```
#*****************************************************************************************#
#       MODUL:      recherche.4gl                                                             #
#                                                                                             #
#       BESCHREIBUNG: Mittels inhaltlichen Angaben werden die Medien recherchiert.            #
#                                                                                             #
#       DATUM : 25.11.1994                                                                    #
#                                                                                             #
#       AUTOR : Jinsook Kim                                                                   #
#                                                                                             #
#*****************************************************************************************#

GLOBALS
        "globals.4gl"

#       FUNKTION :      such_i()
#       Aufgerufen in : main

FUNCTION such_i()

        MENU "SUCH.I-MENUE"
                COMMAND "Deskriptoren" "Mittels Deskriptoren nach Medien suchen"
                        CALL d_suchen()

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

END FUNCTION

#       FUNKTION:       d_suchen()
#       Aufgerufen in : dok_menue()

FUNCTION d_suchen()

        MENU "DES.-SUCHEN"
                COMMAND "UND-Verknuepfung" "Deskriptoren UND verknuepft"
                        CALL und_suchen()

                COMMAND "ODER-Verknuepfung" "Deskriptoren ODER verknuepft"
                        CALL oder_suchen()

                COMMAND "Ende" "Zurueck zum SUCH.I-MENUE"
                        EXIT MENU
        END MENU

END FUNCTION

#       FUNKTION:       oder_suchen
#       Aufgerufen in : d_suchen()

FUNCTION oder_suchen()

        DEFINE          i,j,k       SMALLINT

        OPEN FORM des_form FROM "des_tab"
        DISPLAY FORM des_form

        MESSAGE "Deskriptoren eingeben"
        INPUT ARRAY p_des FROM s_des.*

        IF int_flag THEN
```

```
        LET int_flag = FALSE
        CLOSE FORM des_form
        CLEAR SCREEN
        CALL ring_menue()
        RETURN(FALSE)
END IF

LET i = arr_count()

CASE i

WHEN 1
        DECLARE do1_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE signatur IN (SELECT signatur FROM deskriptor
                        WHERE desnr IN (SELECT UNIQUE desnr
                                        FROM desdef
                                        WHERE desname MATCHES p_des[1].desname)
                        )
        ORDER BY titel

WHEN 2
        DECLARE do2_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT UNIQUE desnr FROM desdef
                                WHERE   desname MATCHES p_des[1].desname OR
                                        desname MATCHES p_des[2].desname)
                )
        ORDER BY titel

OTHERWISE
        DECLARE do3_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT UNIQUE desnr FROM desdef
                                WHERE   desname MATCHES p_des[1].desname OR
                                        desname MATCHES p_des[2].desname OR
                                        desname MATCHES p_des[3].desname )
                )
        ORDER BY titel

END CASE

LET k = 1

OPEN FORM buch_f FROM "buch_tab"
DISPLAY FORM buch_f

CASE i

 WHEN 1

        OPEN do1_curs
        WHILE status != NOTFOUND
                FETCH FIRST do1_curs INTO p_buch.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
                        SLEEP 3
```

```
                    CLEAR SCREEN
                    CLOSE FORM buch_f
                    CALL ring_menue()
                    EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*
                MESSAGE ""

                MENU "ODER-VERKNUEPFUNG"

                    COMMAND "Naechster"
                        FETCH NEXT do1_curs INTO p_buch.*
                        IF status = NOTFOUND THEN
                            ERROR "Keinen Datensatz mehr."
                                ATTRIBUTE (RED, REVERSE)
                            FETCH LAST do1_curs INTO p_buch.*
                        END IF

                        DISPLAY BY NAME p_buch.*

                    COMMAND KEY(h) "vorHeriger"
                        FETCH PREVIOUS do1_curs INTO p_buch.*
                        IF status = NOTFOUND THEN
                            ERROR "Keinen Datensatz mehr."
                                ATTRIBUTE(RED, REVERSE)
                            FETCH FIRST do1_curs INTO p_buch.*
                        END IF

                        DISPLAY BY NAME p_buch.*

                    COMMAND KEY (t) "ersTer"
                        FETCH FIRST do1_curs INTO p_buch.*
                        DISPLAY BY NAME p_buch.*

                    COMMAND "Letzter"
                        FETCH LAST do1_curs INTO p_buch.*
                        DISPLAY BY NAME p_buch.*

                    COMMAND "Ende"
                        CLOSE FORM des_form
                        CLOSE FORM buch_f
                        CLEAR SCREEN

                        CALL ring_menue()
                        RETURN
                END MENU
            END WHILE
            CLOSE do1_curs

        WHEN 2
            OPEN do2_curs
            WHILE status != NOTFOUND

                FETCH FIRST do2_curs INTO p_buch.*

                IF status = NOTFOUND THEN
                    MESSAGE "Datensatz nicht gefunden"
                    SLEEP 3
                    CLEAR SCREEN
                    CLOSE FORM buch_f
                    CALL ring_menue()
```

```
                        EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*
                MESSAGE ""
                MENU "ODER-VERKNUEPFUNG"
                        COMMAND "Naechster"
                                FETCH NEXT do2_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE (RED, REVERSE)
                                        FETCH LAST do2_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY(h) "vorHeriger"
                                FETCH PREVIOUS do2_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST do2_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY (t) "ersTer"
                                FETCH FIRST do2_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Letzter"
                                FETCH LAST do2_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

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

        CLOSE do2_curs

OTHERWISE
        OPEN do3_curs
        WHILE status != NOTFOUND
                FETCH FIRST do3_curs INTO p_buch.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
                        SLEEP 3
                        CLEAR SCREEN
                        CLOSE FORM buch_f
                        CALL ring_menue()
                        EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*
                MESSAGE ""
                MENU "ODER-VERKNUEPFUNG"
```

```
                                COMMAND "Naechster"

                                        FETCH NEXT do3_curs INTO p_buch.*
                                        IF status = NOTFOUND THEN
                                                ERROR "Keinen Datensatz mehr."
                                                        ATTRIBUTE (RED, REVERSE)
                                                FETCH LAST do3_curs INTO p_buch.*
                                        END IF

                                        DISPLAY BY NAME p_buch.*

                                COMMAND KEY(h) "vorHeriger"
                                        FETCH PREVIOUS do3_curs INTO p_buch.*
                                        IF status = NOTFOUND THEN
                                                ERROR "Keinen Datensatz mehr."
                                                        ATTRIBUTE(RED, REVERSE)
                                                FETCH FIRST do3_curs INTO p_buch.*
