

Java Sun - Projekt

2004-07-12

Entwicklung einer Buch- und
Benutzerverwaltung der
Schulbibliothek

Projektmitglieder:

Jörg Zilliken
Tim Kaczmarek
Frederik Stengel
Jens Welski

Betreuer:

Achim Oberreuter
Börde-Berufskolleg

Inhaltsverzeichnis

1 Projektbeschreibung

1.1 Projektumfeld 4

1.2 Projektdefinition 5

1.3 Projektumfang 5

2. Lösungskonzept

2.1 Aufbau der Lösung 5

2.2 Eingesetzte Verfahren..... 6

3. Proram Architecture

3.1 Übersicht und Funktionalitäten.....6-7

4. Benutzerschnittstelle

4.1 Konzept..... 8

4.2 Leihe eintragen..... 9

4.3 Leihe austragen..... 10

4.4 Benutzerverwaltung 11

4.4.1 Kunden anlegen 12

4.4.2 Kunden löschen 13

4.4.3 Kunden bearbeiten..... 14

4.4.4 Kundenstamm ansehen 15

4.5 Inventarverwaltung 16

4.5.1 Buch anlegen..... 17

4.5.2 Buch löschen..... 18

4.5.3 Buch bearbeiten 19

4.5.4 Inventar ansehen.....	20
4.6 Suchen	21
5. Referenzen	
5.1 Projektabschluß.....	22
5.2 Fazit.....	22
5.3 Quellennachweiss.....	22
6. Anhang.....	22
6.1 Anlage: Dokumentierter Quellcode.....	23
6.2 Anlage: Installationsanleitung.....	68

1 Projektbeschreibung

1.1 Projektumfeld

Übersicht

Inventarisierung und Verwaltung einer vorhandenen Schulbibliothek

Schule: **Name:** **Börde-Berufskolleg**
Anschrift: Geschwister-Scholl-Str. 1
Tel.: **(02921) 96390**

Projektteam: **Name:** **Tim Kaczmarek**
Anschrift: Schueltingerstr. 2a
59494 Soest

Tel.: (02921) 667662
Email: t.katsche@web.de

Jahrgang: 1982
Klasse: IT-M

Name: **Jörg Zilliken**
Anschrift: Wilhelm – Wulff – Weg 20
59594 Soest

Tel.: (02921) 65856
Email: j.zilliken@web.de

Jahrgang: 1977
Klasse: IT-M

Name: **Jens Welski**
Anschrift: Stegerwaldstraße 15
45731 Waltrop

Tel.: (02309) 74362
Email: NRWspontan@gmx.de

Jahrgang: 1982
Klasse: IT-M

Name: **Frederik Stengel**
Anschrift: Meerstrasse 8
59510 Lippetal

Tel.: (02527) 919590
Email: frederik.stengel@gmx.de

Jahrgang: 1976
Klasse: IT-M

Fachlehrer: **Name:** **Achim Oberreuter**
Anschrift: k.A.

Tel.: (02921) 31356
Email: oba-privat@t-online.de

1.2 Projektdefinition

Ziel war es ein Programm für die Schulbüchereien zu schreiben, das Modular aufgebaut ist und dadurch den Anforderungen der jeweiligen Büchereien zu entsprechen.

1.3 Projektumfang

Das Programm soll folgende Grundfunktionen beinhalten:

- Ein- / austragen von Büchern in einer Datenbank
- Anlegen und Löschen von Benutzern
- Ein- und Austragen von verliehenen Büchern
- Eine Suchfunktion um Bücher/ Benutzer zu finden
- Anzeige welche Bücher wann zurück kommen

2. Lösungskonzept

2.1 Aufbau der Lösung

Das Projekt baut auf einer MS-Access Datenbank auf, welche 3 Tabellen enthält. Zum einen haben wir eine Tabelle namens Kunden, welche mit den Kundendaten (Kundennummer, Name, Vorname, Strasse, PLZ, Ort) gefüllt wird. Des weiteren wurde eine Tabelle Inventar angelegt, welche die Inventardaten (Inventarnummer, Titel, Autor, Verlag) enthält und eine Tabelle Ausleihe, welche die Rückgabe,- und Ausleihdaten zu den Kundennummern und Inventarnummern enthält.

Über die JDBC-ODBC Schnittstelle werden nun, nach Eingabe der entsprechenden Werte, welche aus der in Java programmierten Eingabemaske übergeben werden, die entsprechenden SQL-Befehle abgesetzt.

Das AWT wird genutzt um eine Benutzeroberfläche zu erstellen, welche leicht erweiterbar (modular) aufgebaut ist. Dadurch können auch noch nachträglich neue Funktionen einfach hinzugefügt werden und das Programm dadurch in seiner Funktionalität erweitert werden.

2.2 Eingesetzte Verfahren

Zum einen ist das Projekt cascadiert programmiert worden, um die Modalität zu erreichen. Zum anderen benutzen wir die ODBC-JDBC Schnittstelle um SQL-Befehle für die Datenbank zu nutzen.

3 Programm-Architektur

3.1 Übersicht und Funktionalitäten

Das Programm besteht aus 15 Klassen, welche jederzeit modular erweiterbar sind. Diese Klassen haben folgende Aufgaben:

Name der Klasse	Aufgabe der Klasse
Gut.java (siehe Abb.1)	Mainklasse, zeigt Soll-Rückgaben von heute an, leitet weiter zu den anderen Funktionen, lädt den Treiber für die DB-Verbindung
Aktion.java	ActionListener für die Klasse Gut.java (Weiterleiten zu anderen Fenstern)
Auswahl.java	ItemListener für die Klasse Verwaltung.java. Zeichnet Fenster neu und löscht Eingabefelder beim anklicken einer Checkbox. Lädt Daten in die Kundenstammliste.
Beenden.java	WindowAdapter zum schließen der Fenster. Alle „Fenster-Klassen“ sind damit verknüpft.
DelBtn.java	ActionListener für die Klasse DelLeihe.java, beinhaltet SQL-Befehle zum löschen eines Datensatzes mit angegebenen Werten
DelLeihe.java (siehe Abb.3)	Stellt die Eingabemaske zum austragen eines Leihgegenstandes dar.

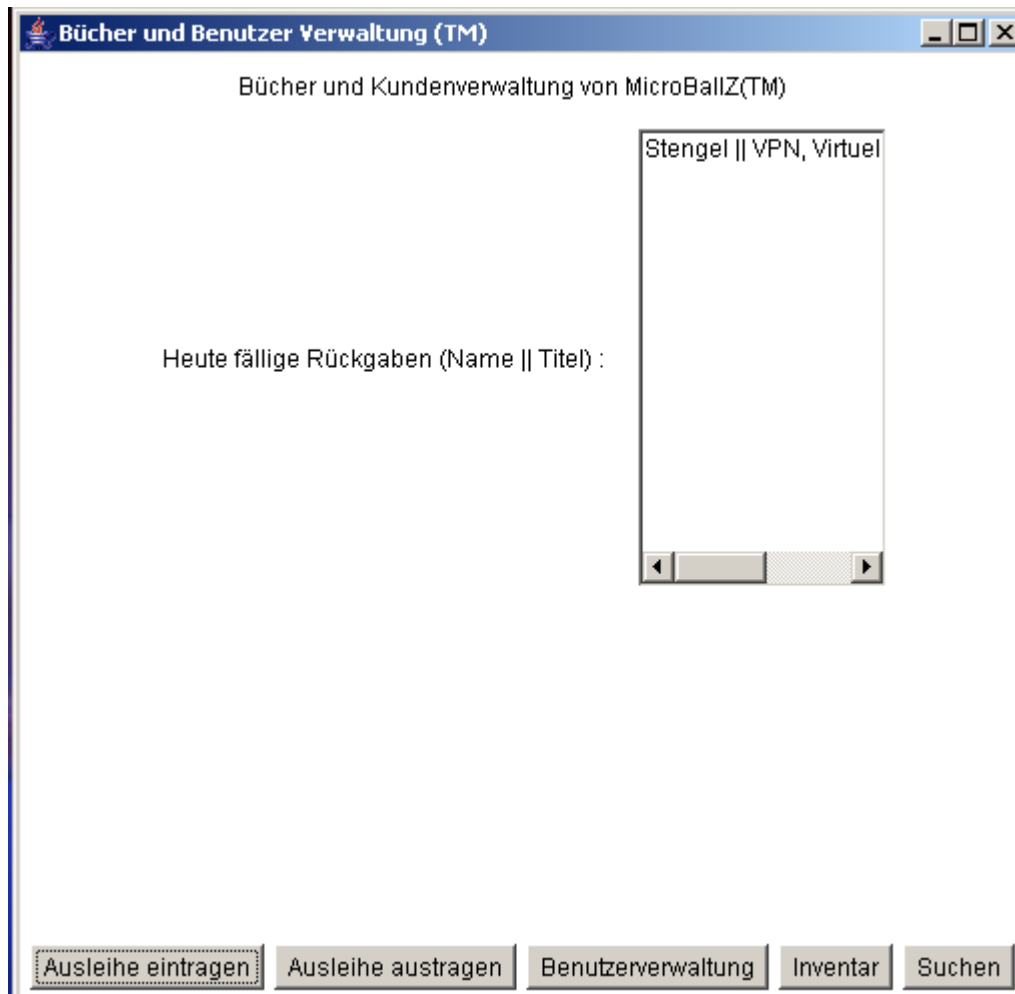
Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

InventBtn.java	ActionListener für die Klasse Inventar.java. Stellt alle Funktionen zum löschen, anlegen, bearbeiten und anzeigen des Inventars zur Verfügung.
InventAuswahl.java	ItemListener für die Klasse Inventar.java. Zeichnet die Fenster neu und löscht Eingabefelder beim anklicken einer Checkbox. Lädt Daten in die Inventarliste.
Inventar.java (siehe Abb.9 - 13)	Das Hauptfenster der Inventarverwaltung beinhaltet die einzelnen Panels und Objekte der Inventarverwaltung.
NewBtn.java	ActionListener für die Klasse NewLeihe.java, beinhaltet SQL-Befehle zum anlegen eines Datensatzes mit angegebenen Werten.
NewLeihe.java (siehe Abb.2)	Stellt die Eingabemaske zum eintragen eines Leihgegenstandes dar.
SuchBtn.java	ActionListener für die Klasse Suchen.java. Beinhaltet SQL-Befehle zum auslesen eines Datensatzes mit angegebenen Werten und gibt diese in eine Liste aus.
Suchen.java (siehe Abb.14)	Stellt die Eingabemaske zum Suchen und ausgeben eines Datensatzes dar.
VerwBtn.java	ActionListener für die Klasse Verwaltung.java stellt alle Funktionen zum löschen, anlegen, bearbeiten und anzeigen der Kunden zur Verfügung.
Verwaltung.java (siehe Abb.4 - 8)	Das Hauptfenster der Benutzerverwaltung beinhaltet die einzelnen Panels und Objekte der Benutzerverwaltung.

4. Benutzerschnittstellen

4.1 Konzept

Das Konzept ist einfach gehalten. Der Benutzer startet das Programm und es öffnet sich ein Fenster, von welchem aus man direkt in alle weiteren Funktionen des Programms per Klick auf einen Button kommt.



(Abb.1)

4.2 Leihe eintragen

Um einen neuen Leihgegenstand einzutragen, gibt der Benutzer die Kundennummer und die Inventarnummer an den vorgegebenen Stellen ein. Danach hat der Benutzer die Möglichkeit noch mal zu überprüfen ob es sich bei den Nummern um die korrekten handelt, indem er auf den Button „Pruefen“ klickt. Nachdem er auf den Button geklickt hat, werden Kundendaten (Name, Strasse, Ort) und die Inventardaten (Titel, Autor, Verlag) in die sich darunter befindlichen Listen übergeben. Sind die Daten korrekt, so kann der Benutzer auf den Button „Neuen Leihgegenstand eintragen“ klicken, um die Ausleihe einzutragen. Dabei werden die Kundennummer, und die Inventarnummer in die Datenbank eingetragen, und zusätzlich noch das Ausleihdatum und das Rückgabedatum. Das Rückgabedatum ist dabei das aktuelle Datum plus 14 Tage.

The screenshot shows a dialog box titled "Leihgegenstand eintragen". It has a blue title bar with standard window controls. The main area contains the following elements:

- Kundennummer:** A text input field containing "10001".
- Inventarnr.:** A text input field containing "1".
- Pruefen:** A button with a dotted border.
- Der Kunde ist:** A list box containing three entries: "Welski", "Stegerwaldstr. 15", and "Waltrop".
- Leihgegenstand ist:** A list box containing three entries: "Oracle 9i", "Hohenstein, Uwe,", and "DPUNKT VERLAG".
- Zurück zum Hauptfenster:** A button at the bottom left.
- Neuen Leihgegenstand eintragen:** A button at the bottom right.

(Abb.2)

4.3 Leihe austragen

Um einen Leihgegenstand auszutragen, gibt der Benutzer auch hier die Kundennummer und die Inventarnummer an den vorgegebenen Stellen ein. Nun hat der Benutzer die Möglichkeit noch mal zu überprüfen ob es sich bei den Nummern um die richtigen handelt, indem er auf den Button „Pruefen“ klickt. Nach der Überprüfung, werden Kundendaten (Name, Strasse, Ort) und die Inventardaten (Titel, Autor, Verlag) in die sich darunter befindlichen Listen übergeben. Sind die Daten richtig, so kann der Benutzer auf den Button „Leihgegenstand austragen“ klicken, um die Ausleihe auszutragen. Der komplette Datensatz wird daraufhin aus der Datenbank gelöscht.

The screenshot shows a window titled "Leihgegenstand austragen". It features two input fields: "Kundennummer:" with the value "10002" and "Inventarnr.:" with the value "2". To the right of these fields is a button labeled "Pruefen". Below the input fields are two list boxes. The first list box, labeled "Der Kunde ist:", contains the text "Stengel", "Meerstrasse8", and "Lippetal". The second list box, labeled "Leihgegenstand ist:", contains the text "VPN, Virtuelle Priva", "Lipp, Manfred", and "ADDISON-WESLEY". At the bottom of the window, there are two buttons: "Zurück zum Hauptfenster" and "Leihgegenstand austragen".

