# Anhang

## A. Programmquelltexte

```
#*****************************************************************************************#
#       MODUL:      globals.4gl                                                               #
#                                                                                             #
#       BESCHREIBUNG : Die globalen Variablen werden definiert.                               #
#                                                                                             #
#       DATUM :         18.11.1994                                                            #
#       AUTOR : Jinsook Kim                                                                   #
#                                                                                             #
#*****************************************************************************************#

DATABASE biblio

GLOBALS

DEFINE
        id CHAR(10),                    # Benutzerkennung, die bei der Identifizierungsprozess eingegeben wird.

        p_buch RECORD LIKE buch.*,      # Programmvariablen : buch

        p_benutzer RECORD               # Programmvariablen : benutzer
                benutzernr      LIKE benutzer.benutzernr,
                nachname        LIKE benutzer.nachname,
                vorname         LIKE benutzer.vorname,
                adresse         LIKE benutzer.adresse,
                plz             LIKE benutzer.plz,
                ort             LIKE ort.ort
        END RECORD,

        p_des ARRAY[3] OF RECORD        # Programmvariablen : desname
                desname           LIKE desdef.desname
        END RECORD,

        pa_benutzernr   LIKE ausleihe.benutzernr,       # Programmvariable : ausleihe
        p_aus1 ARRAY[10] OF RECORD                      # Um die Daten einzugeben
                signatur        LIKE ausleihe.signatur,
                ausleihdatum    LIKE ausleihe.ausleihdatum,
                rueckgabedatum  LIKE ausleihe.rueckgabedatum
        END RECORD,

        p_aus2 RECORD                                   # Programmvariable : ausleihe
                benutzernr      LIKE ausleihe.benutzernr,  # Um die Daten zu zeigen
                signatur        LIKE ausleihe.signatur,
                titel           LIKE buch.titel,
                ausleihdatum    LIKE ausleihe.ausleihdatum,
                rueckgabedatum  LIKE ausleihe.rueckgabedatum,
                saeumnisgebuehr INTERVAL DAY(4) TO DAY
        END RECORD,

        p_rueck RECORD                          # Programmvariable : rueckgabe
                benutzernr      LIKE ausleihe.benutzernr,
                signatur        LIKE ausleihe.signatur,
                titel           LIKE buch.titel,
                ausleihdatum    LIKE ausleihe.ausleihdatum,
```

```
            rueckgabedatum  LIKE ausleihe.rueckgabedatum,
            saeumnisgebuehr INTERVAL DAY(4) TO DAY
        END RECORD,

        p_res1 ARRAY[10] OF RECORD      # Programmvariable : reservierung
            benutzernr      LIKE reservierung.benutzernr,     # Um die Daten anzuzeigen
            signatur        LIKE reservierung.signatur,
            res_datum       LIKE reservierung.res_datum
        END RECORD,

        pv_benutzernr   LIKE reservierung.benutzernr,     #Programmvariable:reservierung
        p_res2 ARRAY[10] OF RECORD                        # Um die Daten einzugeben
            signatur        LIKE reservierung.signatur,
            res_datum       LIKE reservierung.res_datum
        END RECORD,

        p_bk RECORD                                       # Fuer die Benutzerdaten in Konto
            benutzernr      LIKE benutzer.benutzernr,
            nachname        LIKE benutzer.nachname,
            vorname         LIKE benutzer.vorname,
            adresse         LIKE benutzer.adresse,
            plz             LIKE benutzer.plz,
            ort             LIKE ort.ort
        END RECORD,

        p_ben_konto ARRAY[10] OF RECORD                   # Die Dokumente, die jeweiliger Benutzer
            signatur        LIKE ausleihe.signatur,   # ausgeliehen haben.
            ausleihdatum    LIKE ausleihe.ausleihdatum,
            rueckgabedatum  LIKE ausleihe.rueckgabedatum,
            saeumnisgebuehr INTERVAL DAY(4) TO DAY
        END RECORD,

        p_res_zurueck RECORD            # fuer die Zuruecknahme der Reservierung
            benutzernr      LIKE reservierung.benutzernr,
            signatur        LIKE reservierung.signatur,
            titel           LIKE buch.titel,
            res_datum       LIKE reservierung.res_datum
        END RECORD,

        b_selektiert    SMALLINT    # Dieses Flag ist fuer buch. Wenn ein Buch selektiert fuer die
                                    # Funktionen "Modifizieren"und "Loeschen" wird, dann wird es auf 1
                                    # gesetzt, sonst auf 0.
END GLOBALS

#------------------------------- Ende des Moduls ----------------------------------------------#
```

```
#*****************************************************************************************#
#       MODUL:      main.4gl                                                                  #
#                                                                                             #
#       BESCHREIBUNG        : Hauptmenues werden angezeigt.                                   #
#                                                                                             #
#       DATUM : 18.11.1994                                                                    #
#                                                                                             #
#       AUTOR : Jinsook Kim                                                                   #
#                                                                                             #
#*****************************************************************************************#

GLOBALS
        "globals.4gl"

#       FUNKTION :    main
#                     Die Hauptfunktion.
#                     Wenn das System mit dem Befehl "biblsys.4ge" geoeffnet wird, dann wird zuerst
#                     Willkommenseite angezeigt und je nach dem Benutzerwuensch kleine Einfuehrung
#                     angezeigt und anschliessend Identifizierungsprozess ausgefuehrt. Je nach der
#                     Benutzerklasse bekommt der Benutzer unterschiedliche Menues.

MAIN

        DEFINE ant1,ant2        CHAR(1)

        DEFER INTERRUPT         # Diese Anweisung verhindert den sofortigen Programmabbruch. Mit
                                # INTERRUPT-Taste (CTRL-C) beendet nur die aktuell eingeleitete interaktive
                                # Aktion. Mit CTRL-\ Taste kann man auf Programmabbruch erzielen.
        OPTIONS
                PROMPT LINE 30,
                HELP FILE "hilfe.iem",
                MESSAGE LINE LAST-3,
                INPUT WRAP      # Die Schreibmarke bleibt solange in den  Eingabefeldern eines Formats,bis der
                                # Benutzer die Taste ESC drueckt.

        CALL willkommen()       # Willkommenseite ist angezeigt. Modul : hilfsfkt

        PROMPT "  Kleine Einfuehrung? (j/n) : "
        FOR ant1

        IF int_flag THEN
                LET int_flag = FALSE
                EXIT PROGRAM            # Programmabbruch
        END IF

        IF ant1 MATCHES "[jJ]" THEN
                CALL einfuehrung()      # Modul : hilfsfkt

                PROMPT "  Weiter? (j/n) : "
                FOR ant2
                IF ant2 MATCHES "[nN]" THEN
                        EXIT PROGRAM
                END IF
        END IF

        CLEAR SCREEN
        CALL id_prozess()
        MESSAGE "Programm ist beendet."
        SLEEP 2
        CLEAR SCREEN
END MAIN
```

```
#       FUNKTION :     id_prozess()
#       Aufgerufen in : main

FUNCTION id_prozess()

        DEFINE i        SMALLINT

        LABEL marke:

        OPEN WINDOW w_id AT 10,15 WITH 6 ROWS, 45 COLUMNS               #10->y
        ATTRIBUTE(BORDER, PROMPT LINE FIRST+1)                          #15->x
                DISPLAY "Als Gast geben Sie 'g' ein. " AT 4,5
                DISPLAY "Programm beenden : CTRL-\\ " AT 5,5
                PROMPT "    Benutzerkennung eingeben : "
                FOR id

                IF int_flag THEN
                        LET int_flag = FALSE
                        EXIT PROGRAM                    # Programmabbruch
                END IF
        CLOSE WINDOW w_id

        CASE
                WHEN id MATCHES "adm"                   # Bibliothekmitarbeiter
                        CALL mitarb_hauptmenue()

                WHEN id MATCHES "??????"                # Normale Benutzer
                        SELECT COUNT(*) INTO i
                        FROM benutzer
                        WHERE benutzernr = id
                        IF i = 0 THEN
                                ERROR "Ungueltige Benutzernummer!"
                                GOTO marke
                        END IF
                        CALL ben_hauptmenue()

                WHEN id MATCHES "g"                     # Gast
                        CALL gast_hauptmenue()

                OTHERWISE
                        ERROR " Ungueltige Benutzerkennung!"
