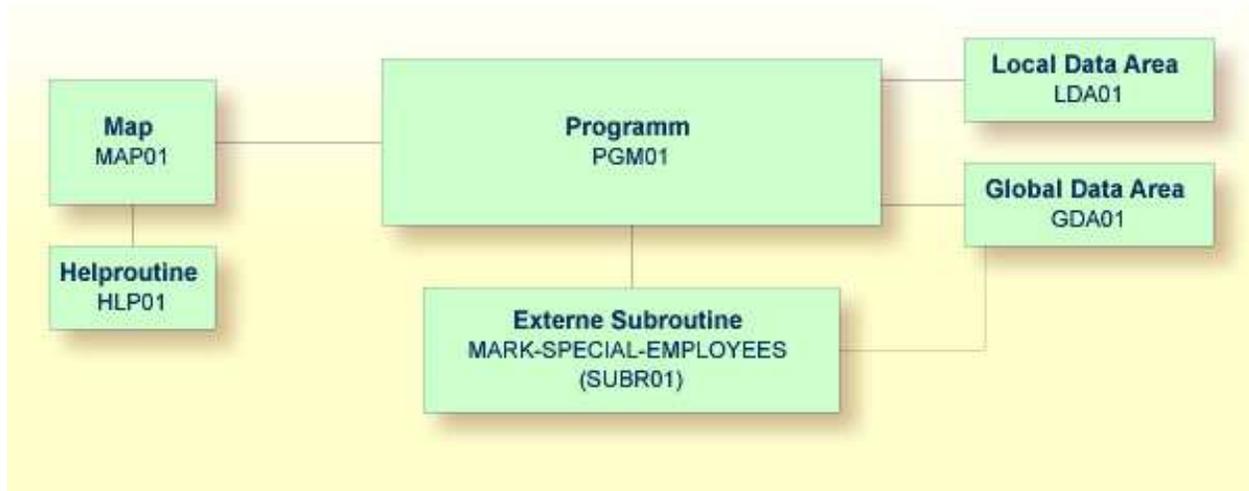


Externe Subroutinen

Bis jetzt ist die Subroutine `MARK-SPECIAL-EMPLOYEES` mit einem `DEFINE SUBROUTINE`-Statement im Programm selbst definiert. Sie werden diese Subroutine jetzt als separates Objekt definieren, das sich außerhalb des Programms befindet.

Wenn Sie mit den Übungen in diesem Kapitel fertig sind, wird Ihre Beispielanwendung aus den folgenden Modulen bestehen:



Dieses Kapitel enthält die folgenden Übungen:

- Externe Subroutine erstellen
- Externe Subroutine aus dem Programm aufrufen

Externe Subroutine erstellen

Da der Code, der zurzeit im Programm definiert ist, in der externen Subroutine wiederverwendet werden soll, werden Sie das Programm jetzt unter einem anderen Namen speichern, den Typ in Subroutine ändern und alle nicht benötigten Zeilen löschen.

Das `DEFINE SUBROUTINE`-Statement wird in der externen Subroutine genauso kodiert wie in der internen Subroutine des Programms.

▶ Externe Subroutine erstellen

1. Geben Sie Folgendes in der Kommandozeile des Programmeditors ein:

```
SA SUBR01
```

Das aktuelle Programm wird unter dem neuen Namen SUBR01 gespeichert. Das Programm wird noch immer im Editor angezeigt.

2. Laden Sie das neu erstellte Objekt mit dem folgenden Kommando in den Editor:

```
E SUBR01
```

Der Objekttyp ist noch immer Programm.

3. Geben Sie das folgende Kommando ein, um das Programm in eine externe Subroutine zu ändern:

```
SET TYPE S
```

wobei "S" für Subroutine steht.

Der im Bildschirm angezeigte Objekttyp ändert sich in "Subroutine".

4. Benutzen Sie das Zeilenkommando .D, um alle Zeilen außer den Folgenden zu löschen:

```
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL USING LDA01
END-DEFINE
*
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
*
END
```

Sie können auch einen kompletten Textblock löschen. Hierzu gehen Sie folgendermaßen vor:

1. Geben am Anfang des Textblocks das Zeilenkommando .X ein.
2. Geben Sie am Anfang der letzten Zeile des Textblocks das Zeilenkommando .Y ein.
3. Drücken Sie EINGABE.

Der zu löschende Textblock ist jetzt mit "X" und "Y" markiert. (Wenn Sie diese Markierungen entfernen möchten, geben Sie RESET in der Kommandozeile ein).