(Abb.3)

4.4 Benutzerverwaltung

In der Benutzerverwaltungsmaske befinden sich mehrere Funktionen, die durch markieren einer Checkbox aufgerufen werden.



Zurück zum Hauptfenster

(Abb.4)

4.4.1 Kunden anlegen

Um einen Kunden anzulegen, trägt man die Daten des Kunden in die dafür vorgesehenen Felder ein und klickt auf „Benutzer anlegen“. Daraufhin wird ein neuer Datensatz, mit den Eingetragenen Werten, in der Datenbank angelegt.

Benutzerverwaltung

Neuen Kunden anlegen Kunden löschen Kunden bearbeiten Kundenstamm ansehen

Kundennummer Name Vorname

Strasse und Hausnr. PLZ Ort

Benutzer anlegen

Zurück zum Hauptfenster

(Abb.5)

4.4.2 Kunden löschen

Um einen Kunden zu löschen, trägt man die Kundennummer in das dafür vorgesehene Feld ein und klickt auf „Pruefen“. Daraufhin werden in die danebenstehende Liste die Kundendaten (Nachname, Strasse, Ort) übernommen, um zu prüfen ob es sich um den richtigen Kunden handelt. Stimmt alles, so kann man den Button „Aus Kundenstamm entfernen“ anklicken, um den Datensatz aus der Datenbank zu löschen.

The screenshot shows a web application window titled "Benutzerverwaltung". At the top, there are four radio buttons for navigation: "Neuen Kunden anlegen", "Kunden löschen" (which is selected), "Kunden bearbeiten", and "Kundenstamm ansehen". Below this, the form for deleting a customer is displayed. It includes a label "Kundennummer:" followed by an empty text input field. To the right of the input field is a button labeled "Pruefen". Further right is the label "Kunde ist:" followed by a larger empty rectangular area, likely intended for a list of customer details. Below these elements is a button labeled "Aus Kundenstamm entfernen". At the bottom center of the window is a button labeled "Zurück zum Hauptfenster".

(Abb.6)

4.4.3 Kunden bearbeiten

Um einen Kunden zu bearbeiten, trägt man die Kundennummer in das dafür vorgesehene Feld ein und klickt auf „Pruefen“. Daraufhin werden in die Textfelder die jeweiligen Werte übernommen. Diese Werte sind nun veränderbar. Sind alle gewünschten Änderungen vorgenommen, so klickt man auf „Änderungen übernehmen“ und der Datensatz wird überschrieben.

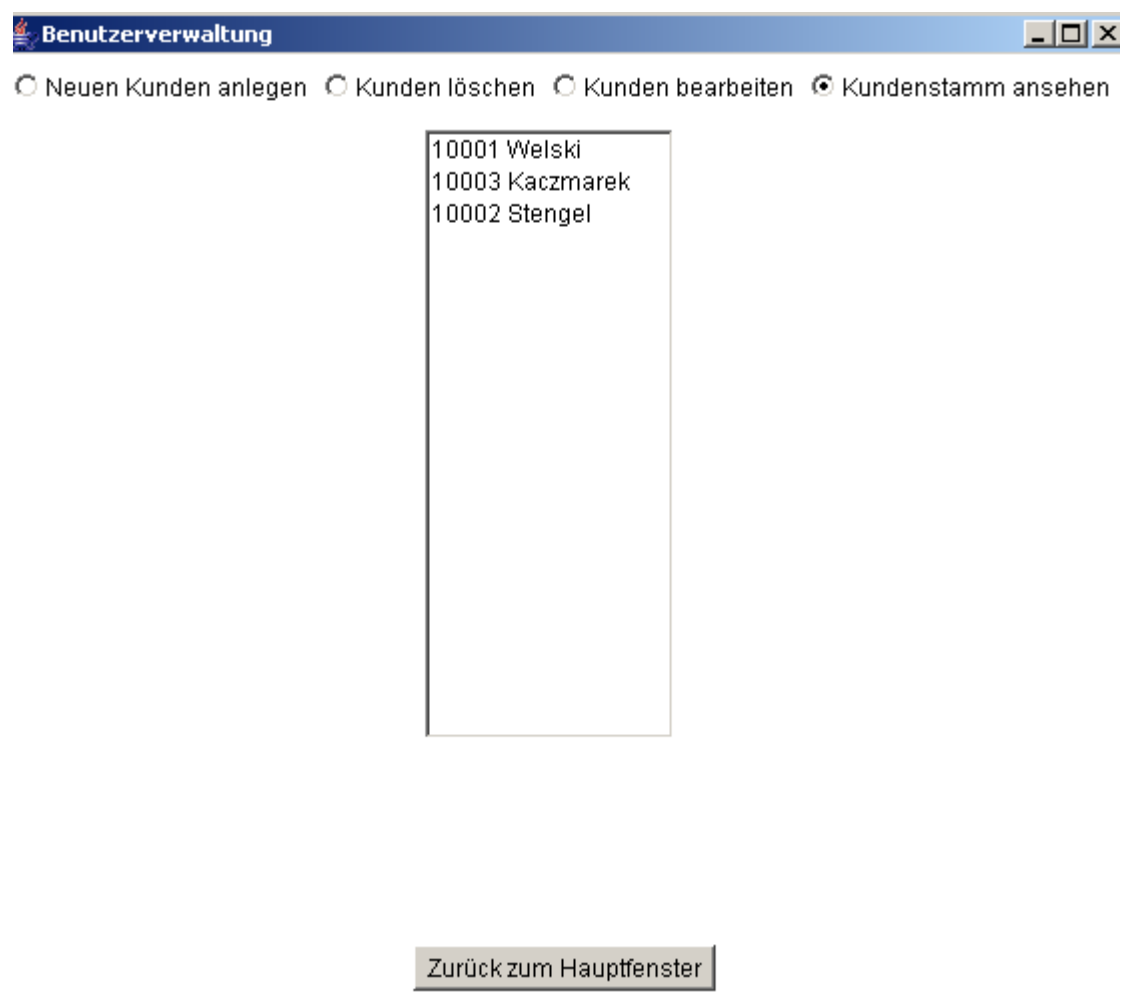
The screenshot shows a window titled "Benutzerverwaltung" with a blue header bar. Below the header, there are four radio buttons: "Neuen Kunden anlegen", "Kunden löschen", "Kunden bearbeiten" (which is selected), and "Kundenstamm ansehen". The form contains several input fields and buttons:

- "Kundennummer:" followed by an empty text box.
- A "Pruefen" button.
- "Name" followed by an empty text box.
- "Vorname" followed by an empty text box.
- An empty text box followed by "Strasse und Hausnr." followed by an empty text box.
- "PLZ" followed by an empty text box.
- "Ort" followed by an empty text box.
- A "Zurück zum Hauptfenster" button at the bottom.
- A "Änderungen übernehmen" button at the bottom right.

(Abb.7)

4.4.4 Kundenstamm ansehen

In dieser Maske befindet sich eine Liste, die den kompletten Kundenstamm anzeigt. Es werden jedoch nur die Werte Kundennummer und Nachname angezeigt.



(Abb.8)

4.5 Inventarverwaltung

In der Inventarverwaltungsmaske befinden sich mehrere Funktionen, die durch markieren einer Checkbox aufgerufen werden.



Zurück zum Hauptfenster

(Abb.9)

4.5.1 Neues Buch anlegen

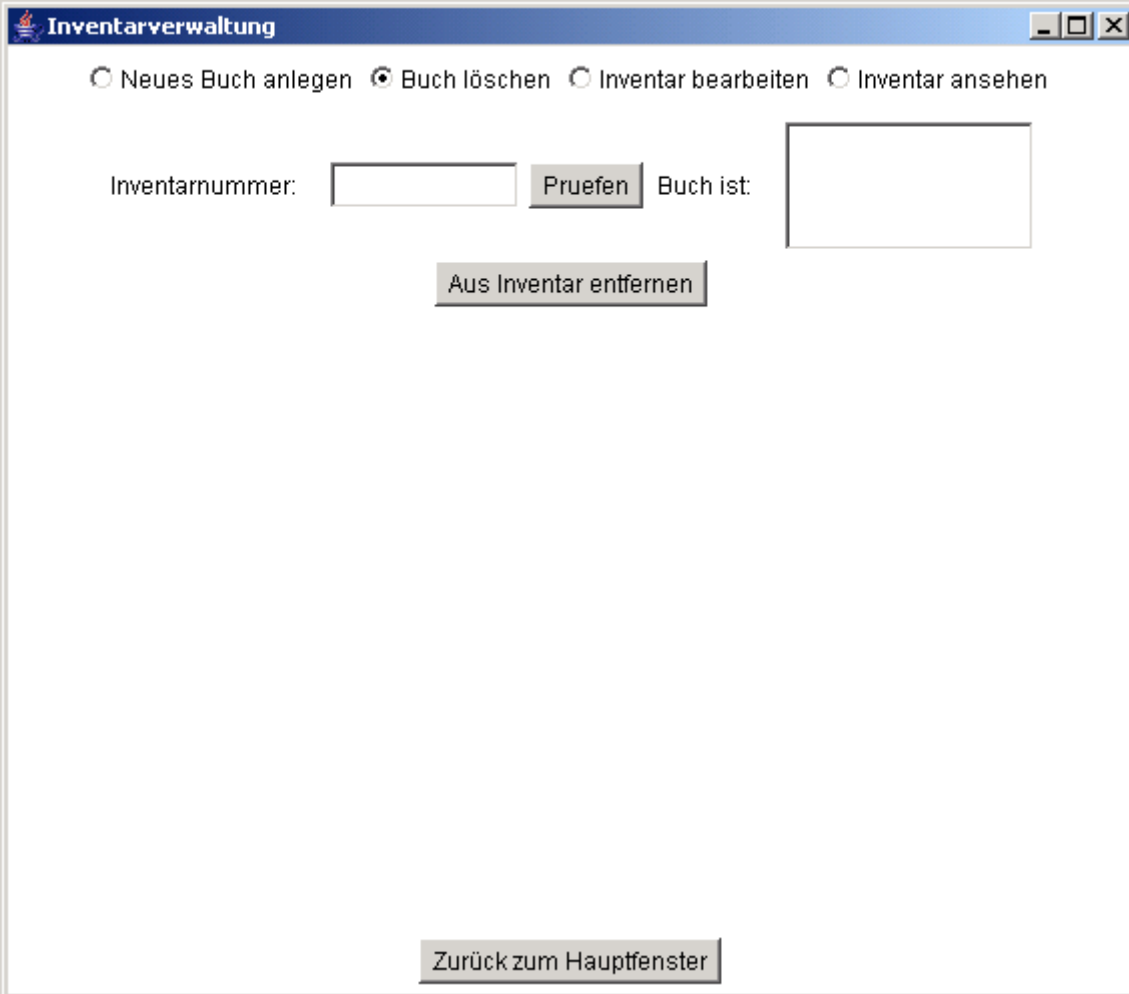
Um ein Buch anzulegen, trägt man die Daten des Buches in die dafür vorgesehenen Felder ein und klickt auf „Buch anlegen“. Daraufhin wird ein neuer Datensatz, mit den eingetragenen Werten, in der Datenbank angelegt.

The screenshot shows a window titled "Inventarverwaltung" with a blue header bar. Below the header, there are four radio buttons: "Neues Buch anlegen" (selected), "Buch löschen", "Inventar bearbeiten", and "Inventar ansehen". Below the radio buttons, there are four input fields labeled "Inventarnummer:", "Titel:", "Autor:", and "Verlag:". Below the "Autor:" field, there is a "Buch anlegen" button. At the bottom center of the window, there is a "Zurück zum Hauptfenster" button. A horizontal line is drawn below the "Zurück zum Hauptfenster" button.

(Abb.10)

4.5.2 Buch löschen

Um ein Buch zu löschen, trägt man die Inventarnummer in das dafür vorgesehene Feld ein und klickt auf „Pruefen“. Daraufhin werden in die danebenstehende Liste die Buchdaten (Titel, Autor, Verlag) übernommen um zu prüfen ob es sich um den richtigen Kunden handelt. Stimmt alles, so kann man den Button „Aus Inventar entfernen“ anklicken, um den Datensatz aus der Datenbank zu löschen.



The screenshot shows a window titled "Inventarverwaltung" with a blue header bar. Below the header, there are four radio buttons: "Neues Buch anlegen", "Buch löschen" (which is selected), "Inventar bearbeiten", and "Inventar ansehen". Below the radio buttons, there is a label "Inventarnummer:" followed by a text input field. To the right of the input field is a button labeled "Pruefen". To the right of the "Pruefen" button is the label "Buch ist:" followed by a large empty rectangular box. Below the "Pruefen" button and the "Buch ist:" label is a button labeled "Aus Inventar entfernen". At the bottom center of the window is a button labeled "Zurück zum Hauptfenster".