                                        END IF

                                        DISPLAY BY NAME p_buch.*

                                COMMAND KEY (t) "ersTer"
                                        FETCH FIRST do3_curs INTO p_buch.*
                                        DISPLAY BY NAME p_buch.*

                                COMMAND "Letzter"
                                        FETCH LAST do3_curs INTO p_buch.*
                                        DISPLAY BY NAME p_buch.*

                                COMMAND "Ende"
                                        CLOSE FORM des_form
                                        CLOSE FORM buch_f
                                        CLEAR SCREEN
                                        CALL ring_menue()

                                        RETURN
                        END MENU
                END WHILE

                CLOSE do3_curs

        END CASE

END FUNCTION

#       FUNKTION:       und_suchen()
#       Aufgerufen in : d_suchen()

FUNCTION und_suchen()

        DEFINE          i,j,k       SMALLINT

        OPEN FORM des_form FROM "des_tab"
        DISPLAY FORM des_form

        MESSAGE "Deskriptoren eingeben"

        INPUT ARRAY p_des FROM s_des.*

        IF int_flag THEN
                LET int_flag = FALSE
                CLOSE FORM des_form
```

```
        CLEAR SCREEN
        CALL ring_menue()
        RETURN(FALSE)
END IF

LET i = arr_count()
DISPLAY  i AT 19,1

CASE i

WHEN 1
        DECLARE du1_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT UNIQUE desnr
                                 FROM desdef
                                 WHERE desname MATCHES p_des[1].desname)
                )
        ORDER BY titel

WHEN 2
        DECLARE du2_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT desnr FROM desdef
                                 WHERE    desname MATCHES p_des[1].desname)
                ) AND

            signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT desnr FROM desdef
                                 WHERE    desname MATCHES p_des[2].desname)
                )
        ORDER BY titel

OTHERWISE
        DECLARE du3_curs SCROLL CURSOR FOR
        SELECT * FROM buch
        WHERE  signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT UNIQUE desnr FROM desdef
                                 WHERE    desname MATCHES p_des[1].desname)
                ) AND
            signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT desnr FROM desdef
                                 WHERE    desname MATCHES p_des[2].desname)
                ) AND
            signatur IN
                (SELECT signatur FROM deskriptor
                 WHERE desnr IN (SELECT desnr FROM desdef
                                 WHERE    desname MATCHES p_des[3].desname)
                )
        ORDER BY titel

END CASE
LET k = 1
OPEN FORM buch_f FROM "buch_tab"
DISPLAY FORM buch_f
```

```
CASE i
 WHEN 1
        OPEN du1_curs

        WHILE status != NOTFOUND

                FETCH FIRST du1_curs INTO p_buch.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
                        SLEEP 3
                        CLEAR SCREEN
                        CLOSE FORM buch_f
                        CALL ring_menue()
                        EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*
                MESSAGE ""
                MENU "UND-VERKNUEPFUNG"
                        COMMAND "Naechster"
                                FETCH NEXT du1_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE (RED, REVERSE)
                                        FETCH LAST du1_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY(h) "vorHeriger"
                                FETCH PREVIOUS du1_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST du1_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY (t) "ersTer"
                                FETCH FIRST du1_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Letzter"
                                FETCH LAST du1_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Ende"
                                CLOSE FORM des_form
                                CLOSE FORM buch_f
                                CLEAR SCREEN
                                CALL ring_menue()
                                RETURN
                END MENU

        END WHILE

        CLOSE du1_curs

 WHEN 2
        OPEN du2_curs
```

```
        WHILE status != NOTFOUND

                FETCH FIRST du2_curs INTO p_buch.*
                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
                        SLEEP 3
                        CLEAR SCREEN
                        CLOSE FORM buch_f
                        CALL ring_menue()
                        EXIT WHILE
                END IF

                DISPLAY BY NAME p_buch.*
                MESSAGE ""
                MENU "UND-VERKNUEPFUNG"
                        COMMAND "Naechster"
                                FETCH NEXT du2_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE (RED, REVERSE)
                                        FETCH LAST du2_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY(h) "vorHeriger"
                                FETCH PREVIOUS du2_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST du2_curs INTO p_buch.*
                                END IF

                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY (t) "ersTer"
                                FETCH FIRST du2_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Letzter"
                                FETCH LAST du2_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Ende"
                                CLOSE FORM des_form
                                CLOSE FORM buch_f
                                CLEAR SCREEN
                                CALL ring_menue()
                                RETURN
                END MENU

        END WHILE

        CLOSE du2_curs

OTHERWISE
        OPEN du3_curs
        WHILE status != NOTFOUND
                FETCH FIRST du3_curs INTO p_buch.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"
```

```
                        SLEEP 3
                        CLEAR SCREEN
                        CLOSE FORM buch_f
                        CALL ring_menue()
                        EXIT WHILE
                END IF
                DISPLAY BY NAME p_buch.*
                MESSAGE ""

                MENU "UND-VERKNUEPFUNG"
                        COMMAND "Naechster"
                                FETCH NEXT du3_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE (RED, REVERSE)
                                        FETCH LAST du3_curs INTO p_buch.*
                                END IF
                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY(h) "vorHeriger"
                                FETCH PREVIOUS du3_curs INTO p_buch.*
                                IF status = NOTFOUND THEN
                                        ERROR "Keinen Datensatz mehr."
                                                ATTRIBUTE(RED, REVERSE)
                                        FETCH FIRST du3_curs INTO p_buch.*
                                END IF
                                DISPLAY BY NAME p_buch.*

                        COMMAND KEY (t) "ersTer"
                                FETCH FIRST du3_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Letzter"
                                FETCH LAST du3_curs INTO p_buch.*
                                DISPLAY BY NAME p_buch.*

                        COMMAND "Ende"
                                CLOSE FORM des_form
                                CLOSE FORM buch_f
                                CLEAR SCREEN
                                CALL ring_menue()
                                RETURN
                END MENU

        END WHILE
        CLOSE du3_curs

    END CASE

    CLOSE FORM buch_f
    CLOSE FORM des_form
    CLEAR SCREEN
    CALL ring_menue()

END FUNCTION

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

