



1

Das Produkt-Pflege-System

Beziehungen und Aktionen

Autor: Christine Janischek
Thema: Beziehungen, Ereignisse und Aktionen

Kaufmännische
Schule
Wangen
Wege zeigen, öffnen, gehen

[Beziehungen, Ereignisse und Aktionen: Benutzer hinzufügen und anmelden]

Schritt 1: Java-Editor öffnen



Java-Editor

Öffnen Sie ihre Entwicklungsumgebung, den Java-Editor: *Start >> Programme >> Java-Editor*



Schritt 2: Datei öffnen



Öffnen Sie die Dateien *Hauptfenster.java* und *Person.java*:

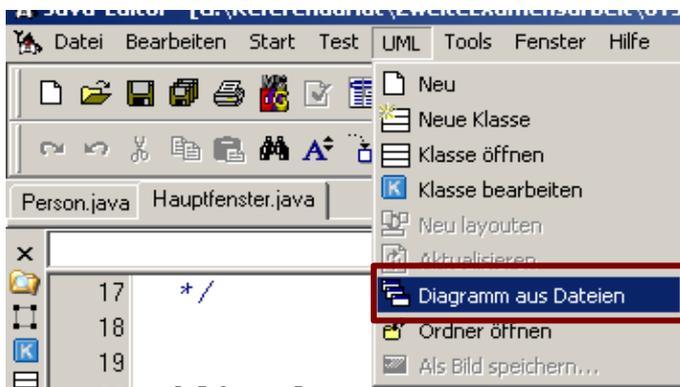


Speichern Sie beide Dateien nacheinander in einem neuen Ordner *Beziehungen*: Datei >> Speichern unter

PHASE 1: ASSOZIATION (*Objekte verschiedener Klassen verbinden*)



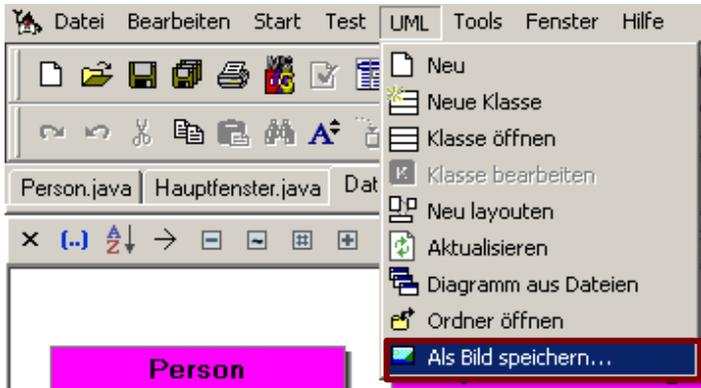
Schritt 3: Betrachtung des UML-Klassendiagramms



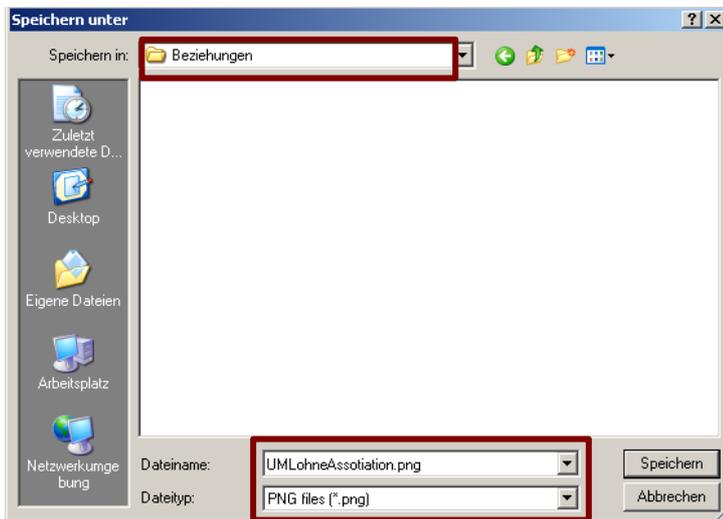
Nutzen Sie die Option in der Menü-Leiste: *UML >> Diagramm aus Dateien* um das Klassendiagramm der aktuellen Anwendung zu erstellen.