(Abb.11)

4.5.3 Buch bearbeiten

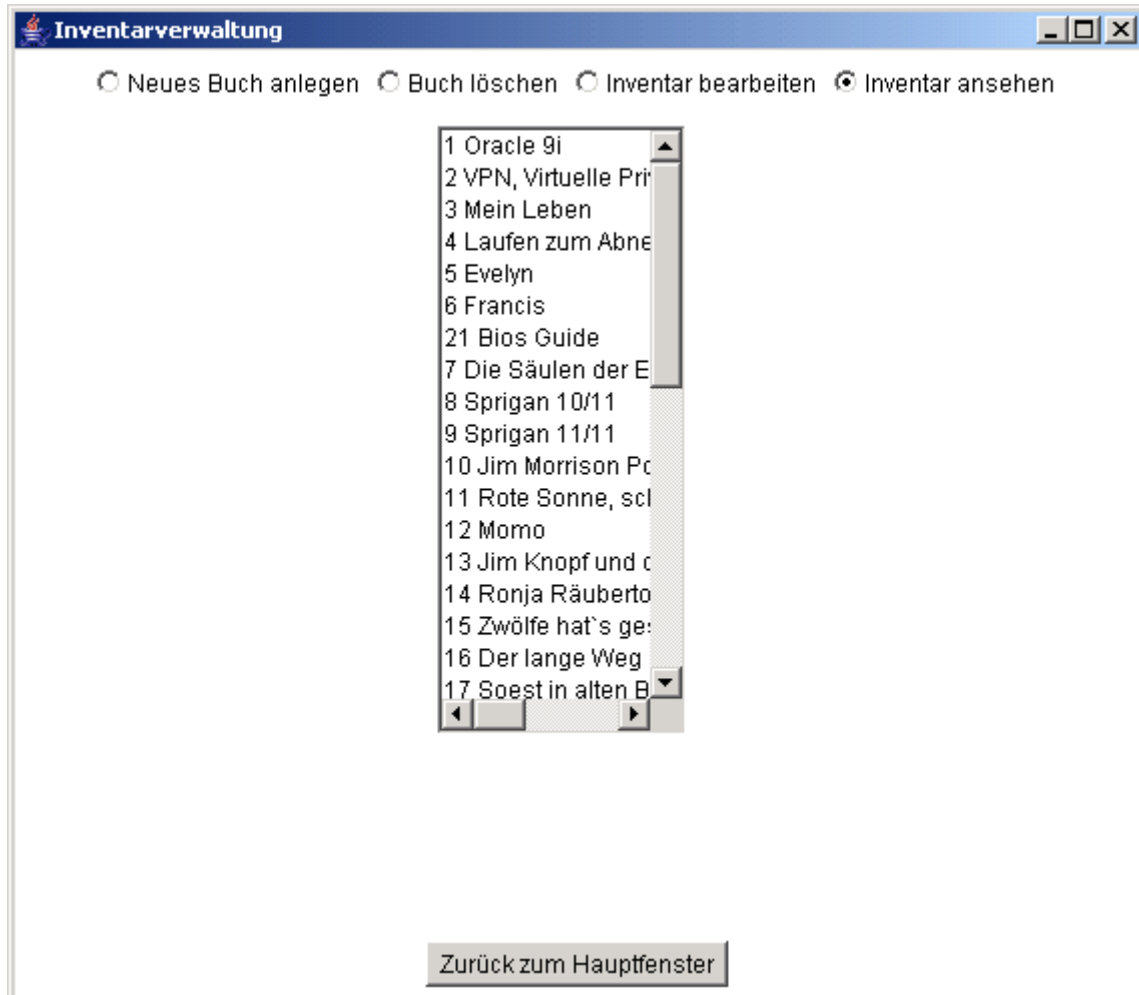
Um ein Buch zu bearbeiten, trägt man die Inventarnummer in das dafür vorgesehene Feld ein und klickt auf „Pruefen“. Daraufhin werden in die Textfelder die jeweiligen Werte übernommen. Diese Werte sind nun veränderbar. Sind alle gewünschten Änderungen vorgenommen, so klickt man auf „Änderungen übernehmen“ und der Datensatz wird überschrieben.

The screenshot shows a web application window titled "Inventarverwaltung". At the top, there are four radio buttons for navigation: "Neues Buch anlegen", "Buch löschen", "Inventar bearbeiten" (which is selected), and "Inventar ansehen". Below this, there are input fields for "Inventarnummer:", "Titel", and "Autor". A "Pruefen" button is positioned between the "Inventarnummer" and "Titel" fields. Below the "Titel" and "Autor" fields, there is a "Verlag" label and an input field, followed by an "Änderungen übernehmen" button. At the bottom of the form area, there is a "Zurück zum Hauptfenster" button. The entire form is enclosed in a light blue border.

(Abb.12)

4.5.4 Inventar ansehen

In dieser Maske befindet sich eine Liste, die das komplette Inventar anzeigt. Es werden jedoch nur die Werte Inventarnummer und Titel angezeigt.



(Abb.13)

4.6 Suchen

In diesem Fenster gibt es 4 verschiedenen Möglichkeiten etwas zu suchen. Es besteht die Möglichkeit, entweder nach der Kundennummer, nach dem Nachnamen, der Inventarnummer oder nach einem Buchtitel zu suchen. Hinter den jeweiligen Textfeldern befindet sich jeweils ein Button, welcher nach dem anklicken die Liste mit 3 Werten (Kundennummer, Name, Vorname) bzw. (Inventarnummer, Titel, Autor) füllt.

The screenshot shows a window titled "Suchen" with the following elements:

- Four search input fields, each with a "Suchen" button to its right:
 - "Hier zu suchende KUNDENNUMMER EINTRAGEN:"
 - "Hier zu suchende INVENTARNUMMER EINTRAGEN:"
 - "Hier zu suchenden KUNDENNAMEN EINTRAGEN:"
 - "Hier zu suchenden BUCHTITEL EINTRAGEN:"
- A large empty rectangular box for search results.
- A button labeled "Eingaben und Liste löschen" positioned to the right of the empty box.
- A button labeled "Zurück zum Hauptfenster" at the bottom center of the window.

(Abb.14)

5. Referenzen

5.1 Projektabschluss

Zum ausführen des Programms ist es erforderlich eine ODBC-Datenquelle in Windows anzulegen. Dazu geht man in der „Systemsteuerung“ auf „Verwalten“ und öffnet dort „Datenquellen (ODBC)“. Hier klickt man auf Hinzufügen und richtet die Datenbank als Datenquelle „Bibliothek“ ein. Erst dann kann das Programm eine Verbindung zur Datenbank herstellen und ist damit lauffähig.

5.2 Fazit

Das Projekt verlief durchgehend Problemlos, es entstanden keine Zwischenfälle, die den Verlauf hätten beeinträchtigen können. Die Gruppenmitglieder haben geschlossen einen wertvollen Einblick in die Programmiersprache Java erhalten.

5.3 Quellennachweiß

- Google (www.google.de)
- API-Referenz (<http://java.sun.com/j2se/1.4.2/docs/api/>)
- Java 4 U (Buch)
- SQL-Hilfe (http://home.tonline.de/home/aam_int/de/sqlhelp/sqldbefl.htm)

6. Anhang

- 45 Seiten Quellcode
- 3 Seiten Installationsanleitung

6.1 Anlage: Dokumentierter Quellcode

Aktion.java

```
import java.sql.*;           // Datenbankzugriffe importieren
import java.awt.*;          // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*;    // Ereignisse in den fenstern werden importiert
public class Aktion implements ActionListener
{
    private Gut f;
    public Aktion(Gut f)
    {
        this.f=f;
    }

    public void actionPerformed(ActionEvent e)           // ActionListener
    {
        if(e.getSource() == f.btnNewBorrow)           // Funktion dem Button hinzufügen
        {
            this.f.setVisible(false);
            NewLeihe f = new NewLeihe();
        }
//-----BUTTON NEUE AUSLEIHE ENDE-----
        if(e.getSource() == f.btnDelBorrow)           // Funktion dem Button hinzufügen
        {
            this.f.setVisible(false);
            DelLeihe f = new DelLeihe();
        }
//-----BUTTON DEL BORROW ENDE-----

        if(e.getSource() == f.btnUserAdmin)           // Funktion dem Button hinzufügen
        {
            this.f.setVisible(false);
            Verwaltung f = new Verwaltung();
        }
//-----BUTTON BENUTZERVERWALTUNG ENDE-----
        if(e.getSource() == f.btnInventar)           // Funktion dem Button hinzufügen
        {
            this.f.setVisible(false);
            Inventar f = new Inventar();
        }
//-----BUTTON Inventarisieren ENDE-----

        if(e.getSource() == f.btnSearch)           // Funktion dem Button hinzufügen
        {
            this.f.setVisible(false);
            Suchen f = new Suchen();
        }
//-----BUTTON SUCHEN ENDE-----
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Auswahl.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Auswahl implements ItemListener
{
    public Verwaltung f;
    public Auswahl(Verwaltung f)
    {
        this.f=f;
    }

    public void itemStateChanged(ItemEvent e)                //Itemlistener
    {
        if(e.getSource() == f.chkNeu)                    //Funktion des Itemlistener
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlUserNew, BorderLayout.CENTER);
            f.doLayout();
            f.pnlUserNew.doLayout();
            f.txtUserNewKdNr.setText("");
            f.txtUserNewName.setText("");
            f.txtUserNewVorname.setText("");
            f.txtUserNewStrasse.setText("");
            f.txtUserNewOrt.setText("");
            f.txtUserNewPLZ.setText("");
        }
        if(e.getSource() == f.chkDel)                    //Funktion des Itemlistener
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlUserDel, BorderLayout.CENTER);
            f.doLayout();
            f.pnlUserDel.doLayout();
            f.lstUserDelKd.removeAll();
            f.txtUserDelKdNr.setText("");
        }
        if(e.getSource() == f.chkEdit)                    //Funktion des Itemlistener
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlUserEdit, BorderLayout.CENTER);
            f.doLayout();
            f.pnlUserEdit.doLayout();
            f.txtUserEditKdNr.setText("");
            f.txtUserEditName.setText("");
            f.txtUserEditVorname.setText("");
            f.txtUserEditStrasse.setText("");
            f.txtUserEditOrt.setText("");
            f.txtUserEditPLZ.setText("");
        }
        if(e.getSource() == f.chkShow)                    //Funktion des Itemlistener
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
f.add(f.pnlUnten, BorderLayout.SOUTH);
f.add(f.pnlKundenstamm, BorderLayout.CENTER);
f.doLayout();
    f.lstKdStamm.removeAll();
f.pnlKundenstamm.doLayout();
f.pnlKdSt.doLayout();
Connection dbVerbindung=null;
Statement sqlStatement=null;
try
{
    dbVerbindung =
    DriverManager.getConnection("jdbc:odbc:bibliothek","",""); //Treiber für Datenbank laden
    System.out.println("Verbindung erfolgreich...");
}

catch(SQLException wert)
{
    wert.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+wert);
    System.exit(0);
}

try
{
    sqlStatement=dbVerbindung.createStatement();
    ResultSet result=sqlStatement.executeQuery("SELECT KdNr, Name FROM Kunden");
    while (result.next())
    {
        f.lstKdStamm.add(""+String.valueOf(result.getInt(1))+" "+String.valueOf(result.getString(2))+");");
    }
    result.close();
    sqlStatement.close();
    dbVerbindung.close();
}

    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+sql);
        System.exit(0);
    }

}

}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Beenden.java

```
import java.awt.event.*; // Für Java Windows Ereignisse importieren
public class Beenden extends WindowAdapter // Klasse Beenden erstellen welche
Funkionsbuttons enthält
{
    public void windowClosing(WindowEvent e) //Funktion zum Schließen des Fensters
    {
        System.exit(0);
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

DelBtn.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class DelBtn implements ActionListener
{
    private DelLeihe f;
    public DelBtn(DelLeihe f)
    {
        this.f=f;
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == f.btnNewBack)
        {
            this.f.setVisible(false);
            Gut f = new Gut();
        }
        if(e.getSource() == f.btnNewPruef)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung = DriverManager.getConnection("jdbc:odbc:bibliothek","","");
                }
                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }
                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT Name, Strasse, Ort FROM Kunden
WHERE Kunden.KdNr = "+f.txtNewKdNr.getText()+"");
                    while (result.next())
                    {
                        f.lstNewBk.removeAll();
                        f.lstNewBk.add(String.valueOf(result.getString(1)));
                        f.lstNewBk.add(String.valueOf(result.getString(2)));
                        f.lstNewBk.add(String.valueOf(result.getString(3)));
                    }
                    result.close();
                    ResultSet Buchresult=sqlStatement.executeQuery("SELECT Titel, Autor, Verlag
FROM Inventar WHERE Inventar.InvNr = "+f.txtNewISBN.getText()+"");
                    while (Buchresult.next())
                    {
                        f.lstNewKd.removeAll();
                        f.lstNewKd.add(String.valueOf(Buchresult.getString(1)));
                        f.lstNewKd.add(String.valueOf(Buchresult.getString(2)));
                        f.lstNewKd.add(String.valueOf(Buchresult.getString(3)));
                    }
                    Buchresult.close();
                    sqlStatement.close();
                    dbVerbindung.close();
                }
                catch (SQLException sql)
            }
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        {
            sql.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+sql);
            System.exit(0);
        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0);
    }

    }
    if (e.getSource() == f.btnNew)
    {
        try
        {
            Connection dbVerbindung=null;
            Statement sqlStatement=null;

            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");
            }

            catch(SQLException q)
            {
                q.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+q);
                System.exit(0);
            }

            try
            {
                sqlStatement=dbVerbindung.createStatement();
                String StrKdNR = f.txtNewKdNr.getText();
                String StrInvNr = f.txtNewISBN.getText();
                String Anlegen = "DELETE FROM Ausleihe WHERE KdNr =
                "+StrKdNR+" AND InvNr = "+StrInvNr+"";
                int Anlage=sqlStatement.executeUpdate(Anlegen);
                if (Anlage == 0)
                {
                }
                else
                {
                }
            }
            sqlStatement.close();
            dbVerbindung.close();
        }
    }
    catch(SQLException bvb)
    {
        bvb.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+bvb);
        System.exit(0);
    }
    f.lstNewBk.removeAll();
    f.lstNewKd.removeAll();
    f.txtNewISBN.setText("");
    f.txtNewKdNr.setText("");

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
}  
catch (Exception dixi)  
{  
    dixi.printStackTrace();  
    System.out.println("DB-Verbindung Fehler: "+dixi);  
    System.exit(0);  
}  
}  
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

