b1).
```

Modularisierung mit Funktionsbausteinen



Modularisierung (Abbildung aus SAP®-Hilfe)

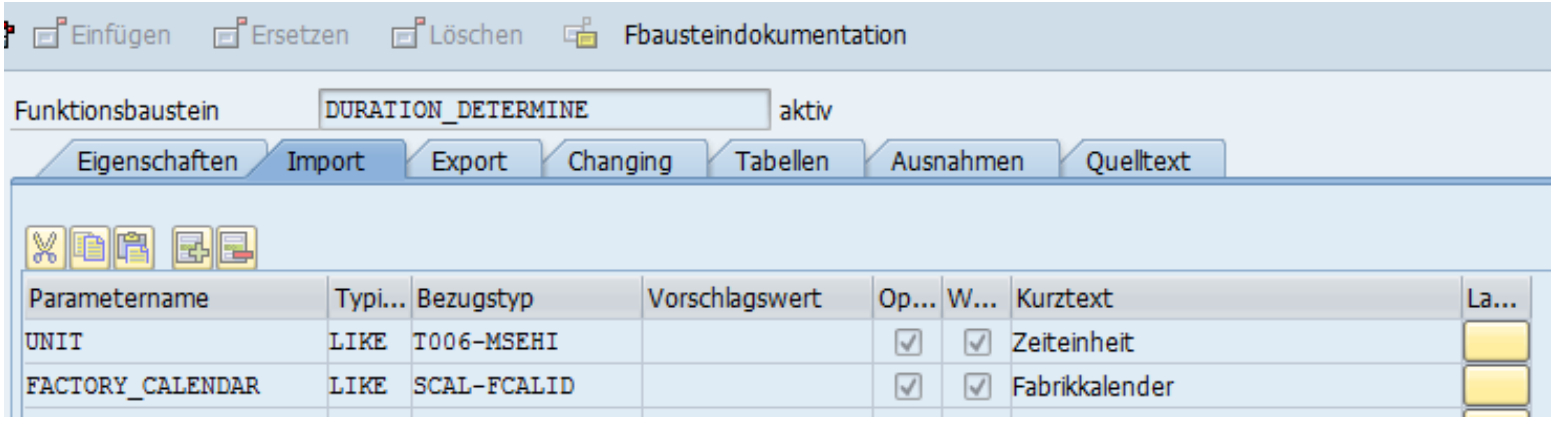
Modularisierung mit Funktionsbausteinen - Funktionsbausteine -

- Funktionsbausteine sind Unterprogramme in ABAP
- Funktionsbausteine werden im Repository zentral verwaltet
- Funktionsbausteine verfügen über Schnittstellen zum Im- und Export von Daten
- Funktionsbeine sind wiederverwendbar
- Schnittstellenparameter werden mit globalen Typen typisiert

Modularisierung mit Funktionsbausteinen - Funktionsgruppen -

- Funktionsbausteine sind Funktionsgruppen zugeordnet.
- Funktionsgruppen fassen mehrere Funktionsbausteine logisch zusammen
- Funktionsgruppen enthalten Funktionsbausteine mit verwandter Funktionalität
- Werden globale Datenobjekte der Funktionsgruppe mit Werten gefüllt, stehen sie anderen Funktionsbausteinen derselben Gruppe zur Verfügung
- Wenn ein Funktionsbaustein einer Funktionsgruppe aufgerufen wird, so wird die gesamte Funktionsgruppe geladen.

Modularisierung mit Funktionsbausteinen



- **Import-Parameter:** Beim Aufruf werden dem Funktionsbaustein Werte oder Variablen übergeben. Optionale Parameter müssen nicht übergeben werden.
- **Export-Parameter:** Mit den Export-Parametern übergibt der Funktionsbaustein Daten an die Empfangsvariablen des aufrufenden Programms. Export-Parameter sind optional.
- **Changing-Parameter :** An Changing-Parameter werden Variable des rufenden Programms übergeben, die durch den Funktionsbaustein geändert werden und an des aufrufende Programm zurückgegeben werden.
- **Tabellen:** Übergabe von internen Tabellen.
- **Ausnahmen :**Sie geben Auskunft über Fehlersituationen im Funktionsbaustein. Ausnahmen werden vom aufrufenden Programm behandelt.

Modularisierung mit Funktionsbausteinen - Beispiel Funktionsbaustein -

```
CALL FUNCTION 'DURATION_DETERMINE'
    EXPORTING
        UNIT                  = 'MIN'
    IMPORTING
        DURATION              = GV_DURATION
    CHANGING
        START_DATE             = PA_SDATE
        START_TIME             = PA_STIME
        END_DATE               = PA_EDATE
        END_TIME               = PA_ETIME
    EXCEPTIONS
        FACTORY_CALENDAR_NOT_FOUND = 1
        DATE_OUT_OF_CALENDAR_RANGE = 2
        DATE_NOT_VALID         = 3
        UNIT_CONVERSION_ERROR  = 4
        SI_UNIT_MISSING        = 5
        PARAMETERS_NOT_VALID   = 6
        OTHERS                 = 7.
```

- Im Exporting-Block werden die Import-Parameter übergeben.
- Im Importing-Block werden die Export-Parameter übergeben.