                        GOTO marke
        END CASE
END FUNCTION

#       FUNKTION :     ben_hauptmenue()
#                      Hauptmenue fuer normalen Benutzer wird angezeigt.
#       Aufgerufen in : main

FUNCTION ben_hauptmenue()

        CALL ring_menue()

        MENU "BEN.-HAUPTMENUE"
                COMMAND KEY(b) "such.B" "Mittels bibliographischen Angaben nach Medien suchen"HELP1
                        CALL medientyp()

                COMMAND KEY(i) "such.I" "Mittels inhaltlichen Angaben nach Medien suchen" HELP 2
                        CALL such_i()

                COMMAND "Ausgabe" "noch zu verfeinern in CSLIB2000" HELP 3
```

```
                COMMAND "Verlaengerung" "Medien verlaengeren" HELP 4
                        CALL ben_verl_menue()

                COMMAND "Reservierung" "Medien reservieren" HELP 5
                        CALL ben_res_menue()

                COMMAND KEY(k) "ben.-Konto" "Benutzerkonto anzeigen und drucken" HELP 6
                        CALL ben_konto_menue()

                COMMAND KEY (t) "sTatistik" "Statistiken ueber Medienbestand,..anzeigen und drucken"
HELP 7
                        CALL statistik_menue_b()        # Statistik fuer normalen Benutzer

                COMMAND "Ende" "Zurueck zum Betriebssystem"
                        CLEAR SCREEN
                        EXIT MENU
        END MENU

END FUNCTION

#       FUNKTION :      medientyp()
#       Aufgerufen in : ben_hauptmenue()

FUNCTION medientyp()

        MENU "SUCH.B-MENUE"
                COMMAND "Buch" "Mittels bibliographischen Suchkriterien nach Buechern suchen"
                        CALL b_suchen()

                        # Da die Funktionen fuer normalen Benutzer und die Verwalter streng getrennt sind,
                        # muss die Suchefunktion  von dem normalen Benutzer keinen Einfluss auf sie von dem
                        # Verwalter geben.
                        LET b_selektiert = 0

                        CLEAR SCREEN
                        CALL ring_menue()

                COMMAND "Zeitschrift" "noch zu verfeineren in CSLIB2000"

                COMMAND KEY (a) "zeitschr.-Artikel" "noch zu verfeineren in CSLIB2000"

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        EXIT MENU
        END MENU
END FUNCTION

#       FUNKTION :      mitarb_hauptmenue()
#                       Hauptmenue fuer Bibliothek-Mitarbeiter wird angezeigt.
#       Aufgerufen in : main

FUNCTION mitarb_hauptmenue()

        CALL ring_menue()

        MENU "MITARB.-HAUPTMENU"
                COMMAND "Medien" "Medien suchen, erfassen und pflegen"
                        CALL medien_menue()             # Modul : medien

                COMMAND "Benutzer" "Benutzerdaten suchen, erfassen und pflegen"
                        CALL benutzer_menue()           # Modul : benutzer

                COMMAND "Such.i" "Mittels inhaltlichen Angaben suchen"
```

```
                        CALL such_i()

                COMMAND "Ausleihe" "Medien ausleihen"
                        CALL ausleihe_menue()           # Modul : verleihe

                COMMAND KEY (i) "reservIerung" "Medien reservieren"
                        CALL res_menue()                # Modul : verleihe

                COMMAND "Verlaengerung" "Medien verlaengeren"
                        CALL verl_menue()               # Modul : verleihe

                COMMAND KEY(h) "maHnungen" "noch zu verfeinern in CSLIB2000"

                COMMAND KEY(u) "rUeckgabe" "Medien zurueckgeben"
                        CALL rueckgabe_menue()          # Modul : verleihe

                COMMAND KEY(t) "sTatistik" "Statistiken ueber Medienbestand, Benutzerbestand,..anzeigen
und drucken"
                        CALL statistik_menue_a()        # Modul : statistik

                COMMAND "Ende" "Zurueck zum Betriebssystem" HELP 3
                        CLEAR SCREEN
                        EXIT PROGRAM
        END MENU
END FUNCTION

#       FUNKTION :      gast_hauptmenue()
#       Aufgerufen in : main

FUNCTION gast_hauptmenue()

        CALL ring_menue()
        MENU "GAST-HAUPTMENUE"
                COMMAND KEY(b) "such.B" "Mittels bibliographischen Angaben suchen"
                        CALL medientyp()

                COMMAND KEY(i) "such.I" "Mittels inhaltlichen Angaben suchen"
                        CALL such_i()

                COMMAND "Ende" "Zurueck zum Betriebssystem"
                        CLEAR SCREEN
                        EXIT PROGRAM
        END MENU
END FUNCTION

#------------------------ Ende des Moduls ----------------------------------------------------#
```

```
#*****************************************************************************************#
#       MODUL:      medien.4gl                                                            #
#                                                                                         #
#       BESCHREIBUNG: Das Dokument-Menue wird angezeigt und die Medien                    #
#                     werden in die Datenbank aufgenommen, geaendert und                  #
#                     geloescht.                                                          #
#                                                                                         #
#       DATUM : 18.11.1994                                                                #
#       AUTOR : Jinsook Kim                                                               #
#                                                                                         #
#*****************************************************************************************#

GLOBALS "globals.4gl"

#       FUNKTION:     medien_menue()
#                     zeigt auf dem Bildschirm ein Menue fuer Medientyp an.
#       Aufgerufen in : adm_hauptmenue()

FUNCTION medien_menue()

        DEFINE antwort CHAR(1)

        MENU "MEDIEN-MENUE"
        COMMAND "Buch" "Buecherdaten suchen, erfassen und pflegen"
                OPEN FORM buch_form FROM "buch_tab"
                DISPLAY FORM buch_form

                MENU "BUCH-MENUE"
                        COMMAND "Suchen" "Mittels Suchkriterien nach Buecher suchen"
                                OPEN FORM buch_form FROM "buch_tab"
                                DISPLAY FORM buch_form
                                CALL b_suchen()

                        COMMAND KEY (f) "erFassen" "Buecherdaten erfassen"
                                OPEN FORM buch1_form FROM  "buch1_tab"
                                DISPLAY FORM buch1_form
                                LET antwort = "j"

                        WHILE antwort = "j"
                                CALL b_erfassen()

                                OPEN WINDOW wd_ant AT 10,10 WITH 1 ROWS, 40 COLUMNS
                                ATTRIBUTE(BORDER)
                                        PROMPT "    Weitere Eingabe? (j/n):  "
                                        FOR antwort
                                CLOSE WINDOW wd_ant

                                IF antwort MATCHES "[^jJ]" THEN
                                        EXIT WHILE
                                END IF

                                END WHILE

                                CLEAR SCREEN
                                CALL ring_menue()
                                CLOSE FORM buch1_form
                                LET b_selektiert = 0

                        COMMAND "Modifizieren" "Buecherdaten modifizieren"
                                IF b_selektiert = 1 THEN
                                        CALL b_modifizieren()
```

```
                                ELSE
                                        ERROR "Kein Medium wurde selektiert."
                                        MESSAGE "Bitte zuerst suchen."
                                        SLEEP 2
                                END IF

                        COMMAND "Loeschen" "Buecherdaten loeschen"
                                IF b_selektiert = 1 THEN

                                        OPEN WINDOW wl_ant AT 10,10 WITH 1 ROWS, 40 COLUMNS
ATTRIBUTE(BORDER)
                                        PROMPT "Sind Sie sicher? (j/n)  "
                                        FOR antwort
                                        CLOSE WINDOW wl_ant

                                        IF antwort MATCHES "[jJ]" THEN
                                                CALL b_loeschen()
                                                MESSAGE "Dieses Medium wurde geloescht"
                                        ELSE
                                                MESSAGE "Kein Medium wurde geloescht"
                                        END IF
                                ELSE
                                        ERROR "Kein Medium wurde selektiert."
                                        MESSAGE "Bitte zuerst suchen."