DelLeihe.java

```
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*; // Ereignisse in
den fenstern werden importiert
public class DelLeihe extends Frame
{
    private Beenden //
Funktion Beenden einleiten (Der Antworter auf den Hörer)
    Beenden = new Beenden(); //
neues Objekt Beenden erstellen und mit Klasse Beenden verknüpfen

    private DelBtn
    // Für die Klasse in der der Buttonhandler drin ist
    DelBtn = new DelBtn(this);

    public Panel pnlUnten = new Panel();
    // Panel wird erstellt
    public Button btnNewBack = new Button("Zurück zum Hauptfenster"); // Button wird erstellt
    public Button btnNew = new Button("Leihgegenstand austragen");// Button wird erstellt

    public Panel pnlNewLend = new Panel(new FlowLayout(FlowLayout.CENTER));
    public Button btnNewPruef = new Button("Pruefen"); //
Button wird erstellt
    public TextField txtNewKdNr = new TextField ("",15);
    public TextField txtNewISBN = new TextField ("",15);
    public Label lblNewKdNr = new Label ("Kundennummer: ");
    public Label lblNewISBN = new Label ("Inventarnr. : ");
    public Label lblNewListKd = new Label ("Der Kunde ist: ");
    public Label lblNewListBk = new Label ("Leihgegenstand ist : ");
    public List lstNewKd = new List(4,true);
    public List lstNewBk = new List(4,true);

    public DelLeihe()
    // enthält Fenstereigenschaften
    {
        super("Leihgegenstand austragen"); // "Super" ist Befehl für die "Caption" -> VB!!
        this.setSize(555,500); // Setzt die Größe des Fensters fest (links,oben)

        this.pnlUnten.add(this.btnNewBack); // Button wird dem Panel zugeordnet
        this.pnlUnten.add(this.btnNew); // Button wird dem Panel zugeordnet

        this.pnlNewLend.add(this.lblNewKdNr);
        this.pnlNewLend.add(this.txtNewKdNr);
        this.pnlNewLend.add(this.lblNewISBN);
        this.pnlNewLend.add(this.txtNewISBN);
        this.pnlNewLend.add(this.btnNewPruef);

        this.pnlNewLend.add(this.lblNewListKd);
        this.pnlNewLend.add(this.lstNewBk);

        this.pnlNewLend.add(this.lblNewListBk);
        this.pnlNewLend.add(this.lstNewKd);

        this.add(this.pnlUnten, BorderLayout.SOUTH); // Panel wird an das Fenster
übergeben und südlich ausgerichtet
        this.add(this.pnlNewLend, BorderLayout.CENTER);
        this.show(); // DAS wird angezeigt (Visible = true)
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich Funktion getSize darin
befindet
    this.setLocation((d.width - getSize().width) / 2, // Wird auf die Bildschirmmitte gesetzt
(d.height- getSize().height) / 2 );
    this.addWindowListener(Beenden);           // Dem Fenster den Hörer anhängen und
mit Beenden verknüpfen
    this.btnNew.addActionListener(DelBtn);
    this.btnNewBack.addActionListener(DelBtn);
    this.btnNewPruef.addActionListener(DelBtn);
}

public static void main (String argv[])           // Ausführende Sequenz starten
{
    DelLeihe fenster = new DelLeihe();           // Klasse Fenster wird
als neues Object auf den Bildschirm geladen
}
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Gut.java

```
import java.sql.*;
    // Datenbankzugriffe importieren
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*;
    // Ereignisse in
den fenstern werden importiert
import java.io.*;
    //
Input Output importieren
public class Gut extends Frame
{
    public int counter = 0;
    //
Variable counter wird als integer deklariert und auf 0 gesetzt

//-----Importe Variable und Klasse-----

    private Panel pnlGutbtns = new Panel();
    private Panel pnlListBox = new Panel(new FlowLayout(FlowLayout.CENTER));
    private Panel pnlOben = new Panel(new FlowLayout(FlowLayout.CENTER));
    private Beenden
    //
Funktion Beenden einleiten (Der Antworter auf den Hörer)
    Beenden = new Beenden();
    //
neues Objekt Beenden erstellen und Klasse Beenden verknüpfen
    private Aktion
    // Für die Klasse in der der Button drin ist
    Aktion = new Aktion(this);

    public List lstListe = new List(15,false);
    // Neue Liste mit 40 Feldern die
mehrfach gewählt werden können
    public Label lblHeadln = new Label("Bücher und Kundenverwaltung von MicroBallZ(TM)");
    public Label lblListUb = new Label("Heute fällige Rückgaben (Name || Titel) :");
    public Button btnNewBorrow = new Button("Ausleihe eintragen");
    public Button btnDelBorrow = new Button("Ausleihe austragen");
    public Button btnUserAdmin = new Button("Benutzerverwaltung");
    public Button btnInventar = new Button("Inventar");
    public Button btnSearch = new Button("Suchen");

//-----Aufruf fürs Beenden des Fensters und die Buttons-----
-

    public Gut()
    // öffnet die Grafik Fenster
    {
        super("Bücher und Benutzer Verwaltung (TM) "); // "Super" ist Befehl für die
"Caption" -> VB!!
        this.pnlGutbtns.add(this.btnNewBorrow);
        //      Fügt   Button
dem Panel hinzu
        this.pnlGutbtns.add(this.btnDelBorrow);
        //      Fügt   Button
dem Panel hinzu
        this.pnlGutbtns.add(this.btnUserAdmin);
        //      Fügt   Button
dem Panel hinzu
        this.pnlGutbtns.add(this.btnInventar);
        //      Fügt   Button
dem Panel hinzu
        this.pnlGutbtns.add(this.btnSearch);
        //      Fügt   Button
dem Panel hinzu
        this.pnlListBox.add(this.lblListUb);
        this.pnlListBox.add(this.lstListe);
        this.pnlOben.add(this.lblHeadln);
        this.add(this.pnlGutbtns, BorderLayout.SOUTH); // Fügt das Panel dem Fenster
hinzu
        this.add(this.pnlListBox, BorderLayout.CENTER);
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        this.add(this.pnlOben, BorderLayout.NORTH);
        this.setSize(510,500); // Setzt
die Größe des Fensters fest (links,oben)
        this.show();
        // DAS wird angezeigt (Visible = true)

Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich Funktion getSize darin
befindet
        this.setLocation((d.width - getSize().width ) / 2, // Wird auf die Bildschirmmitte gesetzt
(d.height- getSize().height) / 2 ); // Wird auf die Bildschirmmitte gesetzt

        this.addWindowListener(Beenden); // Dem Fenster
die Klasse Beenden übergeben
        this.btnNewBorrow.addActionListener(Aktion); // Die Buttons auf die Klasse
Aktion verweisen
        this.btnDelBorrow.addActionListener(Aktion);
        this.btnUserAdmin.addActionListener(Aktion);
        this.btnInventar.addActionListener(Aktion);
        this.btnSearch.addActionListener(Aktion);

        dbConnection();
    }

public void dbConnection()
{
    Connection dbVerbindung=null;
    Statement sqlStatement=null;
    ResultSet resultSet=null;

//-----Datenbankname und SQL-Anbindung-----
-----

    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); //Lade von sun den treiber für
Java - Open DBC
        System.out.println("Treiber erfolgreich geladen..."); // Ausgabe bei erfolgreichem Laden in der Console
    }
        //try schließen

    catch(ClassNotFoundException e)
    {
        System.out.println("Fehler beim laden des Treibers "
            +e); //Wird an der console
ausgegeben bei fehlerhafter verbindung
        System.exit(0);
        //Abbruch des Programm bei fehlerhaftem Laden des Treibers
    }
        // catch(ClassNotFoundException e) schließen

//-----Laden des Treibers (mit Exceptionhandler)-----
--

    try
    {
        dbVerbindung =
        DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER EINMAL DIE DATENBANK
ÄNDERN!!!!
        System.out.println("Verbindung erfolgreich..."); //try schließen
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
catch(SQLException e)
{
    e.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+e);
    System.exit(0);
}

// catch(SQLException e) schließen

try
{
    sqlStatement=dbVerbindung.createStatement();

    ResultSet result=sqlStatement.executeQuery("SELECT Kunden.Name, Inventar.Titel FROM
Ausleihe, Inventar, Kunden WHERE Ausleihe.Rueckgabe = date() AND Ausleihe.KdNr = Kunden.KdNr AND
Ausleihe.InvNr = Inventar.InvNr");

    while (result.next())
    {
        IstListe.add(""+String.valueOf(result.getString(1))+
"+String.valueOf(result.getString(2))+""");
        //System.out.println("Kundennummer: "+result.getInt(1));
    }

    result.close();
    sqlStatement.close();
    dbVerbindung.close();
}
catch (SQLException sql)
{
    sql.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+sql);
    System.exit(0);
}

public static void main (String argv[]) // Ausführende Sequenz
starten
{
    Gut fenster = new Gut();
        // Klasse Fenster wird als neues Object auf den
        // Bildschirm geladen

