

**Bearbeitungszeitraum: 3 Wochen****Bearbeitungszeit: 19,5 Stunden (Nacharbeiten Vorlesung ist inbegriffen)**

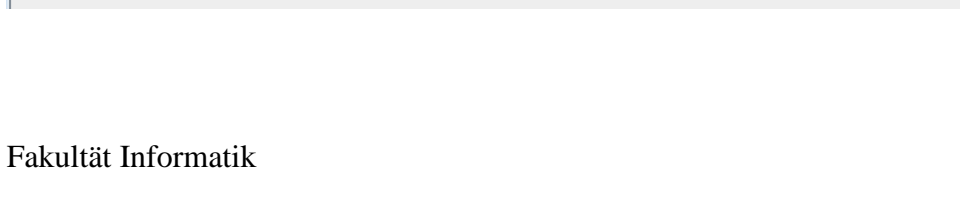
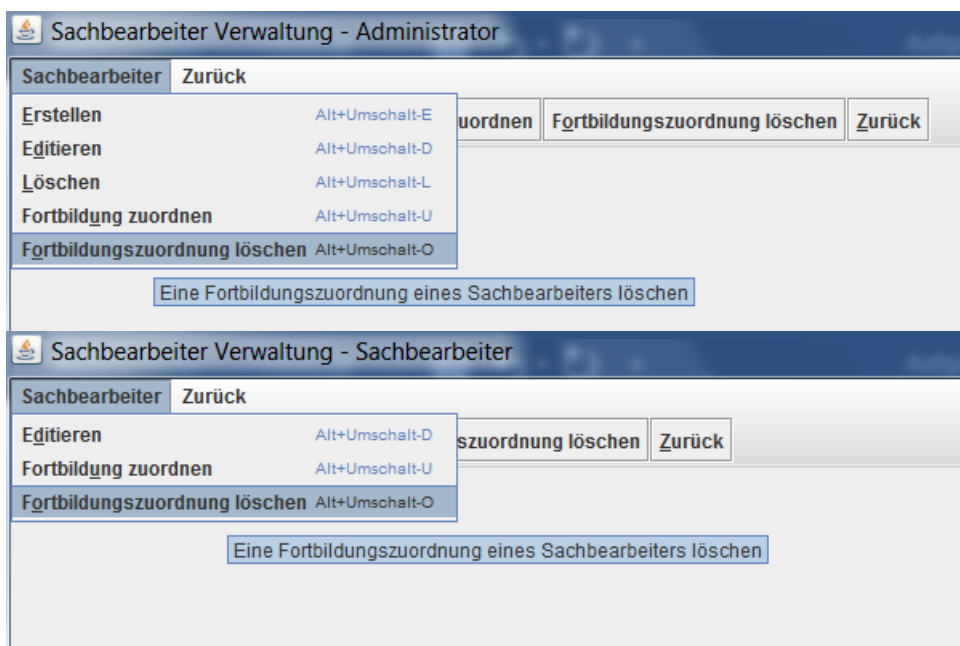
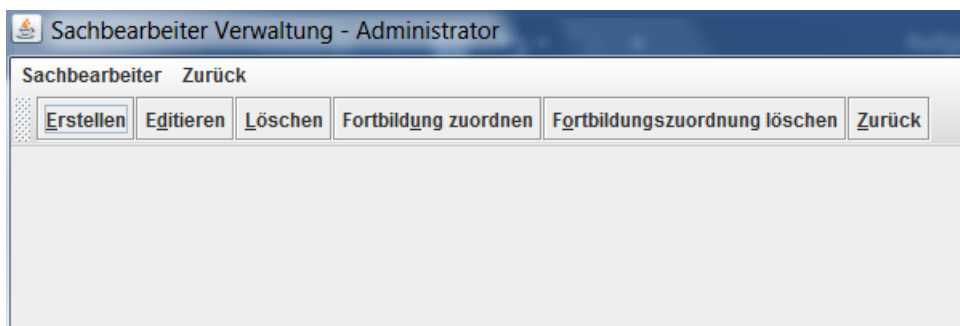
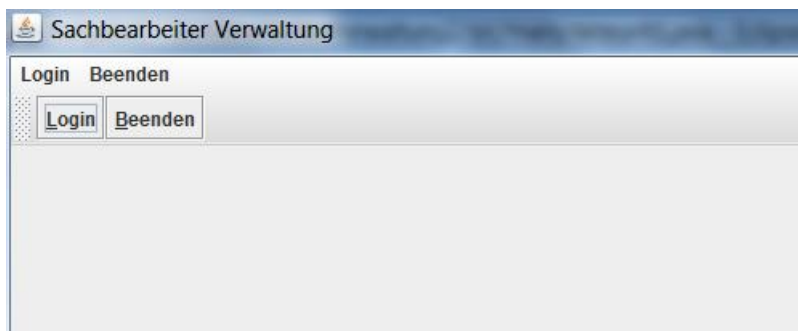
**Zur Abgabe dieser und der folgenden Aufgabenblätter gehört unbedingt auch die Zeit, die Sie für die Lösung dieser Aufgaben benötigt haben. Notieren Sie die benötigte Zeit pro Aufgabe! Die Zeit soll auch die Zeit enthalten, die Sie für das Nachschauen / Nacharbeiten des Vorlesungsstoffs benötigt haben.**