```
#*****************************************************************************************#
#       MODUL:      verleihe.4gl                                                              #
#                                                                                             #
#       BESCHREIBUNG:       Die Medien werden ausgeliehen, reserviert, zurueckgegeben und verlaengert.  #
#                                                                                             #
#       DATUM : 18.11.1994                                                                    #
#       AUTOR : Jinsook Kim                                                                   #
#                                                                                             #
#*****************************************************************************************#

GLOBALS "globals.4gl"

DEFINE sig      LIKE reservierung.signatur

#       FUNKTION:       ausleihe_menue()
#                       Der Bibliothekverwalter gibt die Medien, die ausgeliehen werden, ein.
#       Aufgerufen in : adm_hauptmenue()

FUNCTION ausleihe_menue()

        MENU "AUSLEIHE-MENU"

                COMMAND "Ausleihen" "Medien ausleihen"
                        CALL ausleihen()

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

END FUNCTION

#       FUNKTION:       ausleihen()
#                       Der Bibliothekverwalter gibt die Medien, die ausgeliehen werden, ein.
#       Aufgerufen in : ausleihe_menue()

FUNCTION ausleihen()

        DEFINE          i,k,k1,k2           SMALLINT, curr, sl      INTEGER

        OPTIONS         INPUT NO WRAP

        OPEN FORM ausleihe_form FROM "ausleihe"
        DISPLAY FORM ausleihe_form
        INPUT pa_benutzernr FROM benutzernr

                AFTER FIELD benutzernr                  # Um zu kontrollieren, ob die
                        SELECT COUNT(*) INTO i          # die eingegebene Benutzernummer
                        FROM benutzer               # gueltig ist.
                        WHERE benutzernr = pa_benutzernr

                        IF i = 0 THEN
                                ERROR "Keine gueltige Benutzernummer!"
                                NEXT FIELD benutzernr
                        END IF
        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde ausgeliehen."
                SLEEP 2
                CLEAR SCREEN
```

```
            CALL ring_menue()
            CLOSE FORM ausleihe_form
            RETURN
    END IF

    INPUT ARRAY p_aus1 FROM s_aus1.*
            AFTER FIELD signatur
                    LET curr = arr_curr()
                    LET sl = scr_line()

                    SELECT COUNT(*) INTO i          # Um zu vermeiden, dass der
                    FROM ausleihe                   # Benutzer zweimal das gleiche
                    WHERE benutzernr = pa_benutzernr AND            # Buch ausleiht.
                        signatur = p_aus1[curr].signatur

                    IF i > 0 THEN
                            ERROR "Sie haben schon das Medium ausgeliehen!"
                            NEXT FIELD signatur
                    END IF

                    # Um zu vermeiden, dass das# ausgeliehene Buch nochmal
                    # ausgeliehen wird. Eigentlich koennte diese Situation nicht
                    # vorkommen, da der Benutzer das Buch in der Hand hat. Aber
                    # es koennte vorkommen, dass der Bibliothekmitarbeiter sich
                    # aus Versehen vertippt. Also Sicherheitshalber wird diese
                    # Pruefung durchgefuehrt.
                    SELECT COUNT(*) INTO i
                    FROM ausleihe
                    WHERE signatur = p_aus1[curr].signatur

                    IF i > 0 THEN
                            ERROR "Dieses Medium wurde schon ausgeliehen!"
                            NEXT FIELD signatur
                    END IF

                    SELECT COUNT(*) INTO i          # Um zu kontrollieren, ob die ein-
                    FROM buch                       # gegebenen Signaturen gueltig sind
                    WHERE signatur = p_aus1[curr].signatur

                    IF i = 0 THEN
                            ERROR "Keine gueltige Signatur!"
                            NEXT FIELD signatur
                    END IF
                    LET p_aus1[curr].ausleihdatum = TODAY

                    AFTER FIELD ausleihdatum        # Das Rueckgabedatum wird bestimmt.
                    LET curr = arr_curr()

                    CASE
                      WHEN p_aus1[curr].signatur MATCHES "8qm*" # Fuer Buch : ein Monat
                            LET p_aus1[curr].rueckgabedatum
                                    = p_aus1[curr].ausleihdatum + 30 UNITS DAY

                      WHEN p_aus1[curr].signatur MATCHES "4xm*" # Fuer zeitschrift : eine Woche
                            LET p_aus1[curr].rueckgabedatum
                                    = p_aus1[curr].ausleihdatum + 7 UNITS DAY

