



Hochschule für  
Technik und Wirtschaft  
Dresden  
University of Applied Sciences

# Entwicklerdokumentation

## Case-Gruppe 04

Modul: Software Engineering II

Studiengang Informatik

Sommersemester 2014

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Entwurf</b>	<b>1</b>
2.1	Grobentwurf . . . . .	1
2.1.1	Architektur . . . . .	1
2.1.2	Paketdiagramm . . . . .	1
<b>3</b>	<b>Implementierungsentwurf</b>	<b>1</b>
3.1	Paket Typen . . . . .	1
3.1.1	Die Klasse Beleg . . . . .	1
3.1.2	Die Klasse Thema . . . . .	1
3.1.3	Die Klasse Gruppe . . . . .	1
3.1.4	Die Klasse Rolle . . . . .	2
3.1.5	Die Klasse Student . . . . .	2
3.2	Paket DozentBelegverwaltungUI . . . . .	2
3.2.1	Die Klasse MainForm . . . . .	2
3.2.2	Die Klasse BelegBearbeiten . . . . .	5
3.2.3	Die Klasse GruppeBearbeiten . . . . .	5
3.2.4	Die Klasse RolleVerwalten . . . . .	5
3.2.5	Die Klasse ThemenVerwalten . . . . .	5
3.2.6	Die Klasse PdfArchivierung . . . . .	5
3.2.7	Die Klasse KontaktForm . . . . .	5
3.2.8	Die Klasse Eingabe . . . . .	5
3.3	Paket StudentBelegverwaltungUI . . . . .	5
3.3.1	Die Klasse LoginForm . . . . .	5
3.3.2	Die Klasse MainForm . . . . .	5
3.3.3	Die Klasse FormErstanmeldung . . . . .	5
3.3.4	Die Klasse FormMitgliederNeuEingeben . . . . .	5
3.4	Paket DBService . . . . .	5
3.4.1	Die Klasse Database . . . . .	5

# 1 Einleitung

Die vorliegende Dokumentation dient zukünftigen Entwicklern dazu sich in das bestehende System einarbeiten zu können und die Arbeit fortsetzen zu können.

## 2 Entwurf

### 2.1 Grobentwurf

#### 2.1.1 Architektur

#### 2.1.2 Paketdiagramm

## 3 Implementierungsentwurf

Unser Programm ist in insgesamt 4 Pakete gegliedert. Die beiden Pakete DozentBelegverwaltungUI und StudentBelegverwaltungUI enthalten sämtliche Klassen für die Benutzeroberflächen (Login, Bearbeitung, ...) des Dozenten- und des Studentenprogrammes. Sie verwenden die Klassen aus dem Typen-paket und greifen gemeinsam auf die Datenbank des Paketes DB.Services zu.

Im folgenden werden für jedes Paket die Klassen aus dem Grobentwurf mit ihren Methoden definiert und ihre Aufgaben beschrieben.

### 3.1 Paket Typen

#### 3.1.1 Die Klasse Beleg

Die Klasse Beleg enthält einen Konstruktor zur Initialisierung der Attribute. Die für einen Beleg relevanten Eigenschaften sind eine Kennung, ein Passwort zur Erstanmeldung, das automatisch generierte Semester, einen Dozenten, ein Start- und Enddatum, sowie die minimale und maximale Gruppenanzahl.

Außerdem beinhaltet die Klasse noch eine Funktion **AddGruppe** mit welcher Studenten-Gruppen vom Typ Gruppe zu einem Beleg hinzugefügt werden können.

#### 3.1.2 Die Klasse Thema

Ein Thema besitzt nur eine Nummer und einen Namen. Diese beiden Attribute werden durch den Konstruktor gesetzt.

#### 3.1.3 Die Klasse Gruppe

Die Attribute der Klasse Gruppe sind eine Kennung, eine Liste mit den Studenten die in der Gruppe arbeiten, ein Passwort, ein von der Gruppe gewähltes Thema (Nummer),

sowie eine Belegkennung. Sie werden vom Konstruktor initialisiert, wobei einer der beiden Konstruktoren die Belegkennung setzt und der andere nicht.

