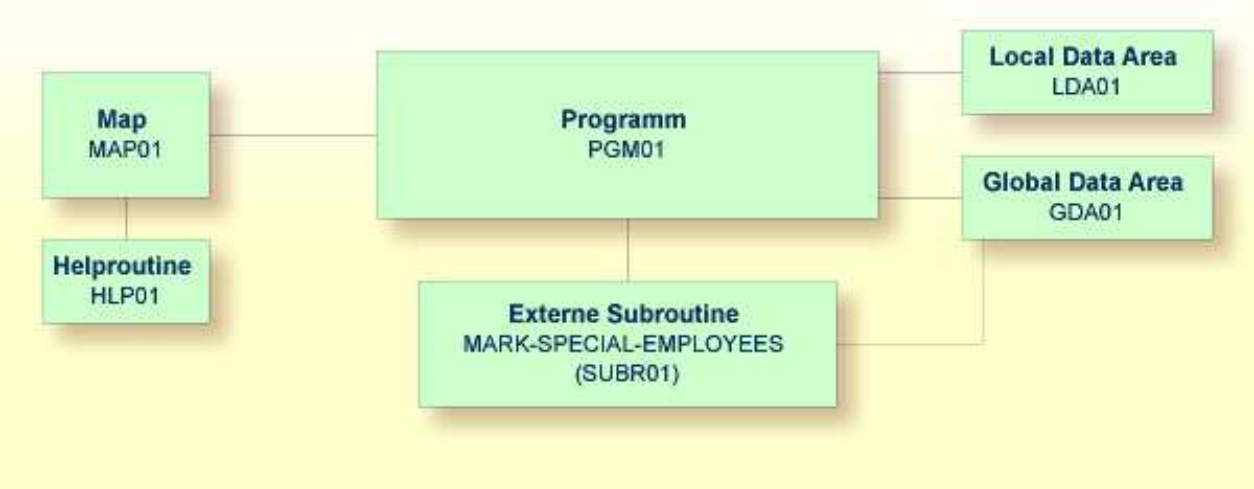
Externe Subroutinen

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

Sie werden jetzt einen Editor aufrufen, in dem Sie den Code für die externe Subroutine eingeben.

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

Externe Subroutine erstellen

1. Markieren Sie im Library-Workspace die Library, die auch Ihr Programm enthält (d.h. markieren Sie den Knoten **TUTORIAL**).
2. Wählen Sie aus dem Kontextmenü den Befehl **New Source > Subroutine**.

   Ein leeres Editorfenster erscheint.
3. Geben Sie Folgendes ein:

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

4. Speichern Sie die Subroutine mit `STOW`.

   Das Dialogfeld **Stow As** erscheint.

5. Geben Sie "SUBR01" als Name für die externe Subroutine ein und wählen Sie die Befehlsschaltfläche **OK**.

   Im Library-Workspace wird die neue externe Subroutine im Knoten **Subroutines** angezeigt. Im Logical-View wird der Name der Subroutine so angezeigt wie im Code: `MARK-SPECIAL-EMPLOYEES`. In den anderen Views wird der Name `SUBR01` angezeigt.

6. Schließen Sie das Editorfenster, in dem Sie die externe Subroutine eingegeben haben.

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.
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. Drücken Sie ESC, um das Ausgabefenster zu schließen.
6. Speichern Sie das Programm mit `STOW`.

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