                      OTHERWISE
                            LET p_aus1[curr].rueckgabedatum
                                    = p_aus1[curr].ausleihdatum + 30 UNITS DAY
                    END CASE
    END INPUT
```

```
        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde ausgeliehen."
                SLEEP 2
                CLEAR SCREEN
                CALL ring_menue()
                CLOSE FORM ausleihe_form
                RETURN(FALSE)
        END IF

        CALL ausleihe_ok()
        CLEAR SCREEN
        CALL ring_menue()
        CLOSE FORM ausleihe_form

END FUNCTION

#       FUNKTION:       ausleihe_ok()
#       Aufgerufen in : ausleihen()

FUNCTION ausleihe_ok()

        DEFINE i        SMALLINT

        FOR i = 1 TO arr_count()
                INSERT INTO ausleihe
                VALUES(         pa_benutzernr, p_aus1[i].signatur,
                        p_aus1[i].ausleihdatum,p_aus1[i].rueckgabedatum)
        END FOR

        MESSAGE "Diese Medien wurden ausgeliehen."
        SLEEP 2

END FUNCTION

#       FUNKTION:       res_menue()
#       Aufgerufen in : adm_hauptmenue()

FUNCTION res_menue()

        MENU "RES.-MENUE"

                COMMAND "Reservierung" "Medien reservieren"
                        CALL res_medien()

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

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

END FUNCTION

#       FUNKTION:       res_medien()
#       Aufgerufen in : res_menue()

FUNCTION res_medien()

        MENU "RES.-MENUE"
                COMMAND "Buch" "Buecher reservieren"
                        CALL res_buch_pr()
```

```
                COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"

                COMMAND "Ende"
                        EXIT MENU
        END MENU

END FUNCTION

#       FUNKTION:       res_buch_pr()
#       Aufgerufen in : res_medien()

FUNCTION res_buch_pr()

        DEFINE          i       SMALLINT, curr          INTEGER

        OPTIONS         INPUT NO WRAP

        OPEN FORM res_tab_form FROM "res_tab"
        DISPLAY FORM res_tab_form

        INPUT pv_benutzernr FROM benutzernr
                AFTER FIELD benutzernr
                        SELECT COUNT(*) INTO i
                        FROM benutzer
                        WHERE benutzernr = pv_benutzernr
                        IF i = 0 THEN
                                ERROR "Keine gueltige Benutzernummer!"
                                NEXT FIELD benutzernr
                        END IF
        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde reserviert."
                SLEEP 2
                CLEAR SCREEN
                CALL ring_menue()
                CLOSE FORM res_tab_form
                RETURN(FALSE)
        END IF
        INPUT ARRAY p_res2 FROM s_res2.*
                AFTER FIELD signatur
                        LET curr = arr_curr()

                        SELECT COUNT(*) INTO i
                        FROM buch
                        WHERE signatur = p_res2[curr].signatur
                        IF i = 0 THEN
                                ERROR "Keine gueltige Signatur!"
                                NEXT FIELD signatur
                        END IF
                        SELECT COUNT(*) INTO i          # Um zu vermeiden, dass der Benutzer zweimal
                        FROM reservierung               # das gleiche Buch ausleiht.
                        WHERE benutzernr = pv_benutzernr AND
                            signatur = p_res2[curr].signatur

                        IF i > 0 THEN
                                ERROR "Sie haben schon das Medium reserviert!"
                                NEXT FIELD signatur
                        END IF

                        SELECT COUNT(*) INTO i                  # Um zu vermeiden, dass der Benutzer das
```

```
                        FROM ausleihe                           # ausgeliehenen Buch reserviert, um spaeter
                        WHERE benutzernr = pv_benutzernr AND # verlaengert werden zu koennen.
                            signatur = p_res2[curr].signatur

                        IF i > 0 THEN
                                ERROR "Sie haben schon das Medium ausgeliehen!"
                                NEXT FIELD signatur
                        END IF

                        SELECT COUNT(*) INTO i
                        FROM ausleihe
                        WHERE signatur = p_res2[curr].signatur

                        IF i = 0 THEN

                        OPEN WINDOW w_respr AT 10,10 WITH 4 ROWS, 45 COLUMNS
                        ATTRIBUTE(BORDER)
                                DISPLAY "Dieses Buch braucht nicht reserviert" AT 1,3
                                DISPLAY "zu werden, da noch Exemplar/e im Regal" AT 2,3
                                DISPLAY "stehen. " AT 3,3
                                SLEEP 8
                        CLOSE WINDOW w_respr
                        RETURN
                        END IF

                        LET p_res2[curr].res_datum = TODAY
        END INPUT

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde reserviert."
                SLEEP 2
                CLEAR SCREEN
                CALL ring_menue()
                CLOSE FORM res_tab_form
                RETURN(FALSE)
        END IF

        CALL res_buch_ok()
        CLEAR SCREEN
        CALL ring_menue()
        CLOSE FORM res_tab_form
END FUNCTION

#       FUNKTION:       res_buch_ok()
#       Aufgerufen in : res_buch()

FUNCTION res_buch_ok()

        DEFINE i        SMALLINT

        FOR i = 1 TO arr_count()
                INSERT INTO reservierung
                VALUES (pv_benutzernr, p_res2[i].signatur,p_res2[i].res_datum)
        END FOR

        CALL set_count(i-1)
        DISPLAY ARRAY p_res2 TO s_res2.*
        MESSAGE "Diese Medien wurden reserviert."
        SLEEP 3