//-----Gibt das Fenster aus-----

}

// Schließt das aufführende Ding
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Inventar.java

```
import java.sql.*;
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*;
                                                    // Ereignisse in den fenstern

werden importiert
public class Inventar extends Frame
{
    private Beenden
    Funktion Beenden einleiten (Der Antworter auf den Hörer)
        Beenden = new Beenden();
    neues Objekt Beenden erstellen und mit Klasse Beenden verknüpfen

    private InventBtn
    die Klasse in der der Buttonhandler drin ist
        InventBtn = new InventBtn(this);

//-----Hauptfenster anfang-----

    public CheckboxGroup Wahl = new CheckboxGroup();

    public Checkbox chkNeu = new Checkbox ("Neues Buch anlegen",false,Wahl);    // Erstellen von
nicht gewählter Checkbox
    public Checkbox chkDel = new Checkbox ("Buch löschen",false,Wahl);        // welche
Mitglied in der Gruppe Wahl ist
    public Checkbox chkEdit = new Checkbox ("Inventar bearbeiten",false,Wahl); //
    public Checkbox chkShow = new Checkbox ("Inventar ansehen",false,Wahl); //

    public Panel pnlOben = new Panel(new FlowLayout(FlowLayout.CENTER));      // neues Panel
einrichten
    public Panel pnlUnten = new Panel(new FlowLayout(FlowLayout.CENTER));
    public Button btnInvent = new Button("Inventarisieren");                  // Button wird
erstellt

//-----Hauptfenster Ende-----
//-----Panel Invent anlegen Anfang-----

    public Panel pnlInventNew = new Panel(new FlowLayout(FlowLayout.CENTER));

        public Label lblInventNewKdNr = new Label ("Inventarnummer:");
// erstellt label
        public Label lblInventNewName = new Label ("Titel:");
// erstellt label
        public Label lblInventNewVorname = new Label ("Autor:");
// erstellt label
        public Label lblInventNewStrasse = new Label ("Verlag:");              // erstellt label

        public TextField txtInventNewKdNr = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventNewName = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventNewVorname = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventNewStrasse = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventNewOrt = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventNewPLZ = new TextField("",10);
//erstellt Textfeld
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        public Button btnInventNewSave = new Button ("Buch anlegen");
//erstellt Button

//-----Panel Invent anlegen Ende-----
//-----Panel Invent Löschen Anfang-----

        public Panel pnlInventDel = new Panel(new FlowLayout(FlowLayout.CENTER)); // Panel
erstellen mit LayoutManager

        public TextField txtInventDelKdNr = new TextField("",10); //
erstellt Textfeld

        public Label lblInventDelKdNr = new Label ("Inventarnummer:");
// erstellt label
        public Label lblInventDelKdIs = new Label ("Buch ist:"); //
erstellt label

        public Button btnInventDel = new Button ("Aus Inventar entfernen"); // erstellt
Button

        public Button btnInventDelChk = new Button ("Pruefen");
// erstellt Button

        public List lstInventDelKd = new List(4,true);
// Neue Liste mit 40 Feldern die mehrfach gewählt werden können

//-----Panel Invent löschen Ende-----
//-----Panel Invent editieren Anfang-----

        public Panel pnlInventEdit = new Panel(new FlowLayout(FlowLayout.CENTER));

        public Label lblInventEditKdNr = new Label ("Inventarnummer:");

        public Label lblInventEditName = new Label ("Titel");
// erstellt label
        public Label lblInventEditVorname = new Label ("Autor");
// erstellt label
        public Label lblInventEditStrasse = new Label ("Verlag"); // erstellt label

        public TextField txtInventEditKdNr = new TextField("",10);
                public TextField txtInventEditName = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventEditVorname = new TextField("",10);
//erstellt Textfeld
        public TextField txtInventEditStrasse = new TextField("",10);
//erstellt Textfeld

        public Button btnInventEditChk = new Button ("Pruefen"); //
erstellt Button

        public Button btnInventEdit = new Button ("Änderungen übernehmen"); //
erstellt Button

//-----Panel Invent editieren Ende-----
//-----Panel Kundenstamm Anfang-----

        public Panel pnlKundenstamm = new Panel(new FlowLayout(FlowLayout.CENTER));
        public List lstKdStamm = new List(20,true); // Neue Liste
mit 40 Feldern die mehrfach gewählt werden können
        public Panel pnlKdSt = new Panel(new FlowLayout(FlowLayout.CENTER));

//-----Panel Kundenstamm Ende-----
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
public Button btnInventBack = new Button ("Zurück zum Hauptfenster"); //
Button für das Hauptfenster

private InventAusw
InventAusw = new InventAusw(this);
// Itemlistener integrieren

//-----Aufruf fürs Beenden des Fensters und die Buttons-----
-

public Inventar() //
öffnet die Grafik Fenster
{
    super("Inventarverwaltung"); // "Super" ist
Befehl für die "Caption" -> VB!!

//-----Panel Benutzer hinzufügen-----

hinzufügen      this.pnlInventNew.add(this.lblInventNewKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventNew.add(this.txtInventNewKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventNew.add(this.lblInventNewName); // Objekt dem Panel
hinzufügen      this.pnlInventNew.add(this.txtInventNewName); // Objekt dem Panel
hinzufügen      this.pnlInventNew.add(this.lblInventNewVorname); // Objekt dem Panel hinzufügen
hinzufügen      this.pnlInventNew.add(this.txtInventNewVorname); // Objekt dem Panel hinzufügen
hinzufügen      this.pnlInventNew.add(this.lblInventNewStrasse); // Objekt dem Panel hinzufügen
hinzufügen      this.pnlInventNew.add(this.txtInventNewStrasse); // Objekt dem Panel hinzufügen
hinzufügen      this.pnlInventNew.add(this.btnInventNewSave); // Objekt dem Panel

//-----Panel Benutzer löschen-----

hinzufügen      this.pnlInventDel.add(this.lblInventDelKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventDel.add(this.txtInventDelKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventDel.add(this.btnInventDelChk); // Objekt dem Panel
hinzufügen      this.pnlInventDel.add(this.lblInventDelKdIs); // Objekt dem Panel
hinzufügen      this.pnlInventDel.add(this.lstInventDelKd); // Objekt dem Panel
hinzufügen      this.pnlInventDel.add(this.btnInventDel); // Objekt dem Panel

//-----Panel Benutzer editieren-----

hinzufügen      this.pnlInventEdit.add(this.lblInventEditKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventEdit.add(this.txtInventEditKdNr); // Objekt dem Panel
hinzufügen      this.pnlInventEdit.add(this.btnInventEditChk); // Objekt dem Panel
hinzufügen      this.pnlInventEdit.add(this.lblInventEditName); // Objekt dem Panel
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        this.pnlInventEdit.add(this.txtInventEditName);           // Objekt dem Panel
hinzufoegen
        this.pnlInventEdit.add(this.lblInventEditVorname); // Objekt dem Panel hinzufoegen
        this.pnlInventEdit.add(this.txtInventEditVorname); // Objekt dem Panel hinzufoegen
        this.pnlInventEdit.add(this.lblInventEditStrasse); // Objekt dem Panel hinzufoegen
        this.pnlInventEdit.add(this.txtInventEditStrasse); // Objekt dem Panel hinzufoegen
        this.pnlInventEdit.add(this.btnInventEdit);           // Objekt dem Panel
hinzufoegen

//-----Panel Kundenstamm-----

        this.pnlKundenstamm.add(this.lstKdStamm, BorderLayout.NORTH); //
Objekt dem Panel hinzufoegen
        this.pnlKundenstamm.add(this.pnlKdSt);

//-----HAUPTFENSTER-----

        this.pnlOben.add(this.chkNeu);                           // Objekt dem
Panel hinzufoegen
        this.pnlOben.add(this.chkDel);                           // Objekt dem
Panel hinzufoegen
        this.pnlOben.add(this.chkEdit);                           // Objekt dem
Panel hinzufoegen
        this.pnlOben.add(this.chkShow);                           // Objekt dem
Panel hinzufoegen
        this.pnlUnten.add(this.btnInventBack);                    // Objekt dem Panel
hinzufoegen
        this.add(this.pnlOben, BorderLayout.NORTH);                // Panel dem Fenster
hinzufoegen
        this.add(this.pnlUnten, BorderLayout.SOUTH);               // Panel dem Fenster hinzufoegen

        this.setSize(570,500);                                     // Setzt
die Groeoe des Fensters fest (links,oben)
        this.show();
        // DAS wird angezeigt (Visible = true)

Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich Funktion getSize darin
befindet
        this.setLocation((d.width - getSize().width) / 2, // Wird auf die Bildschirmmitte gesetzt
(d.height- getSize().height) / 2); // Wird auf die Bildschirmmitte gesetzt

        this.addWindowListener(Beenden);                          // Dem Fenster
den Hoereranhaengen und mit Beenden verknuepfen

        this.chkNeu.addItemListener(InventAusw);                 // Itemlistener mit
Objekt verknuepfen
        this.chkDel.addItemListener(InventAusw);                 // Itemlistener mit
Objekt verknuepfen
        this.chkEdit.addItemListener(InventAusw);                // Itemlistener mit
Objekt verknuepfen
        this.chkShow.addItemListener(InventAusw);                // Itemlistener
mit Objekt verknuepfen

        this.btnInventBack.addActionListener(InventBtn); // ActionListener mit Objekt
verknuepfen
        this.btnInventDel.addActionListener(InventBtn); // ActionListener mit
Objekt verknuepfen
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        this.btnInventDelChk.addActionListener(InventBtn);           //  
ActionListener mit Objekt verknüpfen  
        this.btnInventEdit.addActionListener(InventBtn); // ActionListener mit Objekt  
verknüpfen  
        this.btnInventEditChk.addActionListener(InventBtn); // ActionListener mit  
Objekt verknüpfen  
        this.btnInventNewSave.addActionListener(InventBtn); // ActionListener mit  
Objekt verknüpfen  
    }  
    public static void main (String argv[]) // Ausführende Sequenz starten  
    {  
        Inventar fenster = new Inventar(); // Klasse Fenster wird als neues  
Object auf den Bildschirm geladen  
    }  
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

InventAusw.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class InventAusw implements ItemListener
{
    public Inventar f;
    public InventAusw(Inventar f)
    {
        this.f=f;
    }
    public void itemStateChanged(ItemEvent e)                //ItemIstener
    {
        if(e.getSource() == f.chkNeu)
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlInventNew, BorderLayout.CENTER);
            f.doLayout();
            f.pnlInventNew.doLayout();
        }
        if(e.getSource() == f.chkDel)
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlInventDel, BorderLayout.CENTER);
            f.doLayout();
            f.pnlInventDel.doLayout();
        }
        if(e.getSource() == f.chkEdit)
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlInventEdit, BorderLayout.CENTER);
            f.doLayout();
            f.pnlInventEdit.doLayout();
        }
        if(e.getSource() == f.chkShow)
        {
            f.removeAll();
            f.add(f.pnlOben, BorderLayout.NORTH);
            f.add(f.pnlUnten, BorderLayout.SOUTH);
            f.add(f.pnlKundenstamm, BorderLayout.CENTER);
            f.doLayout();
            f.lstKdStamm.removeAll();
            f.pnlKundenstamm.doLayout();
            f.pnlKdSt.doLayout();
            Connection dbVerbindung=null;
            Statement sqlStatement=null;
            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");//          HIER
                EINMAL DIE DATENBANK ÄNDERN!!!!
                System.out.println("Verbindung erfolgreich...");                //try
            }
            schließen
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        catch(SQLException wert)
        {
            wert.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+wert);
            System.exit(0);
        }

        // catch(SQLException e) schließen
        try
        {
            sqlStatement=dbVerbindung.createStatement();
            ResultSet result=sqlStatement.executeQuery("SELECT InvNr, Titel FROM
Inventar");
            while (result.next())
            {
                f.lstKdStamm.add(""+String.valueOf(result.getInt(1))+
"+String.valueOf(result.getString(2))+""");
            }
            result.close();
            sqlStatement.close();
            dbVerbindung.close();
        }
        catch (SQLException sql)
        {
            sql.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+sql);
            System.exit(0);
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

InventBtn.java

```
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
public class InventBtn implements ActionListener
{
    private Inventar f;
    public InventBtn(Inventar f)
    {
        this.f=f;
    }

    public void actionPerformed(ActionEvent e)
    //Actionlistener
    {
        if(e.getSource() == f.btnInventBack)
        {
            this.f.setVisible(false);
            Gut f = new Gut();
        }
    //-----BUTTON NEUES INVENTAR-----
        if(e.getSource() == f.btnInventNewSave)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                    EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen
                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    String Anlegen = "INSERT INTO Inventar (InvNr, Titel, Autor, Verlag)
VALUES ("+f.txtInventNewKdNr.getText()+", "+f.txtInventNewName.getText()+",
"+f.txtInventNewVorname.getText()+", "+f.txtInventNewStrasse.getText()+")";
                    System.out.println (""+Anlegen);
                    int Anlage=sqlStatement.executeUpdate(Anlegen);
                    if (Anlage == 0)
                    {
                    }
                    else
                    {
                        System.out.println ("Kunde wurde angelegt mit "+Anlage+
"Datensätzen");
                    }
                }
                sqlStatement.close();
            }
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        dbVerbindung.close();
    }

    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("SQL-Eingabefehler: "+sql);
        System.exit(0);
    }
}

catch (Exception k)
{
    k.printStackTrace();
    System.out.println("Irgendwas ist Mist!" +k);
    System.exit(0);
}
}

//-----BUTTON KUNDEN ENTFERNEN - CHECK-----
if(e.getSource() == f.btnInventDelChk)
{
    try
    {
        Connection dbVerbindung=null;
        Statement sqlStatement=null;

        try
        {
            dbVerbindung =
            DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
            EINMAL DIE DATENBANK ÄNDERN!!!!
        }

        catch(SQLException q)
        {
            q.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+q);
            System.exit(0);
        }

        // catch(SQLException e) schließen

        try
        {
            sqlStatement=dbVerbindung.createStatement();
            ResultSet result=sqlStatement.executeQuery("SELECT Titel, Autor, Verlag
            FROM Inventar WHERE InvNr = "+f.txtInventDelKdNr.getText()+"");
            while (result.next())
            {
                f.lstInventDelKd.add(String.valueOf(result.getString(1)));
                f.lstInventDelKd.add(String.valueOf(result.getString(2)));
                f.lstInventDelKd.add(String.valueOf(result.getString(3)));
                //System.out.println("Kundennummer: "+result.getInt(1));
            }
            result.close();
            sqlStatement.close();
            dbVerbindung.close();
        }
        catch (SQLException sql)
        {
            sql.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+sql);
            System.exit(0);
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0); // Button
    }
}
//-----BUTTON Kunde Löschen-----

if (e.getSource() == f.btnInventDel)
{
    try
    {
        Connection dbVerbindung=null;
        Statement sqlStatement=null;

        try
        {
            dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
            EINMAL DIE DATENBANK ÄNDERN!!!!
        }

        catch(SQLException q)
        {
            q.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+q);
            System.exit(0);
        }

        // catch(SQLException e) schließen

        try
        {
            sqlStatement=dbVerbindung.createStatement();
            String StrKdNR = f.txtInventDelKdNr.getText();
            String Anlegen = "DELETE FROM Inventar WHERE InvNr =
"+StrKdNR+"";

            int Anlage=sqlStatement.executeUpdate(Anlegen);
            if (Anlage == 0)
            {
            }
            else
            {
                System.out.println ("Buch "+StrKdNR+" mit "+Anlage+
                "Datensätzen gelöscht!");
            }

            sqlStatement.close();
            dbVerbindung.close();

        }
        catch(SQLException bvb)
        {
            bvb.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+bvb);
            System.exit(0);
        }
        f.lstInventDelKd.removeAll();
        f.txtInventDelKdNr.setText("");
    }
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        } // Verbindung
    catch (Exception BtnDeleten)
    {
        BtnDeleten.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+BtnDeleten);
        System.exit(0);
    }
    // If Anweisung Button von Benutzern!!!!
//-----BUTTON EDIT CHECK-----
    if(e.getSource() == f.btnInventEditChk)
    {
        try
        {
            Connection dbVerbindung=null;
            Statement sqlStatement=null;

            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                EINMAL DIE DATENBANK ÄNDERN!!!!
            }

            catch(SQLException q)
            {
                q.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+q);
                System.exit(0);
            }

            // catch(SQLException e) schließen

            try
            {
                sqlStatement=dbVerbindung.createStatement();
                ResultSet result=sqlStatement.executeQuery("SELECT Titel, Autor, Verlag
                FROM Inventar WHERE InvNr = "+f.txtInventEditKdNr.getText()+"");
                while (result.next())
                {

                    f.txtInventEditName.setText(""+String.valueOf(result.getString(1)));

                    f.txtInventEditVorname.setText(""+String.valueOf(result.getString(2)));

                    f.txtInventEditStrasse.setText(""+String.valueOf(result.getString(3)));
                }
                result.close();
                sqlStatement.close();
                dbVerbindung.close();
            }
            catch (SQLException sql)
            {
                sql.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+sql);
                System.exit(0);
            }
        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0); // Button
    }
}
Anlegen von Benutzern
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

    }
//-----BUTTON BEARBEITEN-----
    if(e.getSource() == f.btnInventEdit)
    {
        try
        {
            Connection dbVerbindung=null;
            Statement sqlStatement=null;

            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                EINMAL DIE DATENBANK ÄNDERN!!!!
            }

            catch(SQLException q)
            {
                q.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+q);
                System.exit(0);
            }

            // catch(SQLException e) schließen

            try
            {
                sqlStatement=dbVerbindung.createStatement();
                int result=sqlStatement.executeUpdate("UPDATE Inventar SET Titel =
                "+f.txtInventEditName.getText()+"", Autor = "+f.txtInventEditVorname.getText()+"",Verlag =
                "+f.txtInventEditStrasse.getText()+" WHERE InvNr = "+f.txtInventEditKdNr.getText()+"");
                sqlStatement.close();
                dbVerbindung.close();
            }
            catch (SQLException sql)
            {
                sql.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+sql);
                System.exit(0);
            }
            f.txtInventEditKdNr.setText("");
            f.txtInventEditName.setText("");
            f.txtInventEditVorname.setText("");
            f.txtInventEditStrasse.setText("");
        }
        catch (Exception Pruef)
        {
            Pruef.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+Pruef);
            System.exit(0); // Button
        }
    }
}
/*
System.out.println("UPDATE Kunden SET Name = "+f.txtUserEditName.getText()+"",
Vorname = "+f.txtUserEditVorname.getText()+"",Strasse = "+f.txtUserEditStrasse.getText()+"", PLZ =
"+f.txtUserEditPLZ.getText()+"", Ort = "+f.txtUserEditOrt.getText()+" WHERE KdNr =
"+f.txtUserEditKdNr.getText()+"");
*/
}
//-----
} // Buttons gesamt
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