                                END IF

                        COMMAND "Ende" "Zurueck zum MEDIEN-MENUE"
                                EXIT MENU
                END MENU

                CLOSE FORM buch_form
                CLEAR SCREEN
                CALL ring_menue()

        COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"
        COMMAND KEY (a) "zeitschrift-Aufsatz" "noch zu verfeinern in CSLIB2000"
        COMMAND "Ende" "Zurueck zum Hauptmenue"
                EXIT MENU

        END MENU
END FUNCTION

#       FUNKTION:       b_erfassen()
#                       Die ueber Format eingegebenen Datensaetze von Buechern werden in der Datenbank
#                       aufgenommen.
#       Aufgerufen in : medien_menue()

FUNCTION b_erfassen()

        DEFINE i,j SMALLINT, antwort  CHAR (1), dnr INTEGER, p_desnr LIKE desdef.desnr

        OPTIONS INPUT NO WRAP
        MESSAGE "Neuen Datensatz eingeben"

        FOR i = 1 TO scr_line()             # Um die Verwirrung zu vermeiden
                CLEAR s_des[i].*            # Ohne diese Operation bleiben die vorherigen
        END FOR                             # ARRAY-Eintraege auf dem Bildschirm.

        INPUT BY NAME p_buch.*
                AFTER FIELD signatur
                        SELECT COUNT(*) INTO i
                        FROM buch
```

```
                        WHERE signatur = p_buch.signatur
                        IF i > 0 THEN
                                ERROR "Dieses Medium existiert schon."
                                NEXT FIELD signatur
                        END IF
        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde erfasst."
                SLEEP 3
                RETURN(FALSE)
        END IF

        INPUT ARRAY p_des FROM s_des.*
        INSERT INTO buch
        VALUES (p_buch.*)

        FOR i=1 TO arr_count()
                DECLARE dnr_curs CURSOR FOR

                SELECT desnr INTO dnr
                FROM desdef
                WHERE desname MATCHES p_des[i].desname
                LET j = 1
                FOREACH dnr_curs
                        LET j=j+1
                END FOREACH

                IF j=1 THEN
                        LET p_desnr = 0
                        INSERT INTO desdef
                        VALUES(0,p_des[i].desname)
                        LET p_desnr = SQLCA.SQLERRD[2]

                        INSERT INTO deskriptor
                        VALUES(p_buch.signatur, p_desnr, "b")
                ELSE
                        INSERT INTO deskriptor
                        VALUES(p_buch.signatur, dnr, "b")

                END IF
        END FOR
        MESSAGE "Ein Medium wurde erfasst."
END FUNCTION

#       FUNKTION:       b_modifizieren()
#       Aufgerufen in : medien_menue

FUNCTION b_modifizieren()

        DEFINE i SMALLINT
        MESSAGE "zu modifizierende Mediendaten eingeben"

        INPUT BY NAME p_buch.* WITHOUT DEFAULTS
        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Keine Medienaten wurden modifiziert."
                SLEEP 3
                MESSAGE ""
                RETURN(FALSE)
        END IF
```

```
        UPDATE buch
        SET buch.* = p_buch.*
        WHERE buch.signatur = p_buch.signatur

        MESSAGE "Diese Mediendaten wurden modifiziert"
END FUNCTION

#       FUNKTION:       b_loeschen()
#                       Beim Loeschen eines Buches mussen die Eintraege der anderen Tabellen mit geloescht
#                       werden. Dadurch kann man die referenzielle Datenintegritaet erzielen.
#       Aufgerufen in : medien_menue

FUNCTION b_loeschen()

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Keine Mediendaten wurden geloescht."
                RETURN
        END IF

        DELETE FROM deskriptor                  # Um die Datenintegritaet zu sichern : Da es in den Tabellen
        WHERE signatur = p_buch.signatu         # "deskriptor" und "dokbez" noch die Eintraege von dem
                                                # geloeschten Buch gibt, muessen sie fuer die Datenintegritaet
                                                #  auch geloescht werden
        DELETE FROM dokbez
        WHERE original = p_buch.signatur OR uebersetzung = p_buch.signatur

        DELETE FROM reservierung
        WHERE signatur = p_buch.signatur

        DELETE FROM ausleihe
        WHERE signatur = p_buch.signatur

        DELETE FROM buch
        WHERE signatur = p_buch.signatur

        CLEAR buch.*
        MESSAGE "Diese Mediendaten wurden geloescht."
END FUNCTION

#---------------------------------- Ende des Moduls ----------------------------------------------#
```

```
#*****************************************************************************************#
#       MODUL: benutzer.4gl                                                               #
#                                                                                         #
#       BESCHREIBUNG : Benutzerdaten werden verwaltet                                     #
#                                                                                         #
#       DATUM :         18.11.1994                                                        #
#       AUTOR : Jinsook Kim                                                               #
#                                                                                         #
#*****************************************************************************************#

GLOBALS
        "globals.4gl"

DEFINE ben_selektiert SMALLINT          # Ein flag ben_selektiert wird gesetzt auf 1, wenn ein Datensatz fuer
                                        # die Funktionen "Modifizieren" und "Loeschen" selektiert ist.

#       FUNKTION :      benutzer_menue()
#       Aufgerufen in : adm_hauptmenue

FUNCTION benutzer_menue()

        DEFINE antwort CHAR(1)

        OPEN FORM ben_form FROM "benu_tab"
        DISPLAY FORM ben_form
        LET ben_selektiert = 0

        MENU "BEN.-MENUE"
                COMMAND "Suchen" "Benutzerdaten suchen"
                        CALL ben_suchen()

                COMMAND KEY (f) "erFassen" "Neue Benutzerdaten erfassen"
                        LET antwort = "j"
                        WHILE antwort MATCHES "[jJ]"
                                CALL ben_erfassen()
                                OPEN WINDOW wb_ant AT 10,10 WITH 1 ROWS,
                                        40 COLUMNS ATTRIBUTE(BORDER)
                                PROMPT "    Weitere Eingabe? (j/n):  "
                                FOR antwort
                                CLOSE WINDOW wb_ant
                                IF antwort MATCHES "[^jJ]" THEN
                                        EXIT WHILE
                                END IF
                        END WHILE
                        MESSAGE "Ende der Datenerfassung"
                        LET ben_selektiert = 0

                COMMAND "Modifizieren" "Benutzerdaten modifizieren"
                        IF ben_selektiert = 1 THEN
                                CALL ben_modifizieren()
                        ELSE
                                ERROR "Keine Benutzerdaten wurden selektiert."
                                MESSAGE "Bitte zuerst suchen!"
                        END IF

                COMMAND "Loeschen" "Benutzerdaten loeschen"
                        IF ben_selektiert = 1 THEN
                                CALL ben_loeschen()
                        ELSE
                                ERROR "Keine Benutzerdaten wurden selektiert."
                                MESSAGE "Bitte zuerst suchen."
                                SLEEP 2
```

```
                END IF
            COMMAND "Ben.-konto" "Benutzerkonto anzeigen"
                IF ben_selektiert = 1 THEN
                    CALL ben_konto_adm()
                ELSE
                    ERROR "Keine Benutzerdaten wurden selektiert."
                    MESSAGE "Bitte zuerst suchen."
                    SLEEP 2
                END IF

            COMMAND "Ende" "Zurueck zum Hauptmenue"
                EXIT MENU
        END MENU

        CLEAR SCREEN
        CALL ring_menue()
        CLOSE FORM ben_form
END FUNCTION

#       FUNKTION :    ben_suchen()
#       Aufgerufen in : benutzer_menue

FUNCTION ben_suchen()

        DEFINE s, bedingung CHAR(300), antwort CHAR(1)

        MESSAGE "Suchkriterien eingeben."
        CONSTRUCT BY NAME bedingung ON benutzer.*

        IF int_flag THEN
            LET int_flag = FALSE
            MESSAGE "Keine Suchkriterien eingegeben."
            LET ben_selektiert = 0
            RETURN
        END IF

        LET s = "SELECT benutzernr,nachname,vorname,adresse, benutzer.plz, ort.ort ",
            "FROM benutzer,ort WHERE benutzer.plz = ort.plz AND ",
            bedingung CLIPPED,
            " ORDER BY nachname"

        PREPARE s1 FROM s
        DECLARE benutzer_curs SCROLL CURSOR FOR s1
        OPEN benutzer_curs

        WHILE status != NOTFOUND
            FETCH FIRST benutzer_curs INTO p_benutzer.*
            IF status = NOTFOUND THEN
                MESSAGE "Keine Benutzerdaten mehr."
