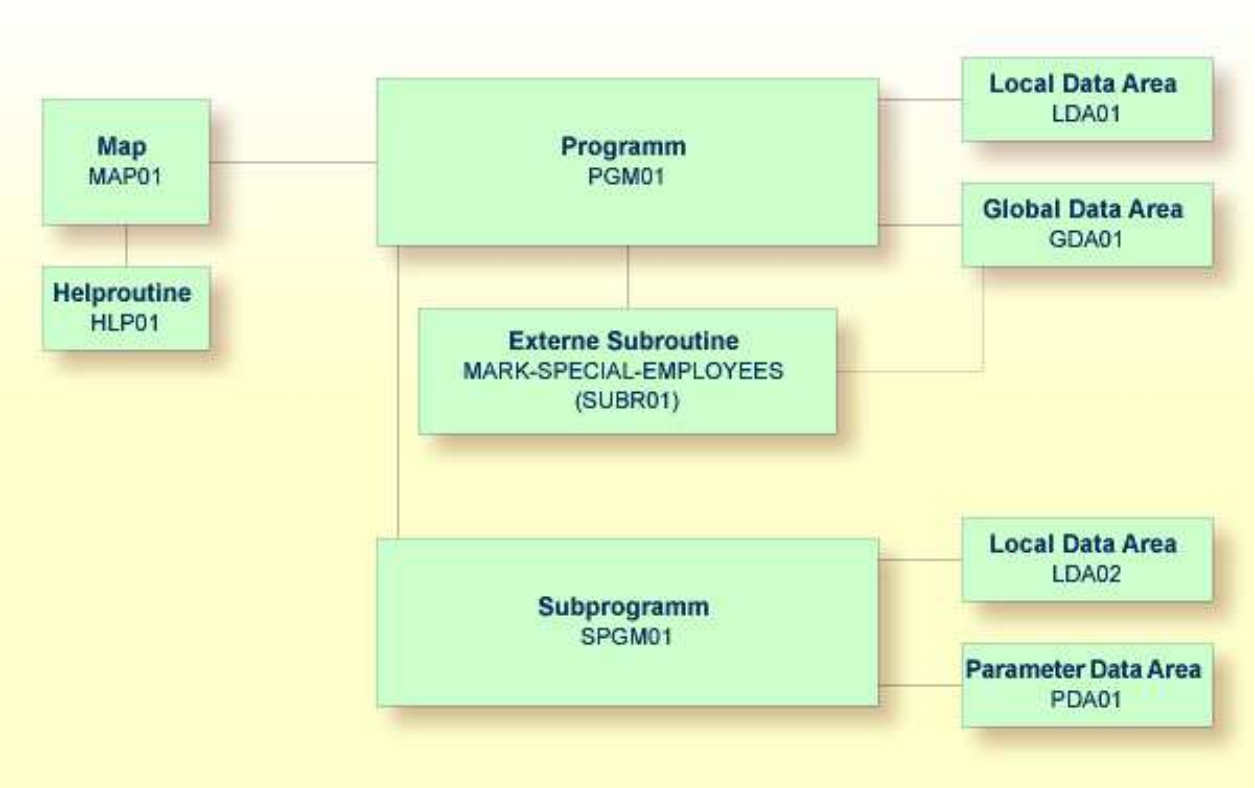
Subprogramme

Sie werden Ihr Programm jetzt durch ein `CALLNAT`-Statement erweitern, mit dem ein Subprogramm aufgerufen wird. In dem Subprogramm bilden die Mitarbeiter, die vom Hauptprogramm gefunden wurden, die Grundlage für eine `FIND`-Anfrage in der `VEHICLES`-Datei. Diese Datei ist ebenfalls Bestandteil der Demodatenbank. In der Ausgabe werden dann Fahrzeuginformationen aus dem Subprogramm und Mitarbeiterinformationen aus dem Hauptprogramm zu sehen sein.

Das neue Subprogramm benötigt eine weitere Local Data Area und eine Parameter Data Area.

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



Dieses Kapitel enthält die folgenden Übungen:

- Local Data Area ändern
- Parameter Data Area mit Hilfe einer bestehenden Local Data Area erstellen
- Eine zweite Local Data Area mit einem anderen View erstellen
- Subprogramm erstellen
- Subprogramm aus dem Programm aufrufen

Local Data Area ändern

Sie werden jetzt weitere Felder in die Local Data Area einfügen, die Sie zuvor erstellt haben. Diese Felder werden von dem Subprogramm benutzt, das Sie später erstellen werden.