END FUNCTION
```

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

FUNCTION zurueck_medien()

        MENU "RES.-ZURUECK.-MENUE"
                COMMAND "Buch" "Reservierung fuer die Buecher zuruecknehmen"
                        CALL zurueck_buch()

                COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"

                COMMAND "Ende"
                        EXIT MENU
        END MENU

END FUNCTION

#       FUNKTION:      zurueck_buch()
#       Aufgerufen in : zurueck_medien()

FUNCTION zurueck_buch()

        DEFINE q, bedingung     CHAR(400)

        OPEN FORM res_zur FROM "res_zur"
        DISPLAY FORM res_zur
        MESSAGE "Suchenkriterien eingeben"

        CONSTRUCT bedingung ON benutzernr,reservierung.signatur,res_datum
                        FROM benutzernr,signatur,res_datum

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde selektiert."
                SLEEP 2
                CLOSE FORM res_zur
                CLEAR SCREEN
                CALL ring_menue()
                RETURN(FALSE)
        END IF
        LET q = "SELECT benutzernr,reservierung.signatur,buch.titel,res_datum ",
                "FROM reservierung, buch ",
                "WHERE reservierung.signatur = buch.signatur AND ",
                bedingung CLIPPED,
                " ORDER BY benutzernr "

        PREPARE zb FROM q
        DECLARE zb_curs SCROLL CURSOR FOR zb
        OPEN zb_curs

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

                DISPLAY BY NAME p_res_zurueck.*
                MESSAGE ""
```

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

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

                                DISPLAY BY NAME p_res_zurueck.*

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

                        COMMAND "Letzter"
                                FETCH LAST zb_curs INTO p_res_zurueck.*
                                DISPLAY BY NAME p_res_zurueck.*

                        COMMAND "Zurueecknehmen" "Diese Reservierung wird zurueckgenommen"
                                CALL zurueck_buch_ok()

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

END FUNCTION

#       FUNKTION:       zurueck_buch_ok()
#       Aufgerufen in : zurueck_buch()

FUNCTION zurueck_buch_ok()

        DEFINE ant      CHAR(1)

        OPEN WINDOW w_zb AT 10,10 WITH 3 ROWS, 35 COLUMNS
        ATTRIBUTE(BORDER, PROMPT LINE LAST-1)
                PROMPT "    Sind Sie sicher? (j/n) : "
                FOR ant
        CLOSE WINDOW w_zb

        IF ant MATCHES "[jJ]" THEN
                DELETE FROM reservierung
                WHERE           benutzernr = p_res_zurueck.benutzernr 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:       rueckgabe_menue()
#                       Das Rueckgabemenue wird angezeigt.
#       Aufgerufen in : adm_hauptmenue()

FUNCTION rueckgabe_menue()

        MENU "RUECKGABE-MENUE"
                COMMAND "Buch" "Buecher zurueckgeben"
                        CALL rueckgabe_suchen()

                COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"

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

END FUNCTION

#       FUNKTION:       rueckgabe_suchen()
#       Aufgerufen in : rueckgabe_menue()

FUNCTION rueckgabe_suchen()
        DEFINE          bedingung, q       CHAR(300),
                s       INTERVAL DAY(4) TO DAY

        OPEN FORM rueckgabe_form FROM "rueckgabe"
        DISPLAY FORM rueckgabe_form

        MESSAGE "Suchkriterien eingeben."
        CONSTRUCT bedingung ON benutzernr,ausleihe.signatur, buch.titel
                         FROM benutzernr,signatur,titel

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde selektiert."
                SLEEP 2
                CLOSE FORM rueckgabe_form
                CLEAR SCREEN
                CALL ring_menue()
                RETURN(FALSE)
        END IF

        LET q = "SELECT benutzernr,ausleihe.signatur,buch.titel,ausleihdatum,rueckgabedatum ",
                "FROM ausleihe, buch ",
                "WHERE ausleihe.signatur = buch.signatur AND ",
                 bedingung CLIPPED,
                " ORDER BY ausleihe.benutzernr "
        PREPARE q1 FROM q
        DECLARE r_curs SCROLL CURSOR FOR q1
        OPEN r_curs

        WHILE status != NOTFOUND
                FETCH FIRST r_curs INTO p_rueck.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden."
```

```
            CLOSE FORM rueckgabe_form
            CLEAR SCREEN
            CALL ring_menue()
            SLEEP 2
            EXIT WHILE
END IF

# Saeumnisgebuehrberechnung:
# Annahme: Ein Buch kann nur einen Monat lang ausgeliehen werden.
#             Wenn das Rueckgabedatum vorbei ist, muss der Benutzer
#         fuer jede nachfolgende Woche 1 DM als Saeumnisgebuehr
#             bezahlen.
# Formel:  Saeumnisgebuehr = (Rueckgabedatum - Heute) / 7 Tage

LET s = ((TODAY - p_rueck.rueckgabedatum) UNITS DAY ) / 7

# Da die INTERVAL-Variable auch die neg.-Zahl annehmen kann,
# soll es geprueft werden, ob die Zahl sinnvoll ist.
# (Bei Geld gibt es kein negatives.)
IF s > INTERVAL(0) DAY TO DAY THEN
        LET p_rueck.saeumnisgebuehr = s
ELSE
        LET p_rueck.saeumnisgebuehr = INTERVAL(0) DAY TO DAY
END IF

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

                DISPLAY BY NAME p_rueck.*

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

                DISPLAY BY NAME p_rueck.*

        COMMAND KEY(t) "ersTer"
                FETCH FIRST r_curs INTO p_rueck.*
```

```
                                LET s=((TODAY-p_rueck.rueckgabedatum)UNITS DAY )/7
                                IF s > INTERVAL(0) DAY TO DAY THEN
                                        LET p_rueck.saeumnisgebuehr = s
                                ELSE
                                        LET p_rueck.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                                END IF
                                DISPLAY BY NAME p_rueck.*