Aufgabe	a)	b)	c)	d)	e)	f)	g)	h)	i)
Zeit / min							-	-	-

- a) Eine Übung zum Swing Zustandmodell (GUI state model):  
Erzeugen Sie ein Fenster mit einer JScrollBar und einem JSlider. Statten Sie beide Komponenten mit demselben BoundedRange Modell aus. Bewegen sich die Schieber jetzt synchron?
- b) Eine Übung zum Swing statefull und lightweight notification  
Programmieren Sie jeweils ein Beispiel für statefull und lightweight notification. Zeigen Sie, dass das jeweilige Event Objekt in einem Fall zustandsbehaftet ist (aktuelle Infos enthält) und im anderen Falle nicht (nur die Quelle enthält).
- c) Ersetzen Sie das Kommandozeilenorientierte Frontend aus Aufgabenblatt durch ein Swing Frontend mit gleicher Funktionalität. Dabei müssen die Vorgaben und Klassen aus Anhang C (Beschreibung des Programms) beachtet werden. Außerdem gibt es folgende Anforderungen:
  1. Beschränken Sie sich auf die Akteur Anwendungsfall Kombinationen
    - Login
    - Admin Sachbearbeiter erstellen
    - Admin Sachbearbeiter bearbeiten
    - Admin Sachbearbeiter löschen
    - Sachbearbeiter Sachbearbeiter editieren
    - Fortbildung zuordnen
    - Fortbildung löschen
  2. Kopieren Sie die alte Anwendung und arbeiten Sie nur auf der Kopie, d.h. unter Eclipse wird ein neues Projekt „Aufgabenblatt 2“ erstellt.
  3. Die Lehrveranstaltungsverwaltung soll ein Hauptfenster haben, das so groß wie der Bildschirm ist. Dieses Fenster soll kleiner gemacht werden können (aufziehen bzw. zuziehen des Fensters).
  4. Führen Sie in die Lehrveranstaltungsverwaltung ein Menü und eine Symbolleiste (JToolBar) ein. Nutzen Sie dabei die Vorteile der Action Klasse: <http://download.oracle.com/javase/tutorial/uiswing/misc/action.html>  
Die Menüs sollen aussehen, wie im Anhang A dargestellt.
  5. Achten Sie darauf, dass bei Fehlersituationen entsprechende Meldungen an den Benutzer gegeben werden (evtl. mit JDialog)
  6. Ihre Benutzeroberfläche soll die im Anhang B dargestellten Layouts haben.
  7. Verwenden Sie in Ihrem Swing GUI mindestens folgende LayoutManager und finden Sie geeignete Stellen in Ihrem Programm:  
BoxLayout  
GridBagLayout