Schritt 4: Speichern des UML-Klassendiagramms *ohne Assoziation*



Speichern Sie das UML-Klassendiagramm indem Sie in der Menü-Leiste die Option: *UML >> Als Bild speichern* wählen.



Nennen Sie das Bild: *UMLohneAssoziation* und speichern Sie es in Ihrem Ordner *Beziehungen*

Schritt 5: Einfügen der Assoziation in die Hauptfenster-Klasse

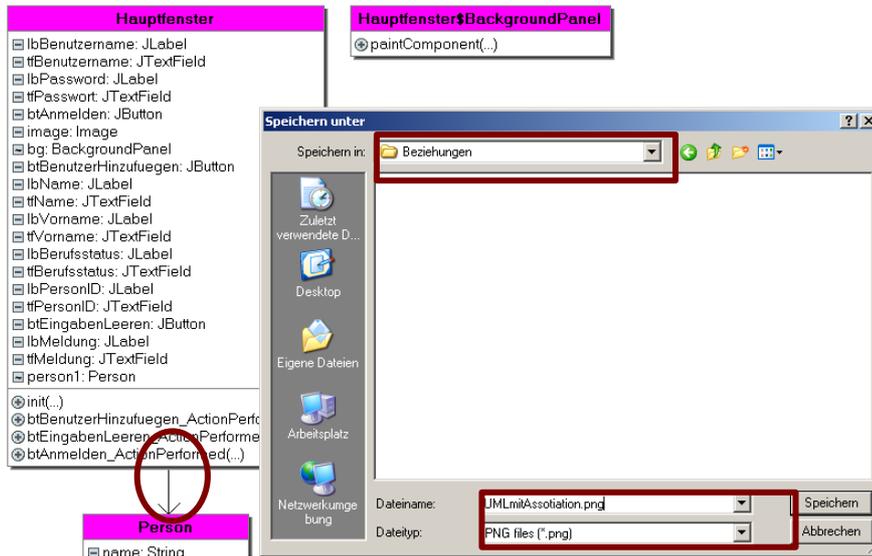
Fügen Sie vor dem Kommentar `//Ende Attribute` in der Klasse `Hauptfenster.java` den folgenden Quellcode für das Objekt `person1` hinzu:

```
Person person1 = new Person();  
// Ende Attribute
```

Nutzen Sie die Tastenkombination `Strg+s` um Ihre Veränderungen zu speichern.



Schritt 6: Betrachtung und Speichern des UML-Klassendiagramms *mit Assoziation*



Wiederholen Sie *Schritt 3* und *4*, speichern Sie das Bild unter dem Namen:
UMLmitAssoziation

Herzlichen Glückwunsch!!

Jetzt können die Objekte der *Hauptfenster*-Klasse mit denen der *Fachklasse Person* kommunizieren.

BEGINNEN SIE MIT PHASE 2



PHASE 2: EREIGNIS-METHODEN (*Impulse lösen Verhalten aus*)



Schritt 7: Quellcode der Ereignis-Methode Benutzer hinzufügen implementieren

Erinnern Sie sich an die Test-Methode der `StarterKlasse.java`? Wir haben in dieser Methode anfangs alle Attribute des Objektes `person1` gesetzt.

Da wir nun keine `StarterKlasse.java` mehr haben wollen wir die Klasse `Hauptfenster.java` (die Benutzeroberfläche) dazu nutzen alle Attribute des Objektes `person1` zu setzen. Der Unterschied ist also, dass die Werte aus den Textfeldern (`tfPersonID`, `tfName`, `tfVorname`, `tfBerufsstatus`, `tfBenutzername`, `tfPasswort`) stammen.