                        COMMAND "Letzter"
                                FETCH LAST r_curs INTO p_rueck.*
                                LET s=((TODAY-p_rueck.rueckgabedatum)UNITS DAY )/7
                                IF s > INTERVAL(0) DAY TO DAY THEN
                                        LET p_rueck.saeumnisgebuehr = s
                                ELSE
                                        LET p_rueck.saeumnisgebuehr=INTERVAL(0) DAY TO DAY
                                END IF
                                DISPLAY BY NAME p_rueck.*

                        COMMAND "Zurueckgeben" "Dieses Medium wird zurueckgegeben"
                                CALL zurueckgeben_ok()

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

#       FUNKTION:       zurueckgeben_ok()
#       Aufgerufen in : rueckgabe_menue()

FUNCTION zurueckgeben_ok()

        DEFINE ant      CHAR(1)

        OPEN WINDOW wr_ant AT 10,10 WITH 1 ROWS, 35 COLUMNS ATTRIBUTE(BORDER)
        PROMPT "   Sind Sie sicher? (j/n) : "
        FOR ant
        CLOSE WINDOW wr_ant

        IF ant MATCHES "[jJ]" THEN
                DELETE FROM ausleihe
                WHERE benutzernr = p_rueck.benutzernr AND
                    signatur = p_rueck.signatur

                MESSAGE "Dieses Medium wurde zurueckgegeben."
                SLEEP 2
        ELSE
                MESSAGE "Dieses Medium wurde NICHT zurueckgegeben."
                SLEEP 2
        END IF
        CLEAR FORM
END FUNCTION

#       FUNKTION:       verl_menue()
#       Aufgerufen in : ()

FUNCTION verl_menue()

        MENU "VERL.-MENUE"
```

```
                COMMAND "Buch" "Buecher verlaengern"
                        CALL verl_buch()

                COMMAND "Zeitschrift" "noch zu verfeinern in CSLIB2000"

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

END FUNCTION

#       FUNKTION:       verl_buch()
#       Aufgerufen in : verl_menue()

FUNCTION verl_buch()

        DEFINE          bedingung, q       CHAR(300),
                s       INTERVAL DAY(4) TO DAY

        OPEN FORM lan2_form FROM "lan2_tab"
        DISPLAY FORM lan2_form
        MESSAGE "Suchkriterien eingeben."

        CONSTRUCT bedingung ON benutzernr,ausleihe.signatur,buch.titel
                        FROM benutzernr,signatur,titel

        IF int_flag THEN
                LET int_flag = FALSE
                MESSAGE "Kein Medium wurde selektiert."
                SLEEP 2
                CLEAR SCREEN
                CLOSE FORM lan2_form
                CALL ring_menue()
                RETURN(FALSE)
        END IF

        LET q = "SELECT benutzernr,ausleihe.signatur,buch.titel,ausleihdatum,rueckgabedatum ",
                "FROM ausleihe, buch ",
                "WHERE ausleihe.signatur = buch.signatur AND ",
                bedingung CLIPPED,
                " ORDER BY ausleihe.signatur "

        PREPARE av FROM q
        DECLARE l_curs SCROLL CURSOR FOR av
        OPEN l_curs

        WHILE status != NOTFOUND
                FETCH FIRST l_curs INTO p_aus2.*

                IF status = NOTFOUND THEN
                        MESSAGE "Datensatz nicht gefunden"      # Ende der Erg.tab
                        CLOSE FORM lan2_form
                        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 l_curs INTO p_aus2.*
                IF status = NOTFOUND THEN
                        ERROR "Keinen Datensatz mehr."
                                ATTRIBUTE (RED, REVERSE)
                        FETCH LAST l_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 l_curs INTO p_aus2.*
                IF status = NOTFOUND THEN
                        ERROR "Keinen Datensatz mehr."
                                ATTRIBUTE(RED, REVERSE)
                        FETCH FIRST l_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 l_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 l_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 verl_buch_pr()

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

        CLOSE l_curs

END FUNCTION

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

FUNCTION verl_buch_pr()

        DEFINE          i,j,k       SMALLINT
        DECLARE bl_curs CURSOR FOR

        SELECT *
        FROM reservierung
        WHERE signatur = p_aus2.signatur

        LET i = 1 # Diese Variable ist nicht nur eine Laufvariable, sondern als
                  # auch ein Flag, das zeigt, ob es keine gefundenen Datensaetze
                  # gibt oder nicht.

        FOREACH bl_curs INTO p_res1[i].*
                IF status != NOTFOUND THEN
                        LET i=i+1
                END IF
        END FOREACH

        IF i = 1 THEN
                IF p_aus2.saeumnisgebuehr > INTERVAL(0) DAY TO DAY THEN

                OPEN WINDOW wv_s AT 7,10 WITH 2 ROWS, 55 COLUMNS ATTRIBUTE(BORDER)
                        DISPLAY "Sie muessen zuerst den Saeumnisgebuehr bezahlen!"
                        AT 1,3
                        SLEEP 5
                CLOSE WINDOW wv_s
                RETURN

                ELSE

                CALL verl_buch_ok()     # Wenn keiner fuers Buch keine Reservierung
                                        # gemacht hat und kein Saeumnisgebuehr
                                        # vorgelegt ist, dann kann es verlaengert
                                        # werden.
                END IF
        ELSE
                LET k = i-1
                DISPLAY k," Datensatz(saetze) gefunden" AT 20,5
                DISPLAY "Das Medium kann nicht verlaengert werden!" AT 22,5

                OPEN WINDOW w_verl_pr AT 5,5 WITH FORM "lan1_tab"
```

```
                ATTRIBUTE(FORM LINE FIRST)

                        CALL set_count(i-1)
                        DISPLAY ARRAY p_res1 TO s_lan1.*
                CLOSE WINDOW w_verl_pr

                DISPLAY "                                       " AT 20,5
                DISPLAY "                                       " AT 22,5