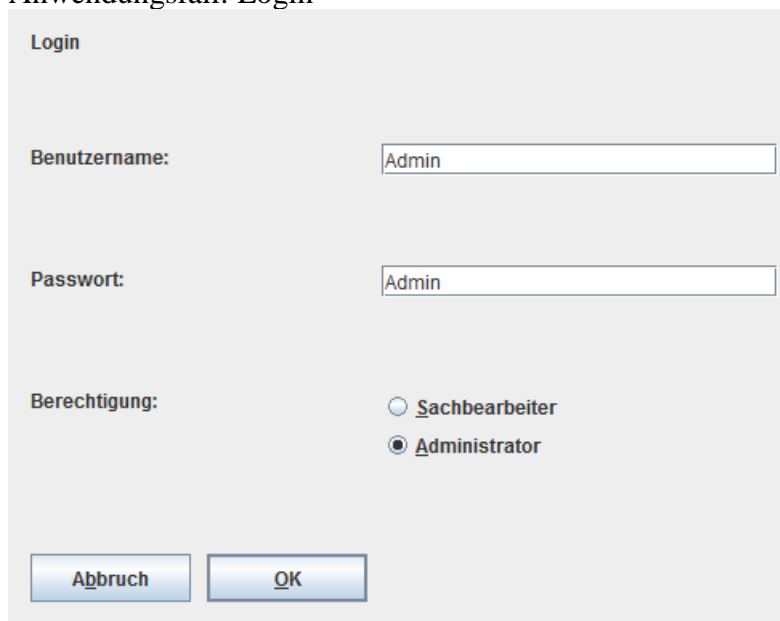
- d) Führen Sie eine Tastatursteuerung Ihrer GUI ein. Achten Sie auf benutzerfreundliche Fokusnavigation. Blockieren Sie den Fokus bei fehlerhafter Eingabe. Setzen Sie den Default Button wo erforderlich.
- e) Wie viele Änderungen müssen Sie nun in der Schnittstellenschicht bzw. Präsentation vornehmen?
- f) Müssen Sie auch Änderungen an tiefer liegenden Schichten vornehmen? Wenn ja, warum?

## Anhang A



## Anhang B

## Anwendungsfall: Login



A login dialog box titled "Login". It contains three input fields: "Benutzername:" with the text "Admin", "Passwort:" with the text "Admin", and "Berechtigung:" with two radio button options: "Sachbearbeiter" (unselected) and "Administrator" (selected). At the bottom, there are two buttons: "Abbruch" and "OK".