Fügen Sie dazu anstelle des Kommentars: `// TODO hier Quelltext einfügen` in die Ereignis-Methode:

```
public void btBenutzerHinzufuegen ActionPerformed(ActionEvent evt) {  
    // TODO hier Quelltext einfügen  
}
```

folgenden Quellcode ein:

```
person1.setPersonID(tfPersonID.getText());  
person1.setName(tfName.getText());  
person1.setVorname(tfVorname.getText());  
person1.setBerufsstatus(tfBerufsstatus.getText());  
person1.setBenutzername(tfBenutzername.getText());  
person1.setPasswort(tfPasswort.getText());  
tfMeldung.setText("Sie wurden als Benutzer erfolgreich hinzugefügt!");
```

Erklärung:

Programmieren und lesen Sie die Methoden-Aufrufe von *innen nach außen*, das ist einfacher. Im jetzigen Fall sind das je Attribut (Zeile) 2 Schritte:

1. Hole den Wert aus dem Textfeld `>> <Textfeldname>.getText();`
2. Setze das entsprechende Attribut von `person1` auf den geholten Wert aus dem Textfeld `>> person1.set<Attributname>(...);`



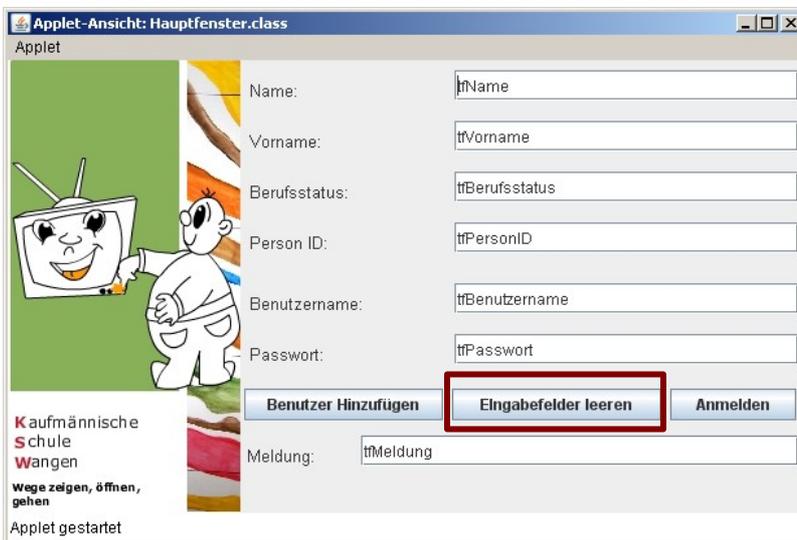
Schritt 8: Kopieren der Grafik-Datei in ihr Arbeitsverzeichnis

Wir möchten links in der Benutzeroberfläche die Grafik angezeigt bekommen.

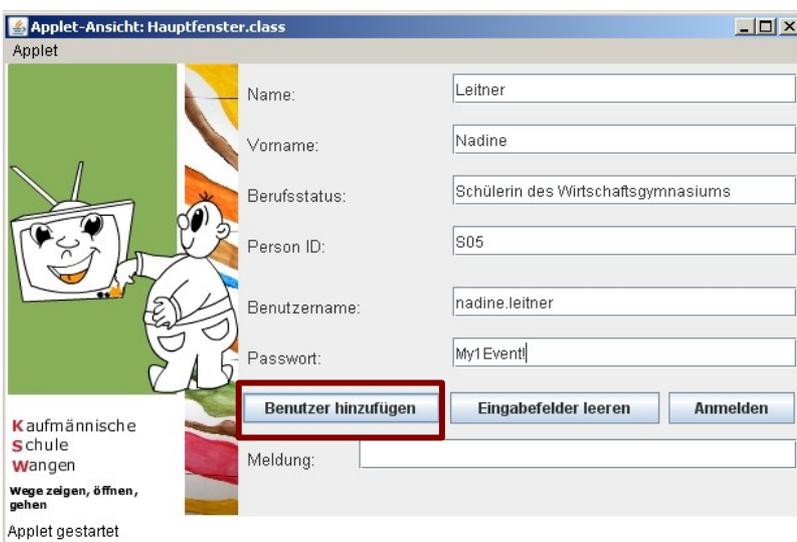
Kopieren Sie die Datei *background.jpg* (in *Eigene Dateien\JanC\UE06*) in Ihr Verzeichnis *Beziehungen*.