4. Geben Sie DX-Y in der Kommandozeile ein, um den markierten Textblock zu löschen.
5. Speichern Sie die Subroutine mit STOW.

Externe Subroutine aus dem Programm aufrufen

Mit dem PERFORM-Statement kann man sowohl interne als auch externe Subroutinen aufrufen. Wenn im Programm keine interne Subroutine gefunden wird, versucht Natural automatisch, eine externe Subroutine mit demselben Namen auszuführen. Natural sucht hierbei nach dem Namen, der im Subroutinencode definiert wurde (d.h. dem Subroutinennamen). Natural sucht nicht nach dem Namen, den Sie beim

Speichern der Subroutine angegeben haben (d.h. dem Objektnamen).

Jetzt, nachdem Sie eine externe Subroutine erstellt haben, müssen Sie die interne Subroutine (die denselben Namen hat wie die externe Subroutine) aus Ihrem Programm entfernen.

▶ Externe Subroutine in Ihrem Programm benutzen

1. Kehren Sie zum Programmeditor zurück, indem Sie Folgendes in der Kommandozeile des Editors eingeben, in dem die Subroutine zurzeit angezeigt wird.

```
E PGM01
```

2. Entfernen Sie die folgenden Zeilen:

```
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
```

Ihr Programm sollte nun folgendermaßen aussehen:

```
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL USING LDA01
END-DEFINE
*
RP1. REPEAT
*
  INPUT USING MAP 'MAP01'
*
  IF #NAME-START = '.' THEN
    ESCAPE BOTTOM (RP1.)
  END-IF
*
  IF #NAME-END = ' ' THEN
    MOVE #NAME-START TO #NAME-END
  END-IF
*
  RD1. READ EMPLOYEES-VIEW BY NAME
    STARTING FROM #NAME-START
    ENDING AT #NAME-END
*
    IF LEAVE-DUE >= 20 THEN
      PERFORM MARK-SPECIAL-EMPLOYEES
    ELSE
      RESET #MARK
    END-IF
*
    DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK

  END-READ
*
  IF *COUNTER (RD1.) = 0 THEN
    REINPUT 'No employees meet your criteria.'
  END-IF
*
END-REPEAT
*
END
```

3. Führen Sie das Programm mit RUN aus.
4. Geben Sie "JONES" als Startname ein und drücken Sie EINGABE.

Die daraufhin erscheinende Liste sollte immer noch einen Stern bei jedem Mitarbeiter anzeigen, der 20 oder mehr Urlaubstage hat.

5. Geben Sie EDIT in der MORE-Zeile ein, um zum Programmeditor zurückzukehren.
6. Speichern Sie das Programm mit STOW.

▶ Identische Subroutinennamen auflisten

1. Geben Sie eines der folgenden Kommandos in der Kommandozeile des Programmeditors ein:

```
LIST EXTENDED SUBROUTINE *
```

```
L EXT S *
```

Der folgende Bildschirm erscheint. In ihm werden alle externen Subroutinenobjekte (Member) mit den entsprechenden Langnamen aufgelistet, die in der aktuellen Natural-Library und Systemdatei enthalten sind.

```
12:21:09          ***** NATURAL LIST COMMAND *****          2007-03-20
User SAG          - LIST Objects in a Library -          Library TUTORIAL

Cmd Subroutine/Class Name          Type S/C Member          Cat Date          Cat Time
--- * _____          S_____ --- * _____          * _____          * _____
___ MARK-SPECIAL-EMPLOYEES          Subro S/C SUBR01          2007-03-20          12:11:56

                                                                    1 Objects found

Top of List.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit Sort          -- - + ++          > Canc
```

Wenn Sie nur Subroutinennamen mit bestimmten Anfangsbuchstaben anzeigen möchten, können Sie den entsprechenden Suchwert gefolgt von einem Stern (*) in dem Feld unter der Überschrift **Subroutine/Class Name** oder unter der Überschrift **Member** angeben. Beispiel:

Cmd	Subroutine/Class Name	Type	S/C	Member	Cat Date	Cat Time
---	MARK*	S	---	*	*	*
__	MARK-SPECIAL-EMPLOYEES	Subro	S/C	SUBR01	2007-03-20	12:11:56

2. Drücken Sie PF3, um zum Programmeditor zurückzukehren.

Sie können nun mit den nächsten Übungen fortfahren: *Subprogramme*.