                SLEEP 2
                LET ben_selektiert = 0
                EXIT WHILE
            END IF

            DISPLAY BY NAME p_benutzer.*
            LET ben_selektiert = 1
            MESSAGE ""

            MENU "BEN.-SUCHEN"
                COMMAND "Naechster"
                    FETCH NEXT benutzer_curs INTO p_benutzer.*
                    IF status = NOTFOUND THEN
```

```
                                        ERROR "Keine Benutzerdaten mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH LAST benutzer_curs INTO p_benutzer.*
                                END IF
                                DISPLAY BY NAME p_benutzer.*
                                LET ben_selektiert = 1

                        COMMAND "Vorheriger"
                                FETCH PREVIOUS benutzer_curs INTO p_benutzer.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keine Benutzerdaten mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST benutzer_curs INTO p_benutzer.*
                                END IF
                                DISPLAY BY NAME p_benutzer.*
                                LET ben_selektiert = 1

                        COMMAND KEY (t) "ersTer"
                                FETCH FIRST benutzer_curs INTO p_benutzer.*
                                DISPLAY BY NAME p_benutzer.*
                                LET ben_selektiert = 1

                        COMMAND "Letzter"
                                FETCH LAST benutzer_curs INTO p_benutzer.*
                                DISPLAY BY NAME p_benutzer.*
                                LET ben_selektiert = 1

                        COMMAND "Ende" "Zurueck zum BEN.-MENUE"
                                RETURN
                END MENU
        END WHILE
        CLOSE benutzer_curs
END FUNCTION

#       FUNKTION :      ben_erfassen()
#       Aufgerufen in : benutzer_menue

FUNCTION ben_erfassen()

        DEFINE i        SMALLINT

        MESSAGE "Neue Benutzerdaten eingeben"

        INPUT BY NAME p_benutzer.*

                AFTER FIELD benutzernr
                        SELECT COUNT(*) INTO i
                        FROM benutzer
                        WHERE benutzernr = p_benutzer.benutzernr

                        IF i > 0 THEN
                                ERROR "Dieser Benutzer existiert schon."
                                NEXT FIELD benutzernr
                        END IF

                AFTER FIELD plz

                        SELECT COUNT(*) INTO i
                        FROM ort
                        WHERE plz = p_benutzer.plz
                        IF i = 0 THEN
                                ERROR "Ungueltige Postleitzahl!"
```

```
                        NEXT FIELD plz
                    END IF

                    SELECT ort INTO p_benutzer.ort
                    FROM ort
                    WHERE plz = p_benutzer.plz

                    DISPLAY p_benutzer.ort TO ort

        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Keine Benutzerdaten wurden erfasst."
                CLEAR FORM
                RETURN
        END IF

        INSERT INTO benutzer
        VALUES( p_benutzer.benutzernr, p_benutzer.nachname, p_benutzer.vorname,
                p_benutzer.adresse, p_benutzer.plz)

        IF status >= 0 THEN
                MESSAGE "Diese Benutzerdaten wurden erfasst"
                DISPLAY BY NAME p_benutzer.*
                SLEEP 3
        ELSE
                MESSAGE "Diese Benutzerdaten wurden NICHT erfasst"
                SLEEP 3
        END IF

        CLEAR FORM
END FUNCTION

#       FUNKTION :      ben_modifizieren()
#       Aufgerufen in : benutzer_menue

FUNCTION ben_modifizieren()

        DEFINE v_ort LIKE ort.ort

        MESSAGE "zu modifizierende Benutzerdaten eingeben"
        INPUT BY NAME p_benutzer.* WITHOUT DEFAULTS
                AFTER FIELD plz
                        SELECT ort INTO v_ort FROM ort
                        WHERE ort.plz = p_benutzer.plz
                        LET p_benutzer.ort = v_ort
        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Keine Benutzerdaten wurden modifiziert."
                RETURN
        END IF

        UPDATE benutzer
        SET     benutzer.benutzernr = p_benutzer.benutzernr,
                benutzer.nachname   = p_benutzer.nachname,
                benutzer.vorname    = p_benutzer.vorname,
                benutzer.adresse    = p_benutzer.adresse,
                benutzer.plz        = p_benutzer.plz
        WHERE benutzer.benutzernr   = p_benutzer.benutzernr
```

```
        MESSAGE "Diese Benutzerdaten wurden modifiziert"
END FUNCTION

#       FUNKTION :      benu_loeschen()
#                       Beim Loeschen sollte es geprueft werden, ob der loeschende Benutzer noch ausgeliehene
#                       Buecher hat oder ob es die Reservierung von ihm immer noch in der DB gibt. Es ist
#                       aeusserst wichtig, die Integritaet der Daten zu beibehalten.
#       Aufgerufen in : benutzer_menue

FUNCTION ben_loeschen()

        DEFINE ant1,ant2,ant3,ant4      CHAR(1),
            i,j1,j2                     SMALLINT

        SELECT COUNT(*) INTO i
        FROM ausleihe
        WHERE benutzernr = p_benutzer.benutzernr

        IF i > 0 THEN
                OPEN WINDOW wb_ant1 AT 10,10 WITH 4 ROWS,
                53 COLUMNS ATTRIBUTE(BORDER)

                DISPLAY "Dieser Benutzer hat noch ausgeliehene/n " AT 1,2
                DISPLAY "Buch/Buecher." AT 2,2
                PROMPT " Wollen Sie trotzdem diese Daten loeschen? (j/n): "
                FOR ant1
                CLOSE WINDOW wb_ant1

                IF ant1 MATCHES "[jJ]" THEN
                        DELETE FROM ausleihe
                        WHERE benutzernr=p_benutzer.benutzernr

                        MESSAGE "Der Eintrag fuer diesen Benutzer in Ausleihe wurde geloescht."
                        SLEEP 3

                        SELECT COUNT(*) INTO j1
                        FROM reservierung
                        WHERE benutzernr = p_benutzer.benutzernr

                        IF j1 > 0 THEN
                                OPEN WINDOW wb_ant2 AT 10,10 WITH 4 ROWS,
                                53 COLUMNS ATTRIBUTE(BORDER)

                                DISPLAY "Dieser Benutzer hat ein/mehrere Buch/Buecher" AT 1,2
                                DISPLAY "reserviert." AT 2,2
                                PROMPT " Wollen Sie trotzdem diese Daten loeschen? (j/n): "
                                FOR ant2
                                CLOSE WINDOW wb_ant2

                                IF ant2 MATCHES "[jJ]" THEN
                                        DELETE FROM reservierung
                                        WHERE benutzernr=p_benutzer.benutzernr

                                        MESSAGE "Der Eintrag fuer diesen Benutzer in Reservierung",
                                                "wurde geloescht."
                                        SLEEP 3

                                        DELETE FROM benutzer
                                        WHERE benutzernr = p_benutzer.benutzernr

                                        LET ben_selektiert = 0
                                        MESSAGE "Diese Benutzerdaten wurden geloescht."
```

```
                                CLEAR FORM
                        ELSE
                                MESSAGE "Nur die Eintraege von Ausleihe wurde geloescht."
                        RETURN
                        END IF # fuer ant2
                ELSE    # fuer j1
                        DELETE FROM benutzer
                        WHERE benutzernr = p_benutzer.benutzernr

                        LET ben_selektiert = 0
                        MESSAGE "Diese Benutzerdaten wurden geloescht."
                        CLEAR FORM
                        RETURN
                END IF # fuer j1
        ELSE    # fuer ant1
                MESSAGE "Diese Benutzerdaten wurden nicht geloescht."
                SLEEP 3
                RETURN
        END IF # fuer ant1
ELSE    # fuer i

        SELECT COUNT(*) INTO j2
        FROM reservierung
        WHERE benutzernr = p_benutzer.benutzernr

        IF j2 > 0 THEN
                OPEN WINDOW wb_ant3 AT 10,10 WITH 4 ROWS,
                53 COLUMNS ATTRIBUTE(BORDER)

                DISPLAY "Dieser Benutzer hat ein/mehrere Buch/Buecher" AT 1,2
                DISPLAY "reserviert." AT 2,2
                PROMPT " Wollen Sie trotzdem diese Daten loeschen? (j/n): "
                FOR ant3

                CLOSE WINDOW wb_ant3

                IF ant3 MATCHES "[jJ]" THEN
                        DELETE FROM reservierung
                        WHERE benutzernr=p_benutzer.benutzernr

                        MESSAGE "Der Eintrag fuer diesen Benutzer in Reservierung",
                                " wurde geloescht."