Weitere Felder in die Local Data Area einfügen

1. Kehren Sie zur Local Data Area zurück.

```
E LDA01
```

2. Definieren Sie die folgenden Felder unter #NAME-END:

Level (Spalte L)	Name	Format (Spalte F)	Länge
1	#PERS-ID	A	8
1	#MAKE	A	20
1	#MODEL	A	20

Die Local Data Area sollte nun folgendermaßen aussehen:

```
Local     LDA01     Library TUTORIAL                      DBID 11177 FNR     8
Command                                                                  > +
I T L  Name                            F Length     Miscellaneous
All -- --------------------------------- - ---------- ------------------------>
     1 #NAME-START                     A         20
     1 #NAME-END                       A         20
     1 #PERS-ID                        A          8
     1 #MAKE                           A         20
     1 #MODEL                          A         20
```

3. Speichern Sie die Local Data Area mit STOW.

Parameter Data Area mit Hilfe einer bestehenden Local Data Area erstellen

Eine Parameter Data Area (PDA) wird zur Angabe der Datenparameter benutzt, die zwischen Ihrem Natural-Programm und dem Subprogramm (das Sie später noch erstellen werden) ausgetauscht werden sollen. Die Parameter Data Area wird im Subprogramm referenziert.

Mit kleinen Änderungen kann Ihre Local Data Area zur Erstellung der Parameter Data Area benutzt werden: Sie werden zwei Datenfelder in der Local Data Area löschen und die so veränderte Local Data Area als Parameter Data Area speichern. Die ursprüngliche Local Data Area wird hierdurch nicht verändert.

Parameter Data Area erstellen

1. Löschen Sie die Felder `#NAME-START` und `#NAME-END` in der Local Data Area.
2. Geben Sie Folgendes in der Kommandozeile des Data-Area-Editors ein.

   ```
   SA PDA01
   ```

   Die aktuelle Data Area wird unter dem neuen Namen `PDA01` gespeichert. Die bestehende Local Data Area wird noch immer im Editor angezeigt.
3. Laden Sie die neu erstellte Data Area mit dem folgenden Kommando in den Editor:

   ```
   E PDA01
   ```
4. Geben Sie das folgende Kommando ein, um die Local Data Area in eine Parameter Data Area zu ändern:

   ```
   SET TYPE A
   ```

   wobei "A" für Parameter Data Area steht.

   Der Objekttyp ändert sich in "Parameter". Dies wird oben links im Bildschirm angezeigt. Die Parameter Data Area sollte nun folgendermaßen aussehen:

```
Parameter PDA01     Library TUTORIAL                      DBID 11177 FNR     8
Command                                                                  > +
I T L  Name                            F Length     Miscellaneous
All -- ------------------------------- - ---------- ------------------------>
     1 #PERS-ID                        A          8
     1 #MAKE                           A         20
     1 #MODEL                          A         20
```

5. Speichern Sie die Parameter Data Area mit `STOW`.

Eine zweite Local Data Area mit einem anderen View erstellen

Sie werden jetzt eine zweite Local Data Area erstellen und Felder aus dem DDM für die Datenbankdatei `VEHICLES` importieren.

Diese Local Data Area wird dann in Ihrem Subprogramm referenziert.

Local Data Area erstellen

1. Geben Sie das folgende Kommando in der Kommandozeile des Data-Area-Editors ein.

   ```
   CLEAR
   ```

   Der Data-Area-Editor ist jetzt leer.

2. Geben Sie das Folgendes in der Kommandozeile ein, um den Typ der Data Area zu ändern:

   ```
   SET TYPE L
   ```

   wobei "L" für Local Data Area steht.

3. Geben Sie Folgendes in der ersten Zeile des Editierbereiches ein, beginnend in der Spalte **T**:

```
.V(VEHICLES)
```

