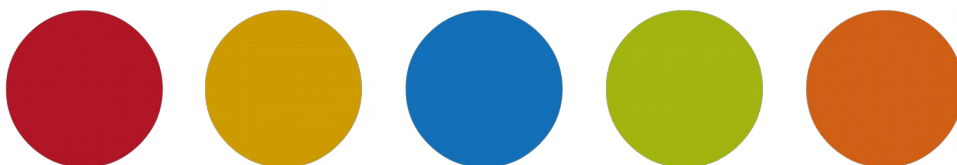


Model, View, Controller

Leittext

Modulname:	Lehrerdaten
------------	-------------

Stand: 28. Nov 2016



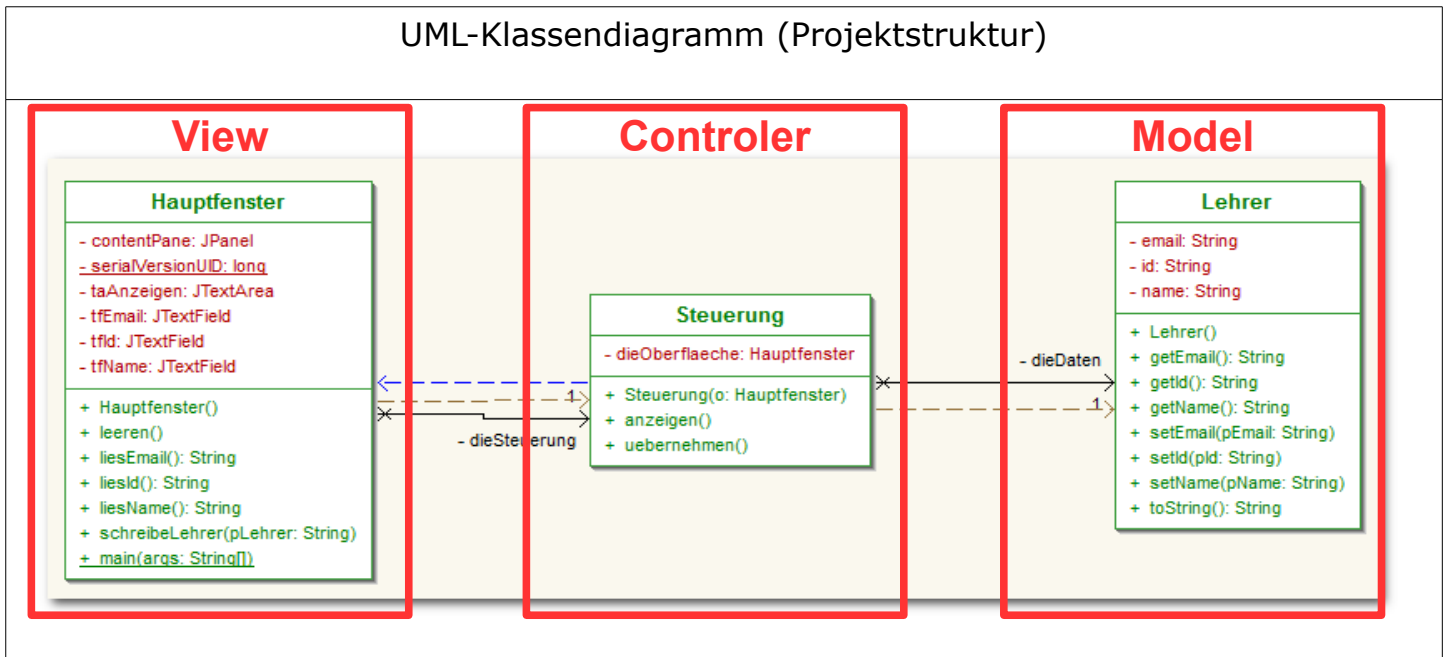
© Christine Janischek

Inhaltsverzeichnis

1 Projekte mit einer MVC-Architektur.....	3
1.1 Lehrerdaten-Projekt.....	3
1.2 Lehrerdaten-Projektstruktur.....	3
1.3 Fensterklasse.....	4
1.4 Eingabe: Lesen.....	5
1.5 Verarbeitung: Übernehmen und Anzeigen.....	5
1.6 Ausgabe: Schreiben.....	5
1.7 Kapselung: Ereignissteuerung.....	5