                        SLEEP 3

                        DELETE FROM benutzer
                        WHERE benutzernr = p_benutzer.benutzernr

                        LET ben_selektiert = 0
                        MESSAGE "Diese Benutzerdaten wurden geloescht."
                        CLEAR FORM
                        RETURN
                ELSE    # fuer ant3
                        MESSAGE "Diese Benutzerdaten wurden nicht geloescht."
                        RETURN
                END IF # fuer ant3

        ELSE    # fuer j2
                OPEN WINDOW wb_ant4 AT 10,10 WITH 1 ROWS,
                40 COLUMNS ATTRIBUTE(BORDER)
                PROMPT " Sind Sie sicher? (j/n): "
                FOR ant4
                CLOSE WINDOW wb_ant4
```

```
                        IF ant4 MATCHES "[jJ]" THEN
                                DELETE FROM benutzer
                                WHERE benutzernr = p_benutzer.benutzernr

                                LET ben_selektiert = 0
                                MESSAGE "Diese Benutzerdaten wurden geloescht."
                                CLEAR FORM
                                RETURN
                        END IF
                END IF # fuer j2
        END IF # fuer i
END FUNCTION

#       FUNKTION:       ben_konto_adm()
#                       Das Benutzerkonto wird angezeigt.
#       Aufgerufen in : ben_hauptmenue()

FUNCTION ben_konto_adm()

        DEFINE          s INTERVAL DAY TO DAY, i SMALLINT

        OPEN WINDOW w_bk_a AT 5,5 WITH FORM "ben_konto"
        ATTRIBUTE(FORM LINE FIRST)

        SELECT benutzernr,nachname,vorname,adresse,benutzer.plz,ort INTO p_bk.*
        FROM benutzer, ort
        WHERE benutzer.benutzernr = p_benutzer.benutzernr AND benutzer.plz = ort.plz

        DECLARE bk_curs CURSOR FOR

        SELECT signatur, ausleihdatum, rueckgabedatum
        FROM ausleihe
        WHERE benutzernr = p_benutzer.benutzernr
        ORDER BY ausleihe.signatur

        LET i = 1

        FOREACH bk_curs INTO p_ben_konto[i].signatur,p_ben_konto[i].ausleihdatum,
                        p_ben_konto[i].rueckgabedatum

                LET s = ((TODAY-p_ben_konto[i].rueckgabedatum) UNITS DAY ) / 7
                IF s > INTERVAL(0) DAY TO DAY THEN
                        LET p_ben_konto[i].saeumnisgebuehr = s
                ELSE
                        LET s = INTERVAL(0) DAY TO DAY
                        LET p_ben_konto[i].saeumnisgebuehr = s
                END IF

                LET i = i+1

        END FOREACH

        DISPLAY BY NAME p_bk.*
        CALL set_count(i-1)
        DISPLAY ARRAY p_ben_konto TO s_ben_konto.*

        CLOSE WINDOW w_bk_a
END FUNCTION

#---------------------------------- Ende des Moduls ----------------------------------------#
```

```
#*****************************************************************************************#
#       MODUL:ben_menue.4gl                                                               #
#                                                                                         #
#       BESCHREIBUNG: Die Funktionen, die in Benutzer-Hauptmenue auftauchen, z.B."Verlaengerung",  #
#                               "Reservierung" etc., werden hier ausgefuehrt.             #
#                                                                                         #
#       DATUM : 18.11.1994                                                                #
#                                                                                         #
#       AUTOR : Jinsook Kim                                                               #
#                                                                                         #
#*****************************************************************************************#

GLOBALS "globals.4gl"

DEFINE                  sig     LIKE ausleihe.signatur

#       FUNKTION:       ben_verl_menue()
#       Aufgerufen in : ben_hauptmenue()

FUNCTION ben_verl_menue()

        MENU "VERL.-MENUE"
                COMMAND "Buch" "Buecher verlaengern"
                        CALL ben_verl_buch()

                COMMAND "Zeitschrift" "noch zu verfeineren"

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        EXIT MENU

        END MENU
END FUNCTION

#       FUNKTION:       ben_verl_buch()
#                       Der Benutzer kann auch direkt auf dem Bildschirm das ausgeliehenen Medium
#                       verlaengern lassen. Diese Funktion stellt fest, ob das Medium schon vorgemerkt ist.
#                       Wenn es vorgemerkt ist, dann gibt die entsprechende Meldung aus.
#       Aufgerufen in : ben_verl_menue()

FUNCTION ben_verl_buch()

        DEFINE s        INTERVAL DAY(4) TO DAY

        OPEN FORM lan2_form FROM "lan2_tab"
        DISPLAY FORM lan2_form

        DECLARE bv_curs SCROLL CURSOR FOR

        SELECT benutzernr,ausleihe.signatur,buch.titel,ausleihdatum,rueckgabedatum
        INTO p_aus2.*
        FROM ausleihe, buch
        WHERE ausleihe.benutzernr = id AND
            ausleihe.signatur = buch.signatur
        ORDER BY ausleihe.signatur

        OPEN bv_curs
        WHILE status != NOTFOUND
                FETCH FIRST bv_curs INTO p_aus2.*
                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
                        CLEAR SCREEN
                        CALL ring_menue()
```

```
            SLEEP 3
            EXIT WHILE
    END IF

    LET s=((TODAY-p_aus2.rueckgabedatum)UNITS DAY )/7
    IF s > INTERVAL(0) DAY TO DAY THEN
            LET p_aus2.saeumnisgebuehr = s
    ELSE
            LET p_aus2.saeumnisgebuehr = INTERVAL(0) DAY TO DAY
    END IF

    DISPLAY BY NAME p_aus2.*    # Aktuelle Werte werden angezeigt.
    MESSAGE ""

    MENU "VERL.-MENUE"
            COMMAND "Naechster"
                    FETCH NEXT bv_curs INTO p_aus2.*
                            IF status = NOTFOUND THEN
                            ERROR "Keinen Datensatz mehr."
                                    ATTRIBUTE (RED, REVERSE)
                            FETCH LAST bv_curs INTO p_aus2.*
                    END IF
                    LET s=((TODAY-p_aus2.rueckgabedatum)UNITS DAY )/7
                    IF s > INTERVAL(0) DAY TO DAY THEN
                            LET p_aus2.saeumnisgebuehr = s
                    ELSE
                    LET p_aus2.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                    END IF
                    DISPLAY BY NAME p_aus2.*

            COMMAND KEY(h) "vorHeriger"
                    FETCH PREVIOUS bv_curs INTO p_aus2.*
                    IF status = NOTFOUND THEN
                            ERROR "Keinen Datensatz mehr."
                                    ATTRIBUTE(RED, REVERSE)
                            FETCH FIRST bv_curs INTO p_aus2.*
                    END IF
                    LET s=((TODAY-p_aus2.rueckgabedatum)UNITS DAY )/7
                    IF s > INTERVAL(0) DAY TO DAY THEN
                            LET p_aus2.saeumnisgebuehr = s
                    ELSE
                    LET p_aus2.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                    END IF
                    DISPLAY BY NAME p_aus2.*

            COMMAND KEY (t) "ersTer"
                    FETCH FIRST bv_curs INTO p_aus2.*
                    LET s=((TODAY-p_aus2.rueckgabedatum)UNITS DAY )/7
                    IF s > INTERVAL(0) DAY TO DAY THEN
                            LET p_aus2.saeumnisgebuehr = s
                    ELSE
                    LET p_aus2.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                    END IF
                    DISPLAY BY NAME p_aus2.*
            COMMAND "Letzter"
                    FETCH LAST bv_curs INTO p_aus2.*
                    LET s=((TODAY-p_aus2.rueckgabedatum)UNITS DAY )/7
                    IF s > INTERVAL(0) DAY TO DAY THEN
                            LET p_aus2.saeumnisgebuehr = s
                    ELSE
                    LET p_aus2.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                    END IF
```

```
                        DISPLAY BY NAME p_aus2.*

                    COMMAND "Verlaengern" "Dieses Medium wird verlaengert."