NewBtn.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class NewBtn implements ActionListener
{
    private NewLeihe f;
    public NewBtn(NewLeihe f)
    {
        this.f=f;
    }

    public void actionPerformed(ActionEvent e) //ActionListener
    {
        if(e.getSource() == f.btnNewBack)
        {
            this.f.setVisible(false);
            Gut f = new Gut();
        }
//-----BUTTON NEUE AUSLEIHE ENDE-----
        if(e.getSource() == f.btnNewPruef)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen
                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT Name, Strasse, Ort
FROM Kunden WHERE Kunden.KdNr = "+f.txtNewKdNr.getText()+"");
                    while (result.next())
                    {
                        f.lstNewBk.add(String.valueOf(result.getString(1)));
                        f.lstNewBk.add(String.valueOf(result.getString(2)));
                        f.lstNewBk.add(String.valueOf(result.getString(3)));
                        //System.out.println("Kundennummer: "+result.getInt(1));
                    }
                    result.close();
                    ResultSet Buchresult=sqlStatement.executeQuery("SELECT Titel, Autor, Verlag
FROM Inventar WHERE Inventar.InvNr = "+f.txtNewISBN.getText()+"");
                    while (Buchresult.next())
                    {
                        f.lstNewKd.add(String.valueOf(Buchresult.getString(1)));
                    }
                }
            }
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        f.lstNewKd.add(String.valueOf(Buchresult.getString(2)));
        f.lstNewKd.add(String.valueOf(Buchresult.getString(3)));
        //System.out.println("Kundennummer: "+result.getInt(1));
    }
    Buchresult.close();
    sqlStatement.close();
    dbVerbindung.close();
    }
    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+sql);
        System.exit(0);
    }
}
catch (Exception Pruef)
{
    Pruef.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+Pruef);
    System.exit(0);    // Button Anlegen von Benutzern
}
}
if (e.getSource() == f.btnNew)
{
    try
    {
        Connection dbVerbindung=null;
        Statement sqlStatement=null;

        try
        {
            dbVerbindung =
            DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
            EINMAL DIE DATENBANK ÄNDERN!!!!
        }

        catch(SQLException q)
        {
            q.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+q);
            System.exit(0);
        }

        // catch(SQLException e) schließen
        try
        {
            sqlStatement=dbVerbindung.createStatement();
            String Anlegen = "INSERT INTO Ausleihe (InvNr, KdNr, Rueckgabe,
Ausleihe) VALUES (" +f.txtNewISBN.getText()+", "+f.txtNewKdNr.getText()+", date()+14, date())";
            //System.out.println("INSERT INTO Ausleihe (InvNr, KdNr,
Rückgabe, Ausleihe) VALUES (" +f.txtNewISBN.getText()+", "+f.txtNewKdNr.getText()+", date()+14,
NOW())");

            int Anlage=sqlStatement.executeUpdate(Anlegen);
            if (Anlage == 0)
            {
            }
            else
            {
                System.out.println ("Kunde wurde angelegt mit "+Anlage+
"Datensätzen");
            }
            sqlStatement.close();
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        dbVerbindung.close();
    }
    catch(SQLException bvb)
    {
        bvb.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+bvb);
        System.exit(0);
    }
    f.lstNewBk.removeAll();
    f.lstNewKd.removeAll();
    f.txtNewISBN.setText("");
    f.txtNewKdNr.setText("");
    // If Anweisung Button Anlegen von Benutzern!!!!
}
catch (Exception dixi)
{
    dixi.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+dixi);
    System.exit(0);
} // Buttons gesamt
}
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

NewLeihe.java

```
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*; // Ereignisse in
den fenstern werden importiert
public class NewLeihe extends Frame
{
    private Beenden //
Funktion Beenden einleiten (Der Antworter auf den Hörer) //
    Beenden = new Beenden(); //
neues Objekt Beenden erstellen und mit Klasse Beenden verknüpfen

    private NewBtn
    // Für die Klasse in der der Buttonhandler drin ist
    NewBtn = new NewBtn(this);

    public Panel pnlUnten = new Panel();
    // Panel wird erstellt
    public Button btnNewBack = new Button("Zurück zum Hauptfenster"); // Button wird erstellt
    public Button btnNew = new Button("Neuen Leihgegenstand eintragen");// Button wird erstellt

    public Panel pnlNewLend = new Panel(new FlowLayout(FlowLayout.CENTER));
    public Button btnNewPruef = new Button("Pruefen"); //
Button wird erstellt
    public TextField txtNewKdNr = new TextField("",15);
    public TextField txtNewISBN = new TextField("",15);
    public Label lblNewKdNr = new Label ("Kundennummer: ");
    public Label lblNewISBN = new Label ("Inventarnr. : ");
    public Label lblNewListKd = new Label ("Der Kunde ist: ");
    public Label lblNewListBk = new Label ("Leihgegenstand ist : ");
    public List lstNewKd = new List(4,true);
    public List lstNewBk = new List(4,true);

//-----Aufruf fürs Beenden des Fensters und die Buttons-----
-

    public NewLeihe()
    // enthält Fenstereigenschaften
    {
        super("Leihgegenstand eintragen"); // "Super" ist Befehl für
die "Caption" -> VB!!
        this.setSize(555,500); // Setzt
die Größe des Fensters fest (links,oben)

        this.pnlUnten.add(this.btnNewBack); // Button wird
dem Panel zugeordnet
        this.pnlUnten.add(this.btnNew); // Button wird
dem Panel zugeordnet

        this.pnlNewLend.add(this.lblNewKdNr);
        this.pnlNewLend.add(this.txtNewKdNr);
        this.pnlNewLend.add(this.lblNewISBN);
        this.pnlNewLend.add(this.txtNewISBN);
        this.pnlNewLend.add(this.btnNewPruef);

        this.pnlNewLend.add(this.lblNewListKd);
        this.pnlNewLend.add(this.lstNewBk);

        this.pnlNewLend.add(this.lblNewListBk);
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        this.pnlNewLend.add(this.lstNewKd);

        this.add(this.pnlUnten, BorderLayout.SOUTH);           // Panel wird an das
Fenster übergeben und südlich ausgerichtet
        this.add(this.pnlNewLend, BorderLayout.CENTER);

        this.show();
// DAS wird angezeigt (Visible = true)

Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich Funktion getSize darin
befindet
        this.setLocation((d.width - getSize().width) / 2, // Wird auf die Bildschirmmitte gesetzt
(d.height- getSize().height) / 2);           // Wird auf die Bildschirmmitte gesetzt

        this.addWindowListener(Beenden);                     // Dem Fenster
den Hörer anhängen und mit Beenden verknüpfen
        this.btnNew.addActionListener(NewBtn);
        this.btnNewBack.addActionListener(NewBtn);
        this.btnNewPruef.addActionListener(NewBtn);
    }

    public static void main (String argv[])                   // Ausführende Sequenz starten
    {
        NewLeihe fenster = new NewLeihe();                   // Klasse Fenster wird
als neues Object auf den Bildschirm geladen
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

SuchBtn.java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class SuchBtn implements ActionListener
{
    private Suchen f;
    public SuchBtn(Suchen f)
    {
        this.f=f;
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == f.btnSuchBack)
        {
            this.f.setVisible(false);
            Gut f = new Gut();
        }
//-----BUTTON NEUE AUSLEIHE ENDE-----
-----
        if(e.getSource() == f.btnSuchKd)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen

                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT KdNr, Name,
Vorname FROM Kunden WHERE Kunden.KdNr = "+f.txtSuchKd.getText()+"");
                    while (result.next())
                    {
                        f.lstSuchErgeb.add(""+String.valueOf(result.getInt(1))+
"+String.valueOf(result.getString(2))+ " | "+String.valueOf(result.getString(3))+""");
                    }
                }
                result.close();
                sqlStatement.close();
                dbVerbindung.close();
            }
            catch (SQLException sql)
            {
                sql.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+sql);
                System.exit(0);
            }
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0); // Button
    }
}
Anlegen von Benutzern
}
}
//-----BUTTON DEL BORROW ENDE-----
-----
        if(e.getSource() == f.btnSuchBuch)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                    EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen

                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT InvNr, Titel, Autor
FROM Inventar WHERE InvNr = "+f.txtSuchBuch.getText()+"");
                    while (result.next())
                    {
                        f.lstSuchErgeb.add(""+String.valueOf(result.getInt(1))+
"+String.valueOf(result.getString(2))+ " | "+String.valueOf(result.getString(3))+""");
                    }
                    result.close();
                    sqlStatement.close();
                    dbVerbindung.close();
                }
                catch (SQLException sql)
                {
                    sql.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+sql);
                    System.exit(0);
                }
            }
        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0); // Button
    }
}
Anlegen von Benutzern
}
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
//-----BUTTON DEL BORROW ENDE-----
-----
        if(e.getSource() == f.btnSuchName)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");//      HIER
                    EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen

                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT  KdNr,  Name,
                    Vorname FROM Kunden WHERE Name = '"+f.txtSuchName.getText()+"'");
                    while (result.next())
                    {
                        f.lstSuchErgeb.add(""+String.valueOf(result.getInt(1))+
                        "+String.valueOf(result.getString(2))+ " | "+String.valueOf(result.getString(3))+""");
                    }
                    result.close();
                    sqlStatement.close();
                    dbVerbindung.close();
                }
                catch (SQLException sql)
                {
                    sql.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+sql);
                    System.exit(0);
                }
            }
        }
        catch (Exception Pruef)
        {
            Pruef.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+Pruef);
            System.exit(0);
        }
        //      Button
Anlegen von Benutzern
    }
}

//-----BUTTON DEL BORROW ENDE-----
-----

        if(e.getSource() == f.btnSuchTitel)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");//      HIER
                    EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen

                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    ResultSet result=sqlStatement.executeQuery("SELECT  KdNr,  Name,
                    Vorname FROM Kunden WHERE Name = '"+f.txtSuchTitel.getText()+"'");
                    while (result.next())
                    {
                        f.lstSuchErgeb.add(""+String.valueOf(result.getInt(1))+
                        "+String.valueOf(result.getString(2))+ " | "+String.valueOf(result.getString(3))+""");
                    }
                    result.close();
                    sqlStatement.close();
                    dbVerbindung.close();
                }
                catch (SQLException sql)
                {
                    sql.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+sql);
                    System.exit(0);
                }
            }
        }
        catch (Exception Pruef)
        {
            Pruef.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+Pruef);
            System.exit(0);
        }
        //      Button
Anlegen von Benutzern
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

{
    dbVerbindung =
        DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
    EINMAL DIE DATENBANK ÄNDERN!!!!
    }

    catch(SQLException q)
    {
        q.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+q);
        System.exit(0);
    }

    // catch(SQLException e) schließen

    try
    {
        sqlStatement=dbVerbindung.createStatement();
        ResultSet result=sqlStatement.executeQuery("SELECT InvNr, Titel, Autor
FROM Inventar WHERE Titel = '"+f.txtSuchTitel.getText()+"'");
        while (result.next())
        {
            f.lstSuchErgeb.add(""+String.valueOf(result.getInt(1))+
"+String.valueOf(result.getString(2))+ " | "+String.valueOf(result.getString(3))+""");
        }
        result.close();
        sqlStatement.close();
        dbVerbindung.close();
    }
    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+sql);
        System.exit(0);
    }
}
catch (Exception Pruef)
{
    Pruef.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+Pruef);
    System.exit(0); // Button
Anlegen von Benutzern
}
}

//-----BUTTON DEL BORROW ENDE-----
-----
if(e.getSource() == f.btnSuchClear)
{
    f.lstSuchErgeb.removeAll();
    f.txtSuchKd.setText("");
    f.txtSuchName.setText("");
    f.txtSuchBuch.setText("");
    f.txtSuchTitel.setText("");
}

//-----BUTTON DEL BORROW ENDE-----
-----
}
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Suchen.java

```
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*; // Ereignisse in
den fenstern werden importiert
public class Suchen extends Frame
{
    private Beenden //
Funktion Beenden einleiten (Der Antworter auf den Hörer)
    Beenden = new Beenden(); //
neues Objekt Beenden erstellen und mit Klasse Beenden verknüpfen
    private SuchBtn
    // Für die Klasse in der der Button drin ist
    SuchBtn = new SuchBtn(this);
    public Panel pnlUnten = new Panel(); //
    public Button btnSuchBack = new Button("Zurück zum Hauptfenster");
    public Panel pnlOben = new Panel (new FlowLayout(FlowLayout.CENTER));
    public Label lblSuchKd = new Label("Hier zu suchende KUNDENNUMMER EINTRAGEN:  ");
    public TextField txtSuchKd = new TextField("",10);
    public Button btnSuchKd = new Button("Suchen");
    public Label lblSuchBuch = new Label("Hier zu suchende INVENTARNUMMER EINTRAGEN:  ");
    public Button btnSuchBuch = new Button("Suchen");
    public TextField txtSuchBuch = new TextField("",10);
    public Label lblSuchName = new Label("Hier zu suchenden KUNDENNAMEN EINTRAGEN:  ");
    public Button btnSuchName = new Button("Suchen");
    public TextField txtSuchName = new TextField("",10);
    public Label lblSuchTitel = new Label("Hier zu suchenden BUCHTITEL EINTRAGEN:  ");
    public Button btnSuchTitel = new Button("Suchen");
    public TextField txtSuchTitel = new TextField("",10);
    public List lstSuchErgeb = new List(8, false);
    public Button btnSuchClear = new Button("Eingaben und Liste löschen");
//-----Aufruf fürs Beenden des Fensters und die Buttons-----
-

    public Suchen() //
öffnet die Grafik Fenster
    {
        super("Suchen");
        // "Super" ist Befehl für die "Caption" -> VB!!
        this.setSize(555,500); // Setzt
die Größe des Fensters fest (links,oben)
        this.pnlUnten.add(this.btnSuchBack);
        this.pnlOben.add(this.lblSuchKd);
        this.pnlOben.add(this.txtSuchKd);
        this.pnlOben.add(this.btnSuchKd);
        this.pnlOben.add(this.lblSuchBuch);
        this.pnlOben.add(this.txtSuchBuch);
        this.pnlOben.add(this.btnSuchBuch);
        this.pnlOben.add(this.lblSuchName);
        this.pnlOben.add(this.txtSuchName);
        this.pnlOben.add(this.btnSuchName);
        this.pnlOben.add(this.lblSuchTitel);
        this.pnlOben.add(this.txtSuchTitel);
        this.pnlOben.add(this.btnSuchTitel);
        this.pnlOben.add(this.lstSuchErgeb);
        this.pnlOben.add(this.btnSuchClear);
        this.add(this.pnlUnten, BorderLayout.SOUTH);
        this.add(this.pnlOben, BorderLayout.CENTER);
        this.show();
        // DAS wird angezeigt (Visible = true)
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich
Funktion getSize darin befindet
        this.setLocation((d.width - getSize().width ) / 2, // Wird auf die Bildschirmmitte
gesetzt
        (d.height- getSize().height) / 2); // Wird auf die Bildschirmmitte gesetzt

        this.addWindowListener(Beenden); // Dem Fenster
den Hörer anhängen und mit Beenden verknüpfen
        this.btnSuchKd.addActionListener(SuchBtn);
        this.btnSuchBack.addActionListener(SuchBtn);
        this.btnSuchBuch.addActionListener(SuchBtn);
        this.btnSuchName.addActionListener(SuchBtn);
        this.btnSuchTitel.addActionListener(SuchBtn);
        this.btnSuchClear.addActionListener(SuchBtn);