Login

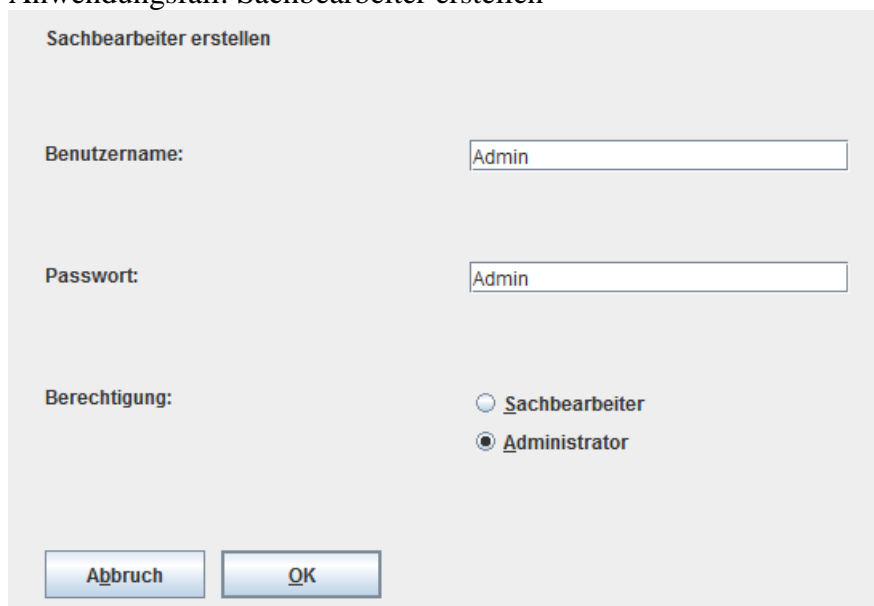
Benutzername: Admin

Passwort: Admin

Berechtigung: ☐ Sachbearbeiter ☒ Administrator

Abbruch OK

## Anwendungsfall: Sachbearbeiter erstellen



A dialog box titled "Sachbearbeiter erstellen". It contains three input fields: "Benutzername:" with the text "Admin", "Passwort:" with the text "Admin", and "Berechtigung:" with two radio button options: "Sachbearbeiter" (unselected) and "Administrator" (selected). At the bottom, there are two buttons: "Abbruch" and "OK".

Sachbearbeiter erstellen

Benutzername: Admin

Passwort: Admin

Berechtigung: ☐ Sachbearbeiter ☒ Administrator

Abbruch OK

## Anwendungsfall: Sachbearbeiter editieren (durch Administrator)

Sachbearbeiter / Admin editieren

Benutzername:

Passwort:

Berechtigung:

☒ Sachbearbeiter

☐ Administrator

Fortbildungen:

Allgemeine Betriebswirtschaft	ist bestanden
Mathematik 1	ist bestanden
Mathematik 2	ist bestanden
Kostenrechnung	ist bestanden

Sachbearbeiter auswählen:

## Anwendungsfall: Fortbildung zuordnen

Fortbildung zuordnen

Ausgewählter Sachbearbeiter:

Ausgewählte Fortbildung:

☒ belegt

☐ bestanden

## Anwendungsfall: Fortbildungszuordnung löschen

Fortbildungszuordnung löschen

Sachbearbeiter  
Carla ▼

Fortbildungen

Allgemeine Betriebswirtschaft	ist bestanden
Mathematik 1	ist bestanden
Mathematik 2	ist bestanden
Kostenrechnung	ist bestanden

Abbruch Löschen

## Anhang C

Beschreibung des Programms:

**main Methode:**

Die main Methode erzeugt die LehrveranstaltungsVerwaltungHS . Dann wird die Methode void oeffnen() von LehrveranstaltungsVerwaltungHS aufgerufen.

**LehrveranstaltungsVerwaltungHS**

Das Hauptfenster wird durch die Klasse LehrveranstaltungsverwaltungHS erzeugt. Dazu hat diese Klasse ein Attribut vom Typ JFrame. In diesen JFrame wird das dargestellte Menü integriert. Das Menü des JFrame soll aussehen, wie im Anhang A dargestellt. Sobald Login ausgewählt wird, wird der Anwendungsfall von LoginAS aufgerufen – die Methode soll login() heißen und einen JPanel mit dem Login GUI (siehe Anhang B) zurückliefern. Dieser JPanel wird dann in den vorhandenen JFrame integriert.

Damit während des Logins kein Menüpunkt aufgerufen werden kann, wird das Menü ausgegraut.

Wenn der login fertig ist, wird von dem ActionListener des Login OK Knopfes die Methode

- public void loginBeenden(int akteur, JPanel loginPanel);

aufgerufen. Die Klasse LehrveranstaltungsVerwaltungHS enthält die Methode loginBeenden.

In dieser Methode wird das übergebene loginPanel aus dem JFrame des Hauptfensters entfernt. Je nach übergebenem akteur wird die Methode

- void start(JFrame hs)

von AdministratorAS oder von SachbearbeiterAS aufgerufen.