Schritt 9: Testen der Ereignis-Methode *Benutzer hinzufügen (Profil anlegen)*

Klicken Sie hierzu in der Symbol-Leiste die Schaltfläche (grüner Pfeil):



Klicken Sie auf die Schaltfläche *Eingabefelder leeren*, um die Inhalte der *Textfelder* zu leeren (die Ereignis-Methode der letzten Stunde)



Geben Sie die in der Grafik angezeigten Werte in die *Textfelder* ein und testen Sie die gerade erstellte Ereignis-Methode *Benutzer hinzufügen*, indem Sie auf die Schaltfläche *Benutzer hinzufügen* klicken.

Welche *Meldung* erscheint im Hauptfenster (der Benutzeroberfläche)?



Schritt 10: Quellcode der Ereignis-Methode *Anmelden* implementieren

Fügen Sie dazu anstelle des Kommentars : `// TODO hier Quelltext einfügen` in die Ereignis-Methode:

```
public void btAnmelden actionPerformed(ActionEvent evt) {  
    // TODO hier Quelltext einfügen  
}
```

folgenden Quellcode ein:

```
person1.identitaetPruefen(tfBenutzername.getText(),tfPasswort.getText());  
boolean mEr = person1.getErfolgreich();  
if(mEr == true){  
    tfMeldung.setText("Anmeldung erfolgreich");  
}else{  
    tfMeldung.setText("Anmeldung nicht erfolgreich");  
}
```

Wir haben Veränderungen im Quellcode durchgeführt. Um die Veränderungen testen zu können müssen wir den Quellcode *erneut* kompilieren:

Klicken Sie hierzu in der Symbol-Leiste die Schaltfläche (*grüner Pfeil*) :





Schritt 11: Testen der Ereignis-Methode *Anmelden* für den korrekten Fall

Wiederholen Sie Schritt 9, um die Benutzerdaten erneut *hinzuzufügen*! Hinweis: noch merkt sich das System nur die Daten welche bis zum erneuten kompilieren eingegeben wurden.

The screenshot shows a Java applet window titled 'Applet-Ansicht: Hauptfenster.class'. On the left, there is a cartoon character and a logo for 'Kaufmännische Schule Wangen'. The main area contains several text input fields: 'Name: tfName', 'Vorname: tfVorname', 'Berufsstatus: tfBerufsstatus', 'Person ID: tfPersonID', 'Benutzername: tfBenutzername', and 'Passwort: tfPasswort'. Below these fields are three buttons: 'Benutzer Hinzufügen', 'Eingabefelder leeren' (highlighted with a red box), and 'Anmelden'. At the bottom, there is a 'Meldung: tfMeldung' field. The status bar at the bottom left says 'Applet gestartet'.

Klicken Sie dann erneut auf die Schaltfläche *Eingabefelder leeren*, um die Inhalte der *Textfelder* zu leeren.

The screenshot shows the same applet window as above, but now the 'Benutzername' field contains 'nadine.leitner' and the 'Passwort' field contains 'My1Event!'. The 'Eingabefelder leeren' button is no longer highlighted. The 'Anmelden' button is now highlighted with a red box. The 'Meldung' field is empty. The status bar at the bottom left still says 'Applet gestartet'.

Testen Sie erst den Fall für „*Anmeldung erfolgreich*“, indem Sie für den Benutzernamen und das Passwort korrekte Eingaben in die *Textfelder Benutzername* und *Passwort* eingeben und die Schaltfläche *Anmelden* anklicken.

Hinweis:

„*korrekt*“ heißt die *hinzugefügten* Daten stimmen mit den bei der *Anmeldung* genutzten Daten überein!

Welche *Meldung* erscheint im Hauptfenster (der Benutzeroberfläche)?