Desweiteren enthält diese Klasse eine Funktion **AddStudent** mit welcher ein Student vom Typ Student hinzugefügt werden kann.

#### 3.1.4 Die Klasse Rolle

Die Klasse Rolle enthält nur den Rollennamen als Attribut, welcher vom Konstruktor initialisiert wird.

#### 3.1.5 Die Klasse Student

Die wichtigsten Informationen über einen Studenten sind der Nach- und Vorname, die s-Nummer, sowie die Email-Adresse und die Rolle. Diese Attribute sind in der Klasse Student enthalten und werden vom Konstruktor gesetzt.

### 3.2 Paket DozentBelegverwaltungUI

#### 3.2.1 Die Klasse MainForm

Die MainForm-Klasse im Programm Dozent ist dafür da, um einen schnellen Überblick über alle laufenden Belege, dessen Gruppen und der darin enthaltenden Studenten zu gewährleisten. Dabei wird sehr eng mit der Datenbank gearbeitet, um immer aktuelle und richtige Daten anzuzeigen.

- **public MainForm()**

Dieser Konstruktor initialisiert die Klasse. Es wird der Titel des Fenster gesetzt und die Startposition auf die Mitte des Bildschirms verlagert, so dass der Nutzer einen direkten Überblick über die Funktionen bekommt. Außerdem wird die Funktion UpdateBelege() aufgerufen, wodurch die aktuell laufenden Belege aus der Datenbank geladen und angezeigt werden. Desweiteren werden die Eventhandler für die Doppelklick-Funktionen der Listboxen für die Belege und Gruppen hinzugefügt, damit man diese intuitiv bearbeiten kann.

- **void mitgliederDataGridView\_UserDeletingRow (object sender, DataGridViewRowCancelEventArgs e)**

Diese Funktion wird aufgerufen, wenn der Nutzer eine Zeile (einen Studenten) aus einer Gruppe löschen möchte. Dies wird gemacht, um bei dem Nutzer nachzufragen, ob er sich über die Konsequenzen im Klaren ist und außerdem, um diese Änderung direkt in der Datenbank einzutragen. Ein Student mit der S-Nummer "na" als Platzhalter kann nicht gelöscht werden, damit die Mindest- und Maximalanzahl der Studenten in der Gruppe intakt bleibt.

- **private void UpdateBelege(object sender)**

Die Funktion UpdateBelege zieht die aktuell laufenden Belege aus der Datenbank und zeigt diese direkt in der belegListBox an. Somit bekommt der Nutzer eine direkte Übersicht auf die Daten in der Datenbank.

- **private void belegListBox\_SelectedIndexChanged  
(object sender, EventArgs e)**

Dies ist ein Eventhandler, der aufgerufen wird, sobald ein neuer Beleg ausgewählt wurde. Er wird dafür genutzt, um die zu dem Beleg gehörigen Gruppen anzuzeigen. Es wird zuerst der ausgewählte Beleg gesucht. Da die Funktion allgemein aufgerufen werden soll, um die ListBox der Gruppen zu aktualisieren, wird zuerst geschaut, ob überhaupt ein Beleg existiert. Wenn keiner existiert, wird der Inhalt der gruppenListBox geleert, da es dann keine zugehörigen Gruppen gibt. Wenn ein Beleg existiert, der ausgewählt ist, dann wird eine Anfrage an die Datenbank geschickt, die die zugehörigen Gruppen zurückgibt. Diese werden direkt in die gruppenListBox eingetragen.

- **private void UpdateRollen(Beleg beleg)**

Diese Funktion zieht die verfügbaren Rollen zu dem ausgewählten Beleg aus der Datenbank. Sie ist dafür notwendig, damit man den Studenten einer Gruppe eine neue Rolle zuweisen kann. Der Beleg als Parameter wird benötigt, um die Belegkennung in der Datenbankabfrage zu verwenden.