                        CALL ben_verl_buch_pr()

                    COMMAND "Ende"
                        CLOSE FORM lan2_form
                        CLEAR SCREEN
                        CALL ring_menue()
                        RETURN

                END MENU

        END WHILE
        CLOSE bv_curs
END FUNCTION

#       FUNKTION:   ben_verl_buch_pr()
#                   Wenn das Medium verlaengert werden darf, wird diese
#                   Funktion ausgefuehrt. D.h. es wird verlaengert.
#       Aufgerufen in : ben_verl()

FUNCTION ben_verl_buch_pr()

        DEFINE      i, j, k     SMALLINT, ant CHAR(1)

        SELECT COUNT(*) INTO i
        FROM reservierung
        WHERE signatur = p_aus2.signatur

        IF i > 0 THEN
            OPEN WINDOW w_bv AT 10,10 WITH 2 ROWS, 50 COLUMNS
            ATTRIBUTE(BORDER)
                DISPLAY "Es gibt ", i ," Reservierung(en). " AT 1,3
                DISPLAY "Dieses Medium kann nicht verlaengert werden!" AT 2,3
                SLEEP 6
            CLOSE WINDOW w_bv
            RETURN
        ELSE
            IF p_aus2.saeumnisgebuehr > INTERVAL(0) DAY TO DAY THEN

            OPEN WINDOW w_s AT 10,10 WITH 1 ROWS, 55 COLUMNS ATTRIBUTE(BORDER)
                DISPLAY "Sie muessen zuerst den Saeumnisgebuehr bezahlen!"
                AT 1,3
                SLEEP 5
            CLOSE WINDOW w_s
            RETURN

            ELSE
                OPEN WINDOW w_f AT 10,20 WITH 1 ROWS, 30 COLUMNS
                ATTRIBUTE(BORDER)
                    PROMPT "  Verlaengern? (j/n) : "
                    FOR ant
                CLOSE WINDOW w_f
                IF ant MATCHES "[jJ]" THEN
                    CALL ben_verl_buch_ok()
                                # Wenn keiner fuers Buch keine  Reservierung gemacht hat
                                # und keinen Saeumnisgebuehr vorgelegt ist, dann kann es
                                # verlaengert werden.
                ELSE
                    MESSAGE "Kein Medium wurde verlaengert."
                END IF
```

```
                END IF
        END IF
END FUNCTION

#       FUNKTION:       ben_verl_buch_ok()
#                       Wenn das Medium verlaengert werden darf, wird diese Funktion ausgefuehrt. D.h. es
#                       wird verlaengert.
#       Aufgerufen in : ben_verl()

FUNCTION ben_verl_buch_ok()

        DEFINE          ant     CHAR(1), signatur       CHAR(15), heute,rueck   DATE

        LET signatur= p_aus2.signatur
        LET heute   = TODAY
        LET rueck   = heute + 30 UNITS DAY

        DELETE FROM ausleihe
        WHERE ausleihe.benutzernr = id AND ausleihe.signatur = signatur

        DISPLAY heute,rueck TO ausleihdatum,rueckgabedatum

        INSERT INTO ausleihe
        VALUES(id,signatur,heute ,rueck)

        MESSAGE "Dieses Medium wurde verlaengert."
END FUNCTION

#       FUNKTION:       ben_res_menue()
#       Aufgerufen in : ben_hauptmenue()

FUNCTION ben_res_menue()

        MENU "RES.-MENUE"
                COMMAND "Reservierung" "Medien reservieren"
                        CALL ben_res_medium()

                COMMAND "Zuruecknahme" "Reservierung zuruecknehmen"
                        CALL ben_zurueck_medium()

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        EXIT MENU
        END MENU
END FUNCTION

#       FUNKTION:       ben_res_medium()
#       Aufgerufen in : ben_res_menue()

FUNCTION ben_res_medium()

        MENU "RES.-MENUE"
                COMMAND "Buch" "Buecher reservieren"
                        CALL ben_res_buch()
                COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"
                COMMAND "Ende"
                        EXIT MENU
        END MENU
END FUNCTION
```

```
#       FUNKTION:      ben_res_buch()
#       Aufgerufen in : ben_res_menue()

FUNCTION ben_res_buch()

        DEFINE q_br_b, bedingung CHAR(800),
             i  SMALLINT,
             antwort CHAR(1)

        OPEN FORM bb_res FROM "buch_tab"
        DISPLAY FORM bb_res

        MESSAGE "Suchkriterien eingeben."
        CONSTRUCT BY NAME bedingung ON buch.*     # Suche ueber Format

        IF int_flag THEN
                MESSAGE "Keine Suchkriterien eingegeben."
                SLEEP 2
                LET int_flag = FALSE
                CLOSE FORM bb_res
                CLEAR SCREEN
                CALL ring_menue()
                RETURN(FALSE)
        END IF
        LET q_br_b = "SELECT UNIQUE buch.* ",
                     "FROM buch WHERE ",
                      bedingung CLIPPED,
                     " ORDER BY signatur"

        PREPARE s_br_b FROM q_br_b

        DECLARE br_b_curs SCROLL CURSOR FOR s_br_b
        OPEN br_b_curs

        WHILE status != NOTFOUND
                FETCH FIRST br_b_curs INTO p_buch.*
                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"      # Ende der Erg.tab
                        SLEEP 3
                        EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*    # Aktuelle Werte werden angezeigt.
                MESSAGE ""

                MENU "RES.-MENUE"
                        COMMAND "Naechster"
                                FETCH NEXT br_b_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE (RED, REVERSE)
                                        FETCH LAST br_b_curs INTO p_buch.*
                                END IF
                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY(h) "vorHeriger"
                                FETCH PREVIOUS br_b_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST br_b_curs INTO p_buch.*
                                END IF
```

```
                        DISPLAY BY NAME p_buch.*

                    COMMAND KEY (t) "ersTer"
                        FETCH FIRST br_b_curs INTO p_buch.*
                        DISPLAY BY NAME p_buch.*

                    COMMAND "Letzter"
                        FETCH LAST br_b_curs INTO p_buch.*
                        DISPLAY BY NAME p_buch.*

                    COMMAND "Reservieren" "Dieses Medium wird reserviert."
                        CALL ben_res_buch_pr()

                    COMMAND "Ende"
                        CLEAR SCREEN
                        CLOSE FORM bb_res
                        CALL ring_menue()
                        RETURN
                END MENU
        END WHILE
        CLOSE br_b_curs
END FUNCTION

#       FUNKTION:     ben_res_buch_pr()
#       Aufgerufen in : ben_res_menue()

FUNCTION ben_res_buch_pr()

        DEFINE          i,j     SMALLINT,
                rd      LIKE    reservierung.res_datum,
                ant     CHAR(1)

        SELECT COUNT(*) INTO i                  # Um zu vermeiden, dass der Benutzer das ausgeliehenen
        FROM ausleihe                           # Buch reserviert, um spaeter mal spaeter verlaengern  lassen
        WHERE benutzernr = id AND               # zu koennen
            signatur = p_buch.signatur

        IF i > 0 THEN
                ERROR "Sie haben schon dieses Medium ausgeliehen!"
                RETURN
        END IF

        SELECT COUNT(*) INTO i
        FROM reservierung
        WHERE benutzernr = id AND signatur = p_buch.signatur

        IF i > 0 THEN                           # Um festzustellen, ob der Benutzer schon dieses Medium