1 Projekte mit einer MVC-Architektur

1.1 Lehrerdaten-Projekt



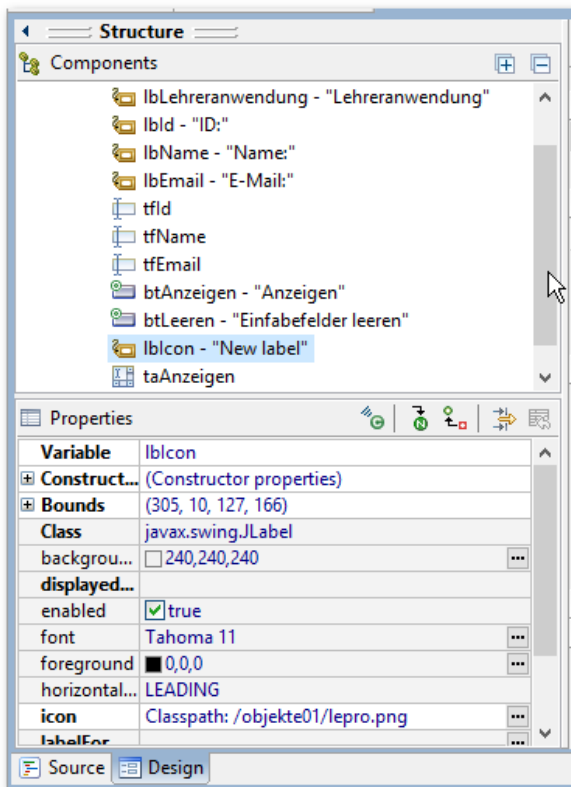
1.2 Lehrerdaten-Projektstruktur

Bild	Hinweise	
	Hauptfenster.java	GUI, Fensterklasse
	Lehrer.java	Modelklasse, Fachklasse
	Starterklasse	Testklasse bzw. Main-Methode auslagern
	Steuerung.java	Controller-Klasse steuert den Zugriff und die Verarbeitung von Objekten der Hauptfenster- und Fachklasse.
	Lepro.png	Grafik für die GUI

1.3 Fensterklasse



Beginnen Sie mit der Benutzeroberflächenklasse `Hauptfenster.java`. Erzeugen Sie dazu folgendes Ergebnis mit Hilfe des WindowsBuilers (SwingDesigner)



Swing-Designer: Properties

```

118     JLabel lbIcon = new JLabel("New label");
119     lbIcon.setIcon(new ImageIcon(
120         Hauptfenster.class
121             .getResource("/objekte01/lepro.png")));
122
123     lbIcon.setBounds(305, 10, 127, 166);
124     contentPane.add(lbIcon);
125

```

Quellcode

Um ein Bild einzufügen müssen Sie die Grafik in das „package-Verzeichnis“ kopieren. Legen Sie als Komponente ein `JLabel` an und fügen Sie in den Properties → icons den Classpath zur Bild-Datei ein.



Hinweis: Falls Sie ein bestehendes package kopiert und umbenannt haben müssen Sie den Classpath nachträglich anpassen.

1.4 Eingabe: Lesen

Implementierung in der *View-Klasse*. Lesen, umwandeln und zurückgeben der Eingabewerte.

View ↔ Controller.

1.5 Verarbeitung: Übernehmen und Anzeigen

Implementierung in der *Controller-Klasse*. Setzen der Eingabewerte im Objekt der Modell-Klasse.

View ↔ Controller → Model

1.6 Ausgabe: Schreiben

Implementierung `toString()`-Methode in der *Model-Klasse*. Implementierung der Schreibmethode in der View. Implementierung der Anzeige-Methode in der Steuerung. Indirekter Zugriff.

1.7 Kapselung: Ereignissteuerung

Anwendung des Prinzips Eingabe-Verarbeitung-Ausgabe (EVA).