**LoginAAS**

Die LoginAAS Klasse holt sich einen JPanel von SachbearbeiterS. Dieses Panel hat das Layout vom Login mit den Knöpfen Abbruch und OK (siehe Anhang B). An diesen beiden Knöpfen werden zwei Klassen AbbruchLauscher und OKLauscher (beides ActionListener) registriert. Diese Klassen sollen innere Klassen von LoginAS sein. Der OKLauscher überprüft die Eingaben und macht mit JOptionPane eventuell Fehlermeldungen. Wenn alles in Ordnung ist wird die Methode

loginBeenden von LehrveranstaltungsVerwaltungHS aufgerufen.

**AdministratorAS**

In der start Methode wird in das übergebene JFrame Hauptfenster ein neues Menü integriert (siehe Anhang A) so dass der Administrator seine zugänglichen Anwendungsfälle aufrufen kann.

Die Klasse AdministratorAS erzeugt Objekte (und speichert sie in Attributen) vom Typ

- AdminSachbearbeiterErstellenAAS
- AdminSachbearbeiterEditierenAAS
- AdminSachbearbeiterLoeschenAAS

Daher kann je nach Menü Auswahl die entsprechende Anwendungsfall Methode aufgerufen werden.

Nehmen wir an, Sachbearbeiter Bearbeiten wurde ausgewählt. Dann wird die Methode

- JPanel adminSachbearbeiterEditieren();

aufgerufen. Der zurückgegebene JPanel wird in das JFrame Hauptfenster integriert.

## **AdminSachbearbeiterEditierenAAS**

Diese Klasse wird in Analogie zu LoginAS entworfen. Die Methode `adminSachbearbeiterEditieren` baut ein `JPanel` zusammen und liefert es zurück. Dazu bedient sie sich eines Objekts vom Typ `SachbearbeiterEditierenAS`, welches eine Methode

- `JPanel sachbearbeiterEditieren (boolean editierbar)`

hat. Die Methode gibt ein `JPanel` mit Textfeldern und `RadioButton` für die Eingabe von Benutzername, Passwort und Berechtigung zurück. Dieses `JPanel` wird in ein neues `JPanel` eingefügt. Mit hinein in das neue `JPanel` kommen noch der Abbruch und der OK Knopf. Die zu den Knöpfen gehörigen Lauscher sollen innere Klassen in der Klasse `AdminSachbearbeiterEditierenAAS` sein. Dabei ist wichtig, dass der OKLauscher bei `SachbearbeiterEditierenAS` Passwort, Benutzername, Berechtigung und den gerade gewählten Sachbearbeiter ermitteln kann.

Sobald der OKLauscher die neuen Werte mit Hilfe der K Klasse `SachbearbeiterEditierenK` in die Entitätsklasse `Sachbearbeiter` eingetragen hat, zeigt er der `AdministratorAS` Klasse sein Ende durch Aufrufen der Methode

- `void adminSachbearbeiterEditierenAbschliessen(JPanel adminEditPanel);`

an. In dieser Methode wird der `adminEditPanel` wieder aus dem Hauptfenster `JFrame` entfernt. Die Methode muss von `AdministratorAS` implementiert werden.

## **SachbearbeiterEditierenAAS**

In der Methode `JPanel sachbearbeiterEditieren (boolean editierbar)` wird ein `JPanel` mit Textfeldern und `RadioButtons` von `SachbearbeiterS` geholt. Dieses `JPanel` wird zusammen mit einer `JComboBox` für die `SachbearbeiterAuswahl` in ein neues `JPanel` eingefügt. Dieses `JPanel` wird zurückgegeben. Diese Klasse kann von folgenden Klassen benutzt werden:

- `AdminSachbearbeiterBearbeitenAAS`
- `SachbearbeiterSachbearbeiterBearbeitenAAS`

An der `JComboBox` muss sich ein `SelectionListener` anmelden, der die Textfeldern und `RadioButtons` von `SachbearbeiterS` entsprechend der Auswahl von `JComboBox` aktualisiert. Dieser `SelectionListener` kann als innere Klasse von `SachbearbeiterS` ausgeführt werden.