                                                # reserviert hat.
                SELECT res_datum INTO rd
                FROM reservierung
                WHERE benutzernr = id AND signatur = p_buch.signatur

                OPEN WINDOW w_brok AT 10,10 WITH 4 ROWS, 50 COLUMNS
                ATTRIBUTE(BORDER)
                        DISPLAY "Sie haben schon dieses Medium am ", rd AT 2,2
                        DISPLAY "reserviert." AT 3,2
                        SLEEP 8
                CLOSE WINDOW w_brok
                RETURN
        END IF
        SELECT COUNT(*) INTO i
        FROM ausleihe
```

```
        WHERE signatur = p_buch.signatur

        IF i = 0 THEN
                OPEN WINDOW w_pr AT 10,10 WITH 3 ROWS, 45 COLUMNS
                ATTRIBUTE(BORDER)
                        DISPLAY "Dieses Medium braucht nicht reserviert" AT 1,3
                        DISPLAY "zu werden, da dieses im Regal steht." AT 2,3
                        SLEEP 8
                CLOSE WINDOW w_pr
                RETURN
        END IF

        SELECT COUNT(*) INTO i
        FROM reservierung
        WHERE reservierung.signatur = p_buch.signatur

        IF i = 0 THEN
                OPEN WINDOW w_bv1 AT 10,10 WITH 4 ROWS, 50 COLUMNS
                ATTRIBUTE(BORDER, PROMPT LINE LAST)
                        DISPLAY "Es gibt keine Reservierungen. " AT 1,3
                        DISPLAY "Die Wartezeit ist maximal 4 Wochen." AT 2,3
                        PROMPT "  Wollen Sie es reservieren? (j/n) : "
                        FOR ant
                CLOSE WINDOW w_bv1
        ELSE
                LET j = i*4                     # Wartezeit
                OPEN WINDOW w_bv2 AT 10,10 WITH 4 ROWS, 50 COLUMNS
                ATTRIBUTE(BORDER, PROMPT LINE LAST)
                        DISPLAY "Es gibt ",i," Reservierung(en). " AT 1,3
                        DISPLAY "Die Wartezeit ist maximal ",j, " Woche(n)." AT 2,3
                        PROMPT "  Wollen Sie es reservieren? (j/n) : "
                        FOR ant
                CLOSE WINDOW w_bv2
        END IF

        IF ant MATCHES "[jJ]" THEN
                CALL ben_res_buch_ok()

        END IF
END FUNCTION

#       FUNKTION:       ben_res_buch_ok()
#       Aufgerufen in : ben_res_menue()

FUNCTION ben_res_buch_ok()

        DEFINE i        LIKE reservierung.res_datum,
            j   SMALLINT
        INSERT INTO reservierung
        VALUES(         id, p_buch.signatur,TODAY)

        MESSAGE "Dieses Medium wurde reserviert."
END FUNCTION

#       FUNKTION:       ben_zurueck_medium()
#       Aufgerufen in : ben_res_menue()

FUNCTION ben_zurueck_medium()

        MENU "RES.-ZURUECK-MENUE"

                COMMAND "Buch" "Reservierung von den Buechern zuruecknehmen"
```

```
                CALL ben_zurueck_buch()
            COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"
            COMMAND "Ende"
                EXIT MENU
    END MENU
END FUNCTION

#       FUNKTION:     ben_zurueck_buch()
#       Aufgerufen in : ben_zurueck_medium()

FUNCTION ben_zurueck_buch()

    OPEN FORM res_zur FROM "res_zur"
    DISPLAY FORM res_zur

    DECLARE bz_curs SCROLL CURSOR FOR

    SELECT benutzernr, reservierung.signatur, buch.titel, res_datum
    INTO p_res_zurueck.*
    FROM reservierung, buch
    WHERE reservierung.benutzernr = id AND
        reservierung.signatur = buch.signatur
    ORDER BY reservierung.signatur

    OPEN bz_curs

    WHILE status != NOTFOUND
        FETCH FIRST bz_curs INTO p_res_zurueck.*
        IF status = NOTFOUND THEN
            MESSAGE "Datensatz nicht gefunden"      # Ende der Erg.tab
            SLEEP 3
            CLEAR SCREEN
            CALL ring_menue()
            EXIT WHILE
        END IF
        DISPLAY BY NAME p_res_zurueck.*

        MENU "RES.-ZURUECK-MENUE"
            COMMAND "Naechster"
                FETCH NEXT bz_curs INTO p_res_zurueck.*
                IF status = NOTFOUND THEN
                    ERROR "Keinen Datensatz mehr."
                        ATTRIBUTE (RED, REVERSE)
                    FETCH LAST bz_curs INTO p_res_zurueck.*
                END IF
                DISPLAY BY NAME p_res_zurueck.*

            COMMAND KEY(h) "vorHeriger"
                FETCH PREVIOUS bz_curs INTO p_res_zurueck.*
                IF status = NOTFOUND THEN
                    ERROR "Keinen Datensatz mehr."
                        ATTRIBUTE(RED, REVERSE)
                    FETCH FIRST bz_curs INTO p_res_zurueck.*
                END IF
                DISPLAY BY NAME p_res_zurueck.*

            COMMAND KEY (t) "ersTer"
                FETCH FIRST bz_curs INTO p_res_zurueck.*
                DISPLAY BY NAME p_res_zurueck.*

            COMMAND "Letzter"
                FETCH LAST bz_curs INTO p_res_zurueck.*
```

```
                        DISPLAY BY NAME p_res_zurueck.*

                    COMMAND "Zuruecknehmen" "Diese Reservierung wird zurueckgenommen"
                        CALL ben_zurueck_buch_ok()

                    COMMAND "Ende"
                        CLOSE FORM res_zur
                        CLEAR SCREEN
                        CALL ring_menue()
                        RETURN
                END MENU
        END WHILE

        CLOSE bz_curs
END FUNCTION

#       FUNKTION:       ben_zurueck_buch_ok()
#       Aufgerufen in : ben_zurueck_buch()

FUNCTION ben_zurueck_buch_ok()

        DEFINE ant      CHAR(1)

        OPEN WINDOW w_brz AT 10,10 WITH 1 ROWS, 40 COLUMNS
        ATTRIBUTE(BORDER)
                PROMPT "  Sind Sie sicher? (j/n) : "
                FOR ant
        CLOSE WINDOW w_brz

        IF ant MATCHES "[jJ]" THEN
                DELETE FROM reservierung
                WHERE benutzernr = id AND signatur = p_res_zurueck.signatur

                CLEAR FORM
                MESSAGE "Diese Reservierung wurde zurueckgenommen."
        ELSE
                MESSAGE "Diese Reservierung wurde NICHT zurueckgenommen."
        END IF
END FUNCTION

#       FUNKTION:       ben_konto_menue()
#       Aufgerufen in : ben_hauptmenue()

FUNCTION ben_konto_menue()

        MENU "BEN.-KONTO"
                COMMAND "Anzeigen" "Benutzerkonto anzeigen"
                        CALL ben_konto_an()

                COMMAND "Drucken" "Benutzerkonto drucken"
                        CALL ben_konto_dr()

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        CLOSE FORM ben_konto_form
                        CLEAR SCREEN
                        CALL ring_menue()
                        EXIT MENU
        END MENU
END FUNCTION
```

```
#       FUNKTION:       ben_konto_an()
#                       Das Benutzerkonto wird angezeigt.
#       Aufgerufen in : ben_konto_menue()

FUNCTION ben_konto_an()
        DEFINE          i       SMALLINT, s INTERVAL DAY(4) TO DAY

        SELECT benutzernr,nachname,vorname,adresse,benutzer.plz,ort INTO p_bk.*
        FROM benutzer, ort
        WHERE benutzernr = id AND benutzer.plz = ort.plz
        DECLARE bk_curs CURSOR FOR

        SELECT signatur, ausleihdatum, rueckgabedatum
        FROM ausleihe
        WHERE benutzernr = id
        ORDER BY ausleihe.signatur

        OPEN FORM ben_konto_form FROM "ben_konto"
        DISPLAY FORM ben_konto_form
        LET i = 1

        FOREACH bk_curs INTO p_ben_konto[i].signatur,p_ben_konto[i].ausleihdatum,
                        p_ben_konto[i].rueckgabedatum
                LET s = ((TODAY-p_ben_konto[i].rueckgabedatum) UNITS DAY ) / 7
                IF s > INTERVAL(0) DAY TO DAY THEN
                        LET p_ben_konto[i].saeumnisgebuehr = s
                ELSE
                        LET s = INTERVAL(0) DAY TO DAY
                        LET p_ben_konto[i].saeumnisgebuehr = s
                END IF

                LET i = i+1
        END FOREACH
        DISPLAY BY NAME p_bk.*
        CALL set_count(i-1)
        DISPLAY ARRAY p_ben_konto TO s_ben_konto.*

END FUNCTION

#       FUNKTION:       ben_konto_dr()
#       Aufgerufen in : ben_konto_menue()

FUNCTION ben_konto_dr()

        DEFINE          i SMALLINT, ant CHAR(1), s INTERVAL DAY(4) TO DAY,
                bkd RECORD
                        benutzernr      LIKE    benutzer.benutzernr,
                        nachname        LIKE    benutzer.nachname,
                        vorname         LIKE    benutzer.vorname,
                        adresse         LIKE    benutzer.adresse,
                        plz             LIKE    benutzer.plz,
                        ort             LIKE    ort.ort,
                        signatur        LIKE    ausleihe.signatur,
                        ausleihdatum    LIKE    ausleihe.ausleihdatum,
                        rueckgabedatum  LIKE    ausleihe.rueckgabedatum,
                        saeumnisgebuehr         INTERVAL DAY(4) TO DAY
                END RECORD

        DECLARE bkd_curs CURSOR FOR

        SELECT benutzer.benutzernr, nachname, vorname, adresse, benutzer.plz, ort, signatur, ausleihdatum,
rueckgabedatum
```

```
        FROM benutzer, ort , ausleihe
        WHERE benutzer.benutzernr = id AND ausleihe.benutzernr = id AND
            benutzer.plz = ort.plz   AND benutzer.benutzernr = ausleihe.benutzernr
        ORDER BY ausleihe.signatur

        LET i = 1

        START REPORT konto_rep TO "BEN-KONTO"

        FOREACH bkd_curs INTO bkd.*
                LET s = ((TODAY-bkd.rueckgabedatum) UNITS DAY ) / 7
                IF s > INTERVAL(0) DAY TO DAY THEN
                        LET bkd.saeumnisgebuehr = s
                ELSE
                        LET s = INTERVAL(0) DAY TO DAY
                        LET bkd.saeumnisgebuehr = s
                END IF

                OUTPUT TO REPORT konto_rep (bkd.*)
                LET i = i+1
        END FOREACH

        FINISH REPORT konto_rep

        PROMPT "Wollen Sie wirklich es drucken? (j/n): "
        FOR ant

        IF ant MATCHES "[jJ]" THEN
                MESSAGE "Es wird gedruckt im Raum fr5085, warten...."