        END IF

END FUNCTION

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

FUNCTION verl_buch_ok()

        DEFINE bnr LIKE ausleihe.benutzernr, ant CHAR(1), signatur CHAR(15), heute,rueckDATE,
                j       SMALLINT


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

        DELETE FROM ausleihe
        WHERE benutzernr = p_aus2.benutzernr AND
            signatur   = p_aus2.signatur

        DISPLAY heute,rueck TO ausleihdatum,rueckgabedatum

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

        MESSAGE "Das Medium wird verlaengert."
        SLEEP 2

END FUNCTION

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

```
#*****************************************************************************************#
#       MODUL:      statistik.4gl                                                             #
#                                                                                             #
#       BESCHREIBUNG: Die Statistiken werden je nach der Benutzerklasse angezeigt.            #
#                                                                                             #
#       DATUM : 18.11.1994                                                                    #
#       AUTOR : Jinsook Kim                                                                   #
#                                                                                             #
#*****************************************************************************************#

GLOBALS "globals.4gl"

#       FUNKTION:       statistik_menue_a()
#                       Mit dieser Option erhaelt der Benutzer Ueberblick
#       Aufgerufen in : ben_menue()

FUNCTION statistik_menue_a()

        MENU "STATISTIK"
                COMMAND "Anzeigen" "Statistiken ueber Medienbestand, Benutzerbestand,... anzeigen"
                        OPEN FORM sta_a FROM "stat_a"
                        DISPLAY FORM sta_a
                        CALL statistik_a()

                COMMAND "Drucken" "noch zu verfeinern in CSLIB2000"

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        CLOSE FORM sta_a
                        CLEAR SCREEN
                        CALL ring_menue()
                        RETURN
        END MENU

END FUNCTION

#       FUNKTION:       statistik_a()
#                       Statistiken fuer den Bibltiothek-Administrator
#       Aufgerufen in : adm_hauptmenue

FUNCTION statistik_a()

        DEFINE          p_sta RECORD
                b,z,benu,aus,res INTEGER
                END RECORD

        SELECT COUNT(*) INTO p_sta.b
        FROM buch

        SELECT COUNT(*) INTO p_sta.z
        FROM zeitschrift

        SELECT COUNT(*) INTO p_sta.benu
        FROM benutzer

        SELECT COUNT(*) INTO p_sta.aus
        FROM ausleihe

        SELECT COUNT(*) INTO p_sta.res
        FROM reservierung
        DISPLAY p_sta.* TO s_sta.*
END FUNCTION
```

```
#       FUNKTION:       statistik_menue_b()
#                       Mit dieser Option erhaelt der Benutzer Ueberblick
#       Aufgerufen in : main

FUNCTION statistik_menue_b()

        MENU "STATISTIK"
                COMMAND "Anzeigen" "Statistiken ueber Medienbestand anzeigen"
                        OPEN FORM sta_b FROM "stat_b"
                        DISPLAY FORM sta_b

                        CALL statistik_b()

                COMMAND "Drucken" "noch zu verfeinern in CSLIB2000"

                COMMAND "Ende" "Zurueck zum Hauptmenue"
                        CLOSE FORM sta_b
                        CLEAR SCREEN
                        CALL ring_menue()
                        RETURN
        END MENU

END FUNCTION

#       FUNKTION:       statistik_b()
#                       Statistiken fuer normalen Benutzer
#       Aufgerufen in : ben_hauptmenue()

FUNCTION statistik_b()

        DEFINE          p_sta RECORD
                b,z     INTEGER
                END RECORD

        SELECT COUNT(*) INTO p_sta.b
        FROM buch

        SELECT COUNT(*) INTO p_sta.z
        FROM zeitschrift

        DISPLAY p_sta.* TO s_sta.*

END FUNCTION

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

```
#*****************************************************************************************#
#       MODUL:      liste.4gl                                                                 #
#                                                                                             #
#*****************************************************************************************#

GLOBALS "globals.4gl"

#       FUNKTION:       b_rep()
#       Aufgerufen in : b_suchen()

REPORT b_rep(b)

        DEFINE b RECORD LIKE buch.*

        OUTPUT
                LEFT MARGIN 0                   # Standartwert : 5
                RIGHT MARGIN 80                 # Standartwert : 132
                TOP MARGIN 1                    # Standartwert : 3
                BOTTOM MARGIN 0                 # Standartwert : 3
                PAGE LENGTH 30                  # Standartwert : 66
        FORMAT
                FIRST PAGE HEADER PRINT "BUECHERLISTE"
                        PRINT "============"
                        SKIP 1 LINES
                        PRINT "SIGNATUR", COLUMN 18, "TITEL"
                        SKIP 1 LINE

                ON EVERY ROW
                        PRINT b.signatur, COLUMN 18, b.titel CLIPPED

END REPORT

#       FUNKTION:       konto_rep(bk)
#       Aufgerufen in : ben_konto_dr()

REPORT konto_rep(bk)

        DEFINE bk  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

        OUTPUT
                LEFT MARGIN 0                   # Standartwert : 5
                RIGHT MARGIN 80                 # Standartwert : 132
                TOP MARGIN 1            # Standartwert : 3
                BOTTOM MARGIN 0                 # Standartwert : 3
                PAGE LENGTH 30                  # Standartwert : 66
        FORMAT
                FIRST PAGE HEADER PRINT "BENUTER-KONTO"
                        PRINT "============="
                        SKIP 1 LINES
                        PRINT 2 SPACES, "Benutzernr   : ", bk.benutzernr
```