4. Drücken Sie EINGABE.

   Der VEHICLES-View erscheint.

```
SYSGDA 4461: Mark fields to incorporate into data area.
Local              Library TUTORIAL                      DBID 11177 FNR     8
View VEHICLES
I T L  Name                             F Length     Miscellaneous
- - -- -------------------------------- - ---------- ------------------------
     2 REG-NUM                          A         15 /* CAR'S REGISTR. NUMBE
     2 CHASSIS-NUM                      I          4 /* MANUFACTURER NUMBER
     2 PERSONNEL-ID                     A          8 /* IDENT. OF CAR USER
  G  2 CAR-DETAILS                                 /* DESCRIPTION OF THE C
     3 MAKE                             A         20
     3 MODEL                            A         20
     3 COLOR                            A         10
     3 COLOUR                           A         10
     2 YEAR                             N        4.0 /* MANUFACTURING YEAR
     2 CLASS                            A          1 /* P=PRIVAT
     2 LEASE-PUR                        A          1 /* L=LEASED
     2 DATE-ACQ                         N        8.0 /* DATE THE CAR WAS ACQ
     2 CURR-CODE                        A          3 /* CURRENCY OF CAR COST
  M  2 MAINT-COST                       P        7.0 (1:60)/* MAINTENANCE COST
     2 MODEL-YEAR-MAKE                  A         24 /* YEAR + CAR MAKE /* SP


- - -- -------------------------------- - ---------- ------------------------
```

5. Markieren Sie die folgenden Felder, indem Sie ein beliebiges Zeichen in der Spalte **I** eingeben:

   PERSONNEL-ID
   CAR-DETAILS
   MAKE
   MODEL

6. Nachdem Sie alle erforderlichen Felder markiert haben, drücken Sie EINGABE, um zum Data-Area-Editor zurückzukehren.

   Die Local Data Area sollte nun folgendermaßen aussehen:

```
Local              Library TUTORIAL                      DBID 11177 FNR     8
Command                                                                  > +
I T L  Name                            F Length     Miscellaneous
All -- ------------------------------- - ---------- ------------------------>
  V  1 VEHICLES-VIEW                                  VEHICLES
     2 PERSONNEL-ID                    A          8 /* IDENT. OF CAR USER
  G  2 CAR-DETAILS                                    /* DESCRIPTION OF THE CAR
     3 MAKE                            A         20
     3 MODEL                           A         20
                                                                    S 5    L 1
```

7. Geben Sie Folgendes in der Kommandozeile ein, um die neue Local Data Area zu speichern:

```
SA LDA02
```

8. Speichern Sie die neue Local Data Area mit `STOW`.

Subprogramm erstellen

Sie werden jetzt ein Subprogramm erstellen, das mit Hilfe einer Parameter Data Area und einer Local Data Area Informationen aus der `VEHICLES`-Datei abruft. Das Programm `PGM01` übergibt das Personalkennzeichen (`PERSONNEL-ID`) an das Subprogramm. Das Subprogramm benutzt dieses Kennzeichen für die Suche in der `VEHICLES`-Datei.

Subprogramm erstellen

1. Geben Sie das folgende Kommando in der Kommandozeile des Data-Area-Editors ein.

```
E N
```

   wobei "N" für Subprogramm steht.

   Ein leerer Programmeditor wird aufgerufen. Der Objekttyp ist auf Subprogramm gesetzt.

2. Geben Sie Folgendes ein:

```
DEFINE DATA
  PARAMETER USING PDA01
  LOCAL USING LDA02
END-DEFINE
```

```
*
FD1. FIND (1) VEHICLES-VIEW
  WITH PERSONNEL-ID = #PERS-ID
  MOVE MAKE (FD1.) TO #MAKE
  MOVE MODEL (FD1.) TO #MODEL
  ESCAPE BOTTOM
END-FIND
*
END
```

Dieses Subprogramm gibt die folgenden Informationen an ein bestimmtes Personalkennzeichen zurück: die Marke und das Modell des Firmenfahrzeuges eines Mitarbeiters.

Auf Grund des Suchkriteriums #PERS-ID wählt das FIND-Statement eine Reihe von Datensätzen (hier: einen Datensatz) aus der Datenbank aus.

Das Feld #PERS-ID des Subprogramms erhält den Wert von PERSONNEL-ID, der vom Programm PGM01 übergeben wurde. Das Subprogramm benutzt diesen Wert für die Suche in der VEHICLES-Datei.

3. Speichern Sie das Subprogramm mit STOW.

```
STOW SPGM01
```

Subprogramm aus dem Programm aufrufen

Ein Subprogramm wird mit einem CALLNAT-Statement aus dem Hauptprogramm aufgerufen. Ein Subprogramm kann nur mit einem CALLNAT-Statement aufgerufen werden; es kann selbst nicht ausgeführt werden. Ein Subprogramm kann nicht auf die Global Data Area zugreifen, die von dem aufrufenden Objekt benutzt wird.