## **SachbearbeiterAuswaehlenAAS**

Diese `JComboBox` hat ein Model. Dort sind die Objekte versammelt, die zur Auswahl stehen. Deren `toString` Methoden werden genutzt, um die Objekte in einer Liste anzuzeigen. Es ist unklug, die `Sachbearbeiter` Objekte selbst in das Model einzutragen. Denn dann sieht die Darstellung (Präsentation) der `Sachbearbeiter` in allen Listen der GUI überall gleich aus, weil überall dieselbe `toString` Methode benutzt wird. Eine Möglichkeit die Präsentation an den Anwendungsfall anzupassen besteht darin, in der zugehörigen K Klasse eine Wrapper Klasse als innere Klasse vorzusehen. Die Wrapper Klasse hat ein Attribut auf ein `Sachbearbeiter` Objekt und überschreibt die `toString` Methode. In der `toString` Methode wird aus den Werten des `Sachbearbeiter` Objekts eine beliebige Präsentation (String) zusammengestellt, z.B. aus Nachnamen und Vornamen oder nur aus dem Nachnamen etc. Diese Wrapped Objekte werden der `JComboBox` übergeben.

Wird vom Benutzer ein bestimmter `Sachbearbeiter` selektiert (`getSelectedItem`) so muss man folgendermaßen vorgehen, um z.B. den Benutzernamen dieses `Sachbearbeiters` zu erhalten: man übergibt das Objekt einer Methode der zugehörigen K Klasse (z.B. `String gibBenutzername(Object objekt)`). Diese holt aus der Wrapped Klasse den `Sachbearbeiter` und aus dem `Sachbearbeiter` den Namen. Der Name wird als String der Schnittstellenschicht zurückgegeben. Durch dieses Vorgehen kann man es vermeiden, dass die



Schnittstellenschicht Details der Entitätsschicht (z.B. den Klassennamen „Sachbearbeiter“) kennt.

### ***SachbearbeiterS***

Die Methode

- JPanel konfiguriereSachbearbeiterPanel(String ueberschrift, String defaultBenutzername, String defaultPasswort, boolean editierbar)

liefert ein Panel mit Textfeldern für Benutzername und Passwort sowie Radiobuttons für die Berechtigung. Für dieses Panel soll ein GridBagLayout zum Einsatz kommen.

Diese Klasse wird von folgenden Klassen benutzt:

- SachbearbeiterBearbeitenAS
- LoginAS

### ***AdminSachbearbeiterErstellenAAS***

Wird analog wie AdminSachbearbeiterEditierenAAS implementiert.

### ***AdminSachbearbeiterLoeschenAAS***

Wird analog wie AdminSachbearbeiterEditierenAAS implementiert.

### ***FortbildungAuswaehlenAAS***

Diese Klasse kann genauso aufgebaut werden, wie SachbearbeiterAuswaehlenAAS. Sie wird von folgenden Klassen benötigt: FortbildungZuordnenAAS, FortbildungsZuordnungLoeschenAAS. In einer JComboBox stehen alle fest im System abgespeicherten Fortbildungen zur Auswahl.

### ***FortbildungZuordnenAAS***

Besteht aus SachbearbeiterAuswaehlenAAS und FortbildungAuswaehlenAAS und zwei RadioButton: belegt und bestanden. Dazu noch die üblichen JButton Abbruch und Ok.

### ***FortbildungsZuordnungLoeschenAAS***

Besteht aus SachbearbeiterAuswaehlenAAS und FortbildungAuswaehlenAAS. Dazu noch die JButtons Abbruch und Löschen.