- **private void belegListBox\_DoubleClicked(object sender, EventArgs e)**

Dies ist der oben angesprochene Eventhandler für das Doppel-Klick-Event in der belegListBox. Es wird eine neue Instanz der Klasse BelegBearbeiten erstellt, wobei direkt der Delegate-Handler festgelegt wird. Dieser ist notwendig, um Änderungen nach beenden dieses Dialoges in den ListBoxen anzuzeigen. Der zweite Parameter in dem Konstruktor ist ein Boolean-Wert, der angibt, ob es sich um einen neu angelegten Beleg handelt. Da hier das Doppel-Klick-Event abgefangen wird, handelt es sich nicht um einen neuen Beleg, sondern um einen existierenden, der bearbeitet werden soll. Danach wird der Dialog mit der Methode show() dem Nutzer angezeigt.

- **private void gruppenListBox\_DoubleClicked(object sender, EventArgs e)**

Dieser Eventhandler behandelt das Doppel-Klick-Event für die gruppenListBox. Wie bei den Belegen auch gibt es hier einen Boolean-Wert, der der Instanz sagt, ob es sich um eine neue oder eine existierende Gruppe handelt. Auch hier handelt es sich um eine existierende Gruppe, weshalb der Wert false übergeben wird.

- **private void gruppenListBox\_SelectedIndexChanged  
(object sender, EventArgs e)**

Dieser Eventhandler wird aufgerufen, sobald der Nutzer eine neue Gruppe in der

gruppenListBox auswählt. An dieser Stelle müssen jetzt die Studenten dieser Gruppe angezeigt werden. Dafür wird zuerst die ausgewählte Gruppen rausgesucht und anhand dieser und der Gruppenkennung (Case-Kennung) eine Datenbankabfrage nach den Studenten gestartet. Mit dem Ergebnis der Abfrage wird das mitglieder-DataGridView gefüllt, sodass der Nutzer alle Studenten dieser Gruppe einsehen kann. Bei dem Füllen werden außerdem Platzhalter für Studenten angezeigt, die repräsentieren, wieviele Studenten noch bis zur festgelegten Minimalanzahl bzw bis zur festgelegten Maximalanzahl fehlen. Noch auszufüllende Studenten sind dabei Gelb markiert.

- **protected override void OnFormClosing(FormClosingEventArgs e)**

Wie das Stichwort override verrät, wird an dieser Stelle die Closing-Methode der Windows-Form überschrieben. Dies ist notwendig, weil es sein kann, dass diese Form nur von einer anderen präsentiert wird. Wenn das Fenster geschlossen wird, soll auf jeden Fall auch der Prozess beendet werden.

- **private void belegAnlegenButton\_Click(object sender, EventArgs e)**

Mit dieser Funktion kann man einen neuen Beleg anlegen. Sie wird geworfen, wenn der entsprechende Button im Interface angeklickt wird. Dafür wird ebenfalls die Klasse BelegBearbeiten verwendet, allerdings wird diesmal im Konstruktor mit dem Boolean "true" mitgeteilt, dass es sich um eine neue Klasse handelt. Auch hier wird ein Delegate-Handler festgelegt, um die Listboxen nach dem Anlegen zu aktualisieren.

- **private void gruppeAnlegenButton\_Click(object sender, EventArgs e)**

Dies ist der äquivalente Button zu der BelegAnlegen-Funktion. Es wird eine neue Gruppe angelegt, indem eine Instanz der Klasse GruppeBearbeiten angelegt wird. Auch hier wird ein Delegate gesetzt, um nach dem Anlegen die gruppenListBox zu aktualisieren.

**3.2.2 Die Klasse BelegBearbeiten**

**3.2.3 Die Klasse GruppeBearbeiten**

**3.2.4 Die Klasse RolleVerwalten**

**3.2.5 Die Klasse ThemenVerwalten**

**3.2.6 Die Klasse PdfArchivierung**

**3.2.7 Die Klasse KontaktForm**

**3.2.8 Die Klasse Eingabe**

### **3.3 Paket StudentBelegverwaltungUI**

**3.3.1 Die Klasse LoginForm**

**3.3.2 Die Klasse MainForm**

**3.3.3 Die Klasse FormErstanmeldung**

**3.3.4 Die Klasse FormMitgliederNeuEingeben**

### **3.4 Paket DBService**

**3.4.1 Die Klasse Database**