Die Daten werden vom Hauptprogramm an das Subprogramm mit Hilfe von Parametern übergeben, die im Subprogramm mit einem DEFINE DATA PARAMETER-Statement referenziert werden.

Die Variablen, die in der Parameter Data Area des Subprogramms definiert sind, müssen nicht unbedingt dieselben Namen haben wie die Variablen im CALLNAT-Statement. Sie müssen nur in der Reihenfolge, im Format und in der Länge übereinstimmen.

Sie werden jetzt Ihr Hauptprogramm ändern, damit es das eben von Ihnen erstellte Subprogramm benutzen kann.

Subprogramm in Ihrem Hauptprogramm benutzen

1. Kehren Sie zum Programmeditor zurück, indem Sie Folgendes in der Kommandozeile eingeben.

```
E PGM01
```

2. Geben Sie Folgendes direkt über dem DISPLAY-Statement ein:

```
RESET #MAKE #MODEL
CALLNAT 'SPGM01' PERSONNEL-ID #MAKE #MODEL
```

Das RESET-Statement setzt die Werte für #MAKE und #MODEL auf Nullwerte.

3. Löschen Sie die Zeile, die das DISPLAY-Statement enthält, und ersetzen Sie sie mit Folgendem:

```
WRITE TITLE
      / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***'
      / '***     ARE MARKED WITH AN ASTERISK        ***'//
*
DISPLAY   1X '//N A M E' NAME
          1X '//DEPT'    DEPT
          1X '/LV/DUE'   LEAVE-DUE
             ' '         #MARK
          1X '//MAKE'    #MAKE
          1X '//MODEL'   #MODEL
```

Der mit dem WRITE TITLE-Statement definierte Text wird bei der Ausgabe oben auf jeder Seite angezeigt. Das WRITE TITLE-Statement setzt den Standardseitentitel außer Kraft: die Informationen, die bisher oben auf jeder Seite angezeigt wurden (Seitenzahl, Datum und Uhrzeit) werden nicht mehr angezeigt. Jeder Schrägstrich (/) sorgt dafür, dass die nachfolgenden Informationen in einer neuen Zeile angezeigt werden.

Da das Subprogramm jetzt zusätzliche Fahrzeuginformationen ausgibt, muss die Größe der Spalten in der Ausgabe angepasst werden. Die Spalten erhalten kürzere Überschriften. Die Spalte, in der der Stern angezeigt wird (#MARK), bekommt überhaupt keine Überschrift. Zwischen den einzelnen Spalten wird je ein Leerzeichen eingefügt (1X). Jeder Schrägstrich in einer Spaltenüberschrift lässt die nachfolgenden Informationen in derselben Spalte in einer neuen Zeile erscheinen.

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
*
```

```
    RESET #MAKE #MODEL
    CALLNAT 'SPGM01' PERSONNEL-ID #MAKE #MODEL
*
    WRITE TITLE
      / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***'
      / '***     ARE MARKED WITH AN ASTERISK        ***'//
*
    DISPLAY   1X '//N A M E' NAME
              1X '//DEPT'    DEPT
              1X '/LV/DUE'   LEAVE-DUE
                 ' '         #MARK
              1X '//MAKE'    #MAKE
              1X '//MODEL'   #MODEL
*
  END-READ
*
  IF *COUNTER (RD1.) = 0 THEN
    REINPUT 'No employees meet your criteria.'
  END-IF
*
END-REPEAT
*
END
```

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

   Die daraufhin erscheinende Liste sollte der folgenden Ausgabe ähneln:

```
MORE

                *** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***
                ***     ARE MARKED WITH AN ASTERISK        ***


                           LV
      N A M E         DEPT  DUE           MAKE                MODEL
 -------------------- ------ --- - -------------------- --------------------

 JONES                SALE30  25 * CHRYSLER             IMPERIAL
 JONES                MGMT10  34 * CHRYSLER             PLYMOUTH
 JONES                TECH10  11   GENERAL MOTORS       CHEVROLET
 JONES                MGMT10  18   FORD                 ESCORT
 JONES                TECH10  21 * GENERAL MOTORS       BUICK
 JONES                SALE00  30 * GENERAL MOTORS       PONTIAC
 JONES                SALE20  14   GENERAL MOTORS       OLDSMOBILE
 JONES                COMP12  26 * DATSUN               SUNNY
 JONES                TECH02  25 * FORD                 ESCORT 1.3
```

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

Sie haben das Tutorial jetzt erfolgreich abgeschlossen.