```
                PRINT 2 SPACES, "Benutzername : ",bk.nachname CLIPPED,", ",
                                                  bk.vorname CLIPPED
                PRINT 2 SPACES, "Adresse      : ", bk.adresse CLIPPED
                PRINT 2 SPACES, "Ort          : ", bk.plz," ", bk.ort
                SKIP 1 LINES
        PRINT "  Signatur   Ausleihdatum   Rueckgabedatum   Saeumnisgebuehr"
        PRINT "  ------------------------------------------------------------"

        ON EVERY ROW
                PRINT 2 SPACES,bk.signatur CLIPPED, COLUMN 15, bk.ausleihdatum,
                        COLUMN 31, bk.rueckgabedatum, COLUMN 46, bk.saeumnisgebuehr
                        CLIPPED, " DM"

END REPORT

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

```
#*****************************************************************************************#
#       MODUL:      hilfsfunktion.4gl                                                       #
#                                                                                           #
#       FUNKTION : Die Hilfsfunktionen                                                      #
#                                                                                           #
#*****************************************************************************************#

GLOBALS
        "globals.4gl"

#       FUNKTION : ring_menue()

FUNCTION ring_menue()

        DISPLAY "+-----------------------------------------",
        "----- Hilfe : Ctrl- w --------+" AT 3,1 ATTRIBUTE(MAGENTA)
END FUNCTION

#       FUNKTION : willkommen()

FUNCTION willkommen()

  DISPLAY "+-----------------------------------------------------------------------------------------------+" AT 1,6
  DISPLAY "|                                                                                               |" AT 2,6
  DISPLAY "|                                                                                               |" AT 3,6
  DISPLAY "|     *****  *  *****  *     **** *    * ****                                                  |" AT 4,6
  DISPLAY "|     *     * * *     * *    *      *  * *                                                     |" AT 5,6
  DISPLAY "|     *     * * *     * *    *      * *  *                                                     |" AT 6,6
  DISPLAY "|     *****  *  *****  *     ****   *   ****                                                   |" AT 7,6
  DISPLAY "|     *     * * *     * *         *   *       *                                                |" AT 8,6
  DISPLAY "|     *     * * *     * *         *   *       *                                                |" AT 9,6
  DISPLAY "|     *****  *  *****  ***  *****   * *****                                                    |" AT 10,6
  DISPLAY "|                                                                                               |" AT 11,6
  DISPLAY "|                                                                                               |" AT 12,6
  DISPLAY "|                 Bibliothekssystem                                                             |" AT 13,6
  DISPLAY "|                Willkommen bei BIBLSYS                                                         |" AT 14,6
  DISPLAY "|                                                                                               |" AT 15,6
  DISPLAY "|                Autor  : Jinsook Kim                                                           |" AT 16,6
  DISPLAY "|                Datum  : 18. Nov. 1994                                                         |" AT 17,6
  DISPLAY "|                Version: 0.5                                                                   |" AT 18,6
  DISPLAY "|                                                                                               |" AT 19,6
  DISPLAY "|           Programmabbruch : Ctrl-c oder Ctrl-\                                                |" AT 20,6
  DISPLAY "|                                                                                               |" AT 21,6
  DISPLAY "+-----------------------------------------------------------------------------------------------+" AT 22,6

END FUNCTION

#       FUNKTION : einfuehrung()

FUNCTION einfuehrung()

  DISPLAY "+-------------------------------------------------------------------------------------+" AT 1,6
  DISPLAY "|                 KLEINE EINFUEHRUNG                                                  |" AT 2,6
  DISPLAY "|                 ==================                                                  |" AT 3,6
  DISPLAY "| BIBLSYS ist ein Prototyp eines Bibliotheksverwaltungssystems.                       |" AT 4,6
  DISPLAY "| Auf dieser Grundlage kann das Bibliothekssystem CSLIB2000 fuer                      |" AT 5,6
  DISPLAY "| die IFB der TUB entwickelt werden.                                                  |" AT 6,6
  DISPLAY "|                                                                                     |" AT 7,6
  DISPLAY "| * TASTENBELEGUNG BEIM EDITIEREN                                                     |" AT 8,6
  DISPLAY "|  CTRL A    :  Zeichen einfuegen/ueberschreiben(Umschaltmodus)                       |" AT 9,6
  DISPLAY "|  CTRL B    :  schnell rueckwaertz in den Feldern wandern                            |" AT 10,6
  DISPLAY "|  CTRL C    :  Abbruch                                                               |" AT 11,6
```

```
  DISPLAY "| CTRL D    : Eine Zeile loeschen(ab Cursorposition)                    |" AT 12,6
  DISPLAY "| CTRL F    : schnell vorwaerts in den Feldern wandern                  |" AT 13,6
  DISPLAY "| CTRL I    : Naechstes Feld                                            |" AT 14,6
  DISPLAY "| CTRL W    : Hilfe                                                     |" AT 15,6
  DISPLAY "| CTRL X    : Ein Zeichen loeschen                                      |" AT 16,6
  DISPLAY "| ESC       : Dateneingabe beenden                                      |" AT 17,6
  DISPLAY "| CR        : Naechstes Feld                                            |" AT 18,6
  DISPLAY "|                                                                       |" AT 19,6
  DISPLAY "| * SUCHOPERATOREN                                                      |" AT 20,6
  DISPLAY "| <     kleiner als          <=    kleiner gleich                       |" AT 21,6
  DISPLAY "| >     groesser als         >=    groesser gleich                      |" AT 22,6
  DISPLAY "| =     gleich               <>    ungleich                             |" AT 23,6
  DISPLAY "| :     zwischen             |     logisches ODER                       |" AT 24,6
  DISPLAY "| ?     Joker (n = 1)        *     Joker (n >= 1)                       |" AT 25,6
  DISPLAY "| leeres Feld bedeutet, es ist egal                                     |" AT 26,6
  DISPLAY "+-----------------------------------------------------------------------+" AT 27,6
END FUNCTION

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