    }

    public static void main (String argv[]) // Ausführende Sequenz starten
    {
        Suchen fenster = new Suchen(); // Klasse Fenster wird als neues
Object auf den Bildschirm geladen
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

Verwaltung.java

```
import java.awt.*;
    // Abstract Window Toolkit importieren für Fenster, Buttons etc.
import java.awt.event.*; // Ereignisse in
den fenstern werden importiert
public class Verwaltung extends Frame
{
    private Beenden //
    Funktion Beenden einleiten (Der Antworter auf den Hörer)
        Beenden = new Beenden(); //
    neues Objekt Beenden erstellen und mit Klasse Beenden verknüpfen

//-----Hauptfenster anfang-----

    public CheckboxGroup Wahl = new CheckboxGroup();

    public Checkbox chkNeu = new Checkbox ("Neuen Kunden anlegen",false,Wahl); // Erstellen von
nicht gewählter Checkbox
    public Checkbox chkDel = new Checkbox ("Kunden löschen",false,Wahl); // welche
Mitglied in der Gruppe Wahl ist
    public Checkbox chkEdit = new Checkbox ("Kunden bearbeiten",false,Wahl); //
    public Checkbox chkShow = new Checkbox ("Kundenstamm ansehen",false,Wahl); //

    public Panel pnlOben = new Panel(new FlowLayout(FlowLayout.CENTER)); // neues Panel
einrichten
    public Panel pnlUnten = new Panel(new FlowLayout(FlowLayout.CENTER));

//-----Hauptfenster Ende-----
//-----Panel User anlegen Anfang-----

    public Panel pnlUserNew = new Panel(new FlowLayout(FlowLayout.CENTER));

        public Label lblUserNewKdNr = new Label ("Kundennummer");
// erstellt label
        public Label lblUserNewName = new Label ("Name");
// erstellt label
        public Label lblUserNewVorname = new Label ("Vorname");
// erstellt label
        public Label lblUserNewStrasse = new Label ("Strasse und Hausnr."); // erstellt label
        public Label lblUserNewOrt = new Label ("Ort");
// erstellt label
        public Label lblUserNewPLZ = new Label ("PLZ");
// erstellt label

        public TextField txtUserNewKdNr = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserNewName = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserNewVorname = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserNewStrasse = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserNewOrt = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserNewPLZ = new TextField("",10);
//erstellt Textfeld

        public Button btnUserNewSave = new Button ("Benutzer anlegen");
//erstellt Button
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
//-----Panel User anlegen Ende-----
//-----Panel User Löschen Anfang-----

    public Panel pnlUserDel = new Panel(new FlowLayout(FlowLayout.CENTER));           //
Panel erstellen mit LayoutManager

        public TextField txtUserDelKdNr = new TextField("",10);
// erstellt Textfeld

        public Label lblUserDelKdNr = new Label ("Kundennummer:");
// erstellt label
        public Label lblUserDelKdIs = new Label ("Kunde ist:");
// erstellt label

        public Button btnUserDel = new Button ("Aus Kundenstamm entfernen");       //      erstellt
Button
        public Button btnUserDelChk = new Button ("Pruefen");
// erstellt Button

        public List lstUserDelKd = new List(4,false);                               // Neue Liste
mit 40 Feldern die mehrfach gewählt werden können

//-----Panel User löschen Ende-----
//-----Panel User editieren Anfang-----

    public Panel pnlUserEdit = new Panel(new FlowLayout(FlowLayout.CENTER));

        public Label lblUserEditKdNr = new Label ("Kundennummer:");

        public Label lblUserEditName = new Label ("Name");
// erstellt label
        public Label lblUserEditVorname = new Label ("Vorname");
// erstellt label
        public Label lblUserEditStrasse = new Label ("Strasse und Hausnr.");       // erstellt label
        public Label lblUserEditOrt = new Label ("Ort");
// erstellt label
        public Label lblUserEditPLZ = new Label ("PLZ");
// erstellt label

        public TextField txtUserEditKdNr = new TextField ("",10);

        public TextField txtUserEditName = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserEditVorname = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserEditStrasse = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserEditOrt = new TextField("",10);
//erstellt Textfeld
        public TextField txtUserEditPLZ = new TextField("",10);
//erstellt Textfeld

        public Button btnUserEditChk = new Button ("Pruefen");                       //
erstellt Button
        public Button btnUserEdit = new Button ("Änderungen übernehmen");         //
erstellt Button

//-----Panel User editieren Ende-----
//-----Panel Kundenstamm Anfang-----

    public Panel pnlKundenstamm = new Panel(new FlowLayout(FlowLayout.CENTER));
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        public List lstKdStamm = new List(20,true);                // Neue Liste
mit 40 Feldern die mehrfach gewählt werden können
        public Panel pnlKdSt = new Panel(new FlowLayout(FlowLayout.CENTER));

//-----Panel Kundenstamm Ende-----

        public Button
        btnVerwBack = new Button ("Zurück zum Hauptfenster");      // Button für das
Hauptfenster

        private Auswahl
        Auswahl = new Auswahl(this);
                                // Itemlistener integrieren

        private VerwBtn
        VerwBtn = new VerwBtn(this);
                                // ActionListener

//-----Aufruf fürs Beenden des Fensters und die Buttons-----
-

        public Verwaltung()                                       //
öffnet die Grafik Fenster
        {
                super("Benutzerverwaltung");                      // "Super" ist
Befehl für die "Caption" -> VB!!

//-----Panel Benutzer hinzufügen-----

        hinzufügen        this.pnlUserNew.add(this.lblUserNewKdNr);        // Objekt dem Panel
        hinzufügen        this.pnlUserNew.add(this.txtUserNewKdNr);        // Objekt dem Panel
        hinzufügen        this.pnlUserNew.add(this.lblUserNewName);        // Objekt dem Panel
        hinzufügen        this.pnlUserNew.add(this.txtUserNewName);        // Objekt dem Panel
        hinzufügen        this.pnlUserNew.add(this.lblUserNewVorname);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.txtUserNewVorname);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.lblUserNewStrasse);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.txtUserNewStrasse);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.lblUserNewPLZ);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.txtUserNewPLZ);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.lblUserNewOrt);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.txtUserNewOrt);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserNew.add(this.btnUserNewSave);        // Objekt dem Panel

//-----Panel Benutzer löschen-----

        hinzufügen        this.pnlUserDel.add(this.lblUserDelKdNr);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserDel.add(this.txtUserDelKdNr);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserDel.add(this.btnUserDelChk);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserDel.add(this.lblUserDelKdIs);        // Objekt dem Panel hinzufügen
        hinzufügen        this.pnlUserDel.add(this.lstUserDelKd);        // Objekt dem Panel
        hinzufügen        this.pnlUserDel.add(this.btnUserDel);        // Objekt dem Panel

//-----Panel Benutzer editieren-----
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

    this.pnlUserEdit.add(this.lblUserEditKdNr);           // Objekt dem Panel
hinzufügen
    this.pnlUserEdit.add(this.txtUserEditKdNr);         // Objekt dem Panel
hinzufügen
    this.pnlUserEdit.add(this.btnUserEditChk);         // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.lblUserEditName);        // Objekt dem Panel
hinzufügen
    this.pnlUserEdit.add(this.txtUserEditName);        // Objekt dem Panel
hinzufügen
    this.pnlUserEdit.add(this.lblUserEditVorname);     // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.txtUserEditVorname);     // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.lblUserEditStrasse);     // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.txtUserEditStrasse);     // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.lblUserEditPLZ);        // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.txtUserEditPLZ);        // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.lblUserEditOrt);        // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.txtUserEditOrt);        // Objekt dem Panel hinzufügen
    this.pnlUserEdit.add(this.btnUserEdit);           // Objekt dem Panel
hinzufügen

//-----Panel Kundenstamm-----

    this.pnlKundenstamm.add(this.lstKdStamm, BorderLayout.NORTH); //
Objekt dem Panel hinzufügen
    this.pnlKundenstamm.add(this.pnlKdSt);

//-----HAUPTFENSTER-----

    this.pnlOben.add(this.chkNeu);                    // Objekt dem
Panel hinzufügen
    this.pnlOben.add(this.chkDel);                    // Objekt dem
Panel hinzufügen
    this.pnlOben.add(this.chkEdit);                  // Objekt dem
Panel hinzufügen
    this.pnlOben.add(this.chkShow);                  // Objekt dem
Panel hinzufügen
    this.pnlUnten.add(this.btnVerwBack);              // Objekt dem Panel
hinzufügen
    this.add(this.pnlOben, BorderLayout.NORTH);        // Panel dem Fenster
hinzufügen
    this.add(this.pnlUnten, BorderLayout.SOUTH);       // Panel dem Fenster hinzufügen

    this.setSize(570,500);                            // Setzt
die Größe des Fensters fest (links,oben)
    this.show();
    // DAS wird angezeigt (Visible = true)

Dimension d = Toolkit.getDefaultToolkit().getScreenSize(); // Toolkit laden da sich Funktion getSize darin
befindet
    this.setLocation((d.width - getSize().width) / 2, // Wird auf die Bildschirmmitte gesetzt
(d.height- getSize().height) / 2); // Wird auf die Bildschirmmitte gesetzt

    this.addWindowListener(Beenden);                 // Dem Fenster
den Hörer anhängen und mit Beenden verknüpfen

    this.chkNeu.addItemListener(Auswahl);            // Itemlistener mit
Objekt verknüpfen
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

Objekt verknüpfen      this.chkDel.addItemListener(Auswahl);           // Itemlistener mit
Objekt verknüpfen      this.chkEdit.addItemListener(Auswahl);          // Itemlistener mit
Objekt verknüpfen      this.chkShow.addItemListener(Auswahl);         // Itemlistener mit

verknüpfen             this.btnVerwBack.addActionListener(VerwBtn);   // ActionListener mit Objekt
Objekt verknüpfen      this.btnUserDel.addActionListener(VerwBtn);    // ActionListener mit
verknüpfen             this.btnUserDelChk.addActionListener(VerwBtn); // ActionListener mit Objekt
verknüpfen            this.btnUserEdit.addActionListener(VerwBtn); // ActionListener mit Objekt
verknüpfen            this.btnUserEditChk.addActionListener(VerwBtn); // ActionListener mit Objekt
verknüpfen            this.btnUserNewSave.addActionListener(VerwBtn); // ActionListener mit Objekt
}

public static void main (String argv[])           // Ausführende Sequenz starten
{
    Verwaltung fenster = new Verwaltung();         // Klasse Fenster wird
als neues Object auf den Bildschirm geladen
}
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

VerwBtn.java

```
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
public class VerwBtn implements ActionListener
{
    private Verwaltung f;
    public VerwBtn(Verwaltung f)
    {
        this.f=f;
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == f.btnVerwBack)
        {
            this.f.setVisible(false);
            Gut f = new Gut();
        }
//-----BUTTON NEUER KUNDE-----
        if(e.getSource() == f.btnUserNewSave)
        {
            try
            {
                Connection dbVerbindung=null;
                Statement sqlStatement=null;

                try
                {
                    dbVerbindung =
                    DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
EINMAL DIE DATENBANK ÄNDERN!!!!
                }

                catch(SQLException q)
                {
                    q.printStackTrace();
                    System.out.println("DB-Verbindung Fehler: "+q);
                    System.exit(0);
                }

                // catch(SQLException e) schließen

                try
                {
                    sqlStatement=dbVerbindung.createStatement();
                    String Anlegen = "INSERT INTO Kunden (KdNr, Name, Vorname, Strasse,
PLZ, Ort) VALUES ("+f.txtUserNewKdNr.getText()+", "+f.txtUserNewName.getText()+",
"+f.txtUserNewVorname.getText()+", "+f.txtUserNewStrasse.getText()+", "+f.txtUserNewPLZ.getText()+",
"+f.txtUserNewOrt.getText()+")";
                    System.out.println (""+Anlegen);
                    int Anlage=sqlStatement.executeUpdate(Anlegen);
                    if (Anlage == 0)
                    {
                    }
                    else
                    {
                        System.out.println ("Kunde wurde angelegt mit "+Anlage+
"Datensätzen");
                    }
                }
                sqlStatement.close();
            }
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```
        dbVerbindung.close();
    }

    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("SQL-Eingabefehler: "+sql);
        System.exit(0);
    }
}

catch (Exception k)
{
    k.printStackTrace();
    System.out.println("Irgendwas ist Mist!" +k);
    System.exit(0);
}
    f.txtUserNewKdNr.setText("");
    f.txtUserNewName.setText("");
    f.txtUserNewVorname.setText("");
    f.txtUserNewStrasse.setText("");
    f.txtUserNewPLZ.setText("");
    f.txtUserNewOrt.setText("");
}

//-----BUTTON KUNDEN ENTFERNEN - CHECK-----
if(e.getSource() == f.btnUserDelChk)
{
    try
    {
        Connection dbVerbindung=null;
        Statement sqlStatement=null;

        try
        {
            dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
            EINMAL DIE DATENBANK ÄNDERN!!!!
        }

        catch(SQLException q)
        {
            q.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+q);
            System.exit(0);
        }