                RUN "a2ps -nn BEN-KONTO | lpr -Pfr5085" WITHOUT WAITING
                        # Die gefundenen Daten werden gedruckt, ohne Programmende
                        # zu warten.

        ELSE
                MESSAGE "Es wird nicht gedruckt."
        END IF

END FUNCTION

#------------------------------ Ende des Moduls ---------------------------------#
```

```
#*****************************************************************************************#
#       MODUL:      suchen.4gl                                                          #
#                                                                                       #
#       BESCHREIBUNG : Die Dokumente werden mittels Attribute gesucht.                  #
#                                                                                       #
#       DATUM : 18.11.1994                                                              #
#       AUTOR : Jinsook Kim                                                             #
#                                                                                       #
#*****************************************************************************************#

GLOBALS "globals.4gl"

#       FUNKTION:       b_suchen()
#                       Die Buecher werden mittels Attribute gesucht.
#       Aufgerufen in : medientyp()

FUNCTION b_suchen()

        DEFINE q_b, bedingung CHAR(800),
            i  SMALLINT,
            antwort CHAR(1)

        OPEN FORM b_form FROM "buch_tab"
        DISPLAY FORM b_form

        MESSAGE "Suchkriterien eingeben"
        LET b_selektiert = 0

        CONSTRUCT BY NAME bedingung ON buch.*     # Suche ueber Format

        IF int_flag THEN
                MESSAGE "Keine Suchkriterien eingegeben"
                SLEEP 2
                LET int_flag = FALSE
                CLOSE FORM b_form
                CLEAR SCREEN
                CALL ring_menue()
                RETURN(FALSE)
        END IF

        LET q_b = "SELECT UNIQUE buch.* ",
                 "FROM buch WHERE ",
                 bedingung CLIPPED,
                 " ORDER BY titel"

        PREPARE s_b FROM q_b            # Mit dieser Anweisung(PREPARE) wird SELECT-Anweisung in eine
                                        #ausfuehrbare  Anweisung umgewandelt s_b enthaelt ausfuehrbare
                                        # Anweisung.

        DECLARE buch_curs SCROLL CURSOR FOR s_b
                                # Der Satzzeiger q_curs wird definiert.  Mit SCROLL ist q_curs ein frei
                                # positionierbaren Satzzeiger.

        OPEN buch_curs          # Mit OPEN-Anweisung wird der Satzzeiger q_curs unmittelbar vor dem ersten
                                # Satz der Ergebnistabelle positioniert.

        WHILE status != NOTFOUND
                FETCH FIRST buch_curs INTO p_buch.*
                                # Der Satzzeiger wird zum ersten Satz der Erg.tab gebracht und die Werte
                                # aus diesem Satz abgerufen
                IF status = NOTFOUND THEN
```

```
                CLEAR SCREEN
                CLOSE FORM b_form
                MESSAGE "Datensatz nicht gefunden"      # Ende der Erg.tab
                SLEEP 3
                CALL ring_menue()

                LET b_selektiert = 0
                EXIT WHILE
        END IF

        DISPLAY BY NAME p_buch.*    # Aktuelle Werte werden angezeigt.
        LET b_selektiert = 1
        MESSAGE ""

        MENU "BUCH SUCHEN"
                COMMAND "Naechster"
                        FETCH NEXT buch_curs INTO p_buch.*
                        IF status = NOTFOUND THEN
                                ERROR "Keinen Datensatz mehr."
                                        ATTRIBUTE (RED, REVERSE)
                                FETCH LAST buch_curs INTO p_buch.*
                        END IF
                        MESSAGE ""
                        LET b_selektiert = 1
                        DISPLAY BY NAME p_buch.*

                COMMAND KEY(h) "vorHeriger"
                        FETCH PREVIOUS buch_curs INTO p_buch.*
                        IF status = NOTFOUND THEN
                                ERROR "Keinen Datensatz mehr."
                                        ATTRIBUTE(RED, REVERSE)
                                FETCH FIRST buch_curs INTO p_buch.*
                        END IF
                        MESSAGE ""
                        LET b_selektiert = 1
                        DISPLAY BY NAME p_buch.*

                COMMAND KEY (t) "ersTer"
                        FETCH FIRST buch_curs INTO p_buch.*
                        MESSAGE ""
                        LET b_selektiert = 1
                        DISPLAY BY NAME p_buch.*

                COMMAND "Letzter"
                        FETCH LAST buch_curs INTO p_buch.*
                        MESSAGE ""
                        LET b_selektiert = 1
                        DISPLAY BY NAME p_buch.*

                COMMAND KEY (o) "Orig./ueb." "Original- oder Uebersetzungsversion suchen"
                        LET b_selektiert = 1
                        CALL orig_ueb()

                COMMAND "Ausdruck"
                        CALL ausdruck()
                COMMAND "Ende"
                        RETURN
        END MENU
    END WHILE
    CLOSE buch_curs

END FUNCTION
```

```
#       FUNKTION:       orig_ueb()
#                       Wenn das Dokument eine weitere Version auf andere Sprache hat,dann wird sie gesucht.
#       Aufgerufen in : b_suchen()

FUNCTION orig_ueb()

        DEFINE i        SMALLINT

        IF p_buch.orig_ueb ="o" THEN
                DECLARE zo_curs CURSOR FOR

                SELECT UNIQUE *
                FROM buch
                WHERE signatur = (SELECT UNIQUE uebersetzung
                                  FROM dokbez
                                  WHERE original = p_buch.signatur)
                LET i = 1
                FOREACH zo_curs INTO p_buch.*
                        LET i=i+1
                        DISPLAY BY NAME p_buch.*
                END FOREACH

                IF i=1 THEN
                        MESSAGE "Keine uerbersetzte Version existiert!"
                ELSE
                        MESSAGE "Das ist die uebersetzte Version!"

                END IF

        ELSE    IF p_buch.orig_ueb ="u" THEN
                        DECLARE zu_curs CURSOR FOR

                        SELECT UNIQUE *
                        FROM buch
                        WHERE signatur = (SELECT UNIQUE original
                                          FROM dokbez
                                          WHERE uebersetzung = p_buch.signatur)
                        LET i = 1
                        FOREACH zu_curs INTO p_buch.*
                                LET i=i+1
                                DISPLAY BY NAME p_buch.*
                        END FOREACH

                        IF i=1 THEN
                                MESSAGE "KeinE Orinialversion existiert in der Bibliothek!"
                        ELSE
                                MESSAGE "Das ist die Originalversion!"
                        END IF
                END IF
        END IF

END FUNCTION

#---------------------------------- Ende des Moduls ----------------------------------------#
```