        // catch(SQLException e) schließen

        try
        {
            sqlStatement=dbVerbindung.createStatement();
            ResultSet result=sqlStatement.executeQuery("SELECT Name, Strasse, Ort
            FROM Kunden WHERE Kunden.KdNr = "+f.txtUserDelKdNr.getText()+"");
            while (result.next())
            {
                f.lstUserDelKd.add(String.valueOf(result.getString(1)));
                f.lstUserDelKd.add(String.valueOf(result.getString(2)));
                f.lstUserDelKd.add(String.valueOf(result.getString(3)));
                //System.out.println("Kundennummer: "+result.getInt(1));
            }
            result.close();
            sqlStatement.close();
        }
    }
}
```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        dbVerbindung.close();
    }
    catch (SQLException sql)
    {
        sql.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+sql);
        System.exit(0);
    }
}
catch (Exception Pruef)
{
    Pruef.printStackTrace();
    System.out.println("DB-Verbindung Fehler: "+Pruef);
    System.exit(0);
} // Button
Anlegen von Benutzern
}
}

//-----BUTTON Kunde Löschen-----

if (e.getSource() == f.btnUserDel)
{
    try
    {
        Connection dbVerbindung=null;
        Statement sqlStatement=null;

        try
        {
            dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
            EINMAL DIE DATENBANK ÄNDERN!!!!
        }

        catch(SQLException q)
        {
            q.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+q);
            System.exit(0);
        }

        // catch(SQLException e) schließen
        try
        {
            sqlStatement=dbVerbindung.createStatement();
            String StrKdNR = f.txtUserDelKdNr.getText();
            String Anlegen = "DELETE FROM Kunden WHERE KdNr =
"+StrKdNR+"";

            System.out.println (""+Anlegen);
            int Anlage=sqlStatement.executeUpdate(Anlegen);
            if (Anlage == 0)
            {
            }
            else
            {
                System.out.println ("Kunde "+StrKdNR+" mit "+Anlage+
"Datensätzen gelöscht!");
            }
            sqlStatement.close();
            dbVerbindung.close();
        }
    }
    catch(SQLException bvb)

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        {
            bvb.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+bvb);
            System.exit(0);
        }
        f.lstUserDelKd.removeAll();
        f.txtUserDelKdNr.setText("");

        } // Verbindung
    catch (Exception BtnDeleten)
    {
        BtnDeleten.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+BtnDeleten);
        System.exit(0);
    }
    // If Anweisung Button von Benutzern!!!!
//-----BUTTON EDIT CHECK-----
    if(e.getSource() == f.btnUserEditChk)
    {
        try
        {
            Connection dbVerbindung=null;
            Statement sqlStatement=null;

            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                EINMAL DIE DATENBANK ÄNDERN!!!!
            }

            catch(SQLException q)
            {
                q.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+q);
                System.exit(0);
            } // catch(SQLException e) schließen

            try
            {
                sqlStatement=dbVerbindung.createStatement();
                ResultSet result=sqlStatement.executeQuery("SELECT Name, Vorname,
                Strasse, PLZ, Ort FROM Kunden WHERE Kunden.KdNr = "+f.txtUserEditKdNr.getText()+"");
                while (result.next())
                {
                    f.txtUserEditName.setText(""+String.valueOf(result.getString(1)));

                    f.txtUserEditVorname.setText(""+String.valueOf(result.getString(2)));
                    f.txtUserEditStrasse.setText(""+String.valueOf(result.getString(3)));
                    f.txtUserEditPLZ.setText(""+String.valueOf(result.getString(4)));
                    f.txtUserEditOrt.setText(""+String.valueOf(result.getString(5)));
                }
            }
            result.close();
            sqlStatement.close();
            dbVerbindung.close();
        }
        catch (SQLException sql)
        {
            sql.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+sql);
            System.exit(0);
        }
    }
}

```

Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek

```

        catch (Exception Pruef)
        {
            Pruef.printStackTrace();
            System.out.println("DB-Verbindung Fehler: "+Pruef);
            System.exit(0); // Button
    }
    Anlegen von Benutzern
    }
}

//-----BUTTON BEARBEITEN-----
    if(e.getSource() == f.btnUserEdit)
    {
        try
        {
            Connection dbVerbindung=null;
            Statement sqlStatement=null;

            try
            {
                dbVerbindung =
                DriverManager.getConnection("jdbc:odbc:bibliothek","","");// HIER
                EINMAL DIE DATENBANK ÄNDERN!!!!
            }
            catch(SQLException q)
            {
                q.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+q);
                System.exit(0);
            }
            // catch(SQLException e) schließen

            try
            {
                sqlStatement=dbVerbindung.createStatement();
                int result=sqlStatement.executeUpdate("UPDATE Kunden SET Name =
                "+f.txtUserEditName.getText()+", Vorname = "+f.txtUserEditVorname.getText()+",Strasse =
                "+f.txtUserEditStrasse.getText()+", PLZ = "+f.txtUserEditPLZ.getText()+", Ort =
                "+f.txtUserEditOrt.getText()+"" WHERE KdNr = "+f.txtUserEditKdNr.getText()+""");
                System.out.println("Kundendaten geändert");
                sqlStatement.close();
                dbVerbindung.close();
            }
            catch (SQLException sql)
            {
                sql.printStackTrace();
                System.out.println("DB-Verbindung Fehler: "+sql);
                System.exit(0);
            }
            f.txtUserEditKdNr.setText("");
            f.txtUserEditName.setText("");
            f.txtUserEditVorname.setText("");
            f.txtUserEditStrasse.setText("");
            f.txtUserEditPLZ.setText("");
            f.txtUserEditOrt.setText("");
        }
    }
    catch (Exception Pruef)
    {
        Pruef.printStackTrace();
        System.out.println("DB-Verbindung Fehler: "+Pruef);
        System.exit(0); // Button Anlegen von Benutzern
    }
}
// Buttons gesamt
}

```

6.2 Anlage: Installationsanleitung

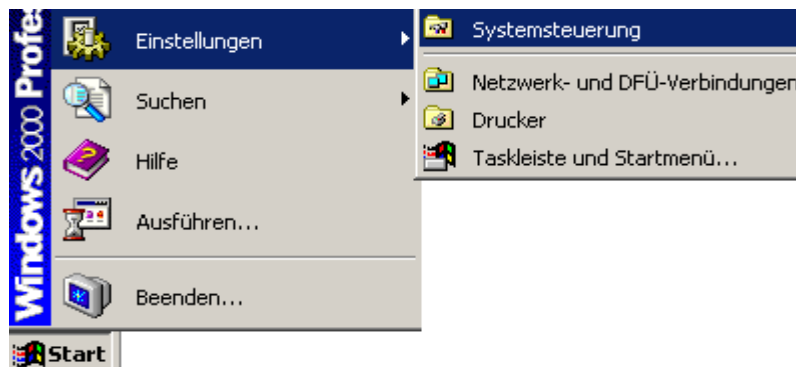
Installationsanleitung der MicroBallZ Bücher und Benutzerverwaltung

1. Kopieren der Dateien

Durch Ausführen der Datei „install.bat“ auf der CD-ROM wird der gesamte Ordner „Buch“ von der CD auf die Festplatte nach C:\Buch kopiert. Damit das Programm fehlerfrei funktioniert sollte der Name des Ordners nicht verändert werden.

2. Einrichten der Datenbank

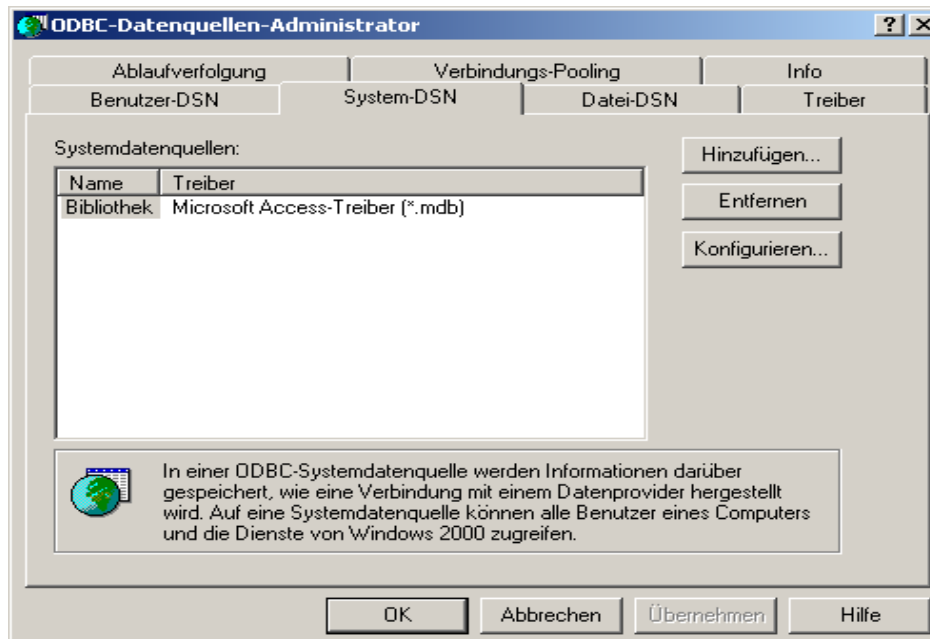
Zum Einrichten der Datenbank klickt man auf Start => Einstellungen => Systemsteuerung



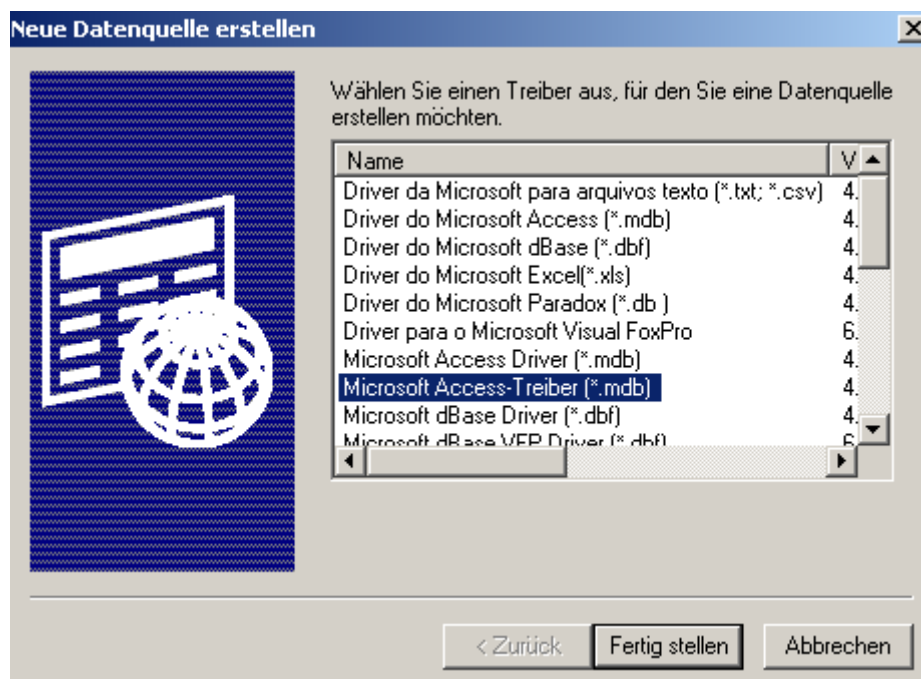
Ist diese geöffnet, doppelklickt man auf  Verwaltung und danach auf  Datenquellen (ODBC) .

In dem sich nun geöffneten Fenster, klickt man auf die Registerkarte „System-DSN“ und in dieser auf den Button „Hinzufügen“.

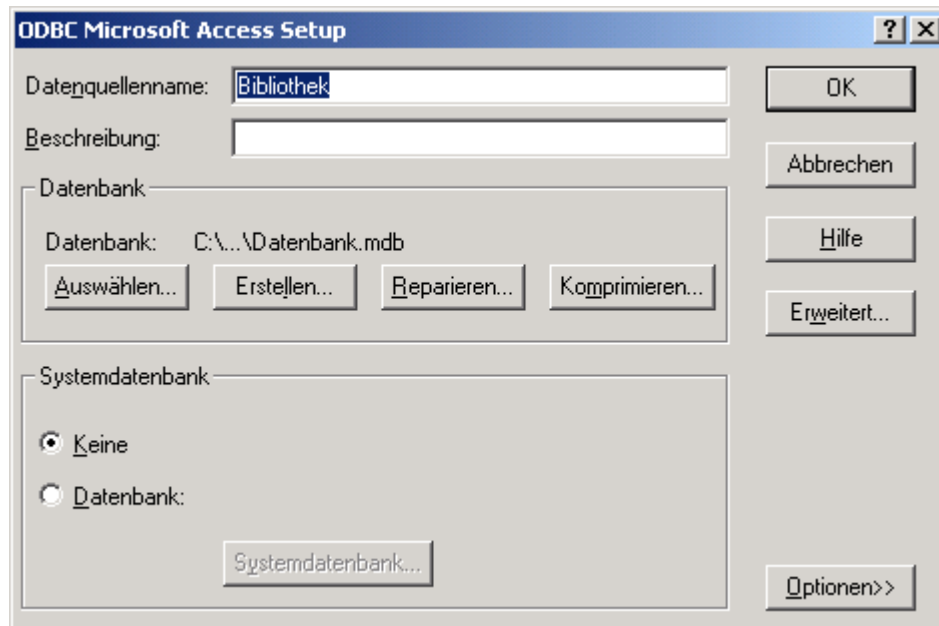
Entwicklung einer Buch- und Benutzerverwaltung der Schulbibliothek



Es öffnet sich ein weiteres Fenster, in dem der Microsoft Access-Treiber gewählt wird und mit „Fertigstellen“ bestätigt wird.



Bei Datenquellennamen „Bibliothek“ eintragen, und unter „Auswählen...“ die Datenbank im vorher gewünschten Verzeichnis markieren und mit OK bestätigen.



3. Starten des Programms

Zum starten des Programms muss man in das Verzeichnis „C:\Buch\“ wechseln und ein Doppelklick auf die Datei „start.bat“ tätigen. Nun erscheint die Benutzeroberfläche der Buch- und Benutzerverwaltung