Schritt 12: Testen der Ereignis-Methode *Anmelden* für den nicht-korrekten Fall

Applet-Ansicht: Hauptfenster.class
Applet

Name:

Vorname:

Berufsstatus:

Person ID:

Benutzername:

Passwort:

Meldung:

Kaufmännische
Schule
Wangen

Wege zeigen, öffnen,
gehen

Applet gestartet

Klicken Sie erneut auf die Schaltfläche *Eingabefelder leeren*, um die Inhalte der *Textfelder* zu leeren.

Applet-Ansicht: Hauptfenster.class
Applet

Name:

Vorname:

Berufsstatus:

Person ID:

Benutzername:

Passwort:

Meldung:

Kaufmännische
Schule
Wangen

Wege zeigen, öffnen,
gehen

Applet gestartet

Testen Sie nun den Fall „Anmeldung nicht erfolgreich“, indem Sie für den Benutzernamen und das Passwort nicht-korrekte Eingaben in die *Textfelder* *Benutzername* und *Passwort* eingeben und die Schaltfläche *Anmelden* anklicken.

Welche *Meldung* erscheint im Hauptfenster (der Benutzeroberfläche)?

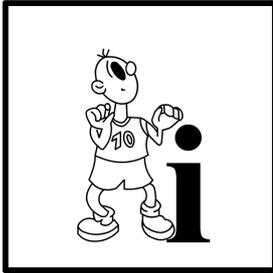
Herzlichen Glückwunsch!!

Ihre Anwendung ist nun fertig! Sie können nun auch die Datei *Hauptfenster.html* in Ihrem Ordner *Beziehungen* mit einem Doppelklick im Browser (IE oder Firefox) öffnen.

BEGINNEN SIE MIT PHASE 3 (LERNSTATION!)

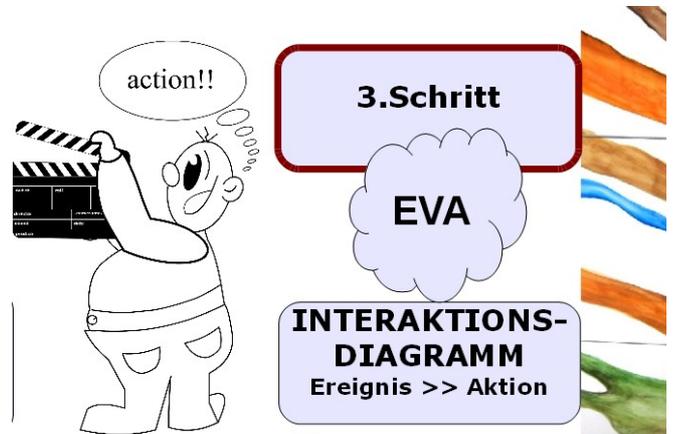


PHASE 3: INTERAKTIONSDIAGRAMM (*Impulse lösen Verhalten aus*)



Mit einem Interaktionsdiagramm kann man die Kommunikation zwischen mehreren Klassen grafisch darstellen. In UML erfasst man diese dynamischen Interaktionen

häufig in sog. Sequenzdiagrammen, einer speziellen Art von Interaktionsdiagrammen:



Sequenzdiagrammbezeichnung
(Tätigkeit: Ereignis >>Aktion)

Kaufmännische
Schule
Wange
Objektkonstruktion
(Assoziation)

sd Ereignis btsucheName_
sucheName (name)

Objekt einer Klasse
(Initiator)

Lifeline
(Zeitachse)

Objekt einer Klasse
(Kommunikationspartner)

: Hauptfenster

```
Quellcode: Person person1 = new Person();  
Diagrammnotation: new() oder create
```

person1: Person

[erfolgreich = wahr] Nachricht_1

```
Quellcode: person1.sucheName (tfName.getText ())  
Diagrammnotation: sucheName (name)  
Diagrammnotation Antwort: Name gefunden
```

[erfolgreich = falsch] Nachricht_2

```
Quellcode: person1.sucheName (tfName.getText ())  
Diagrammnotation: sucheName (name)  
Diagrammnotation Antwort: Name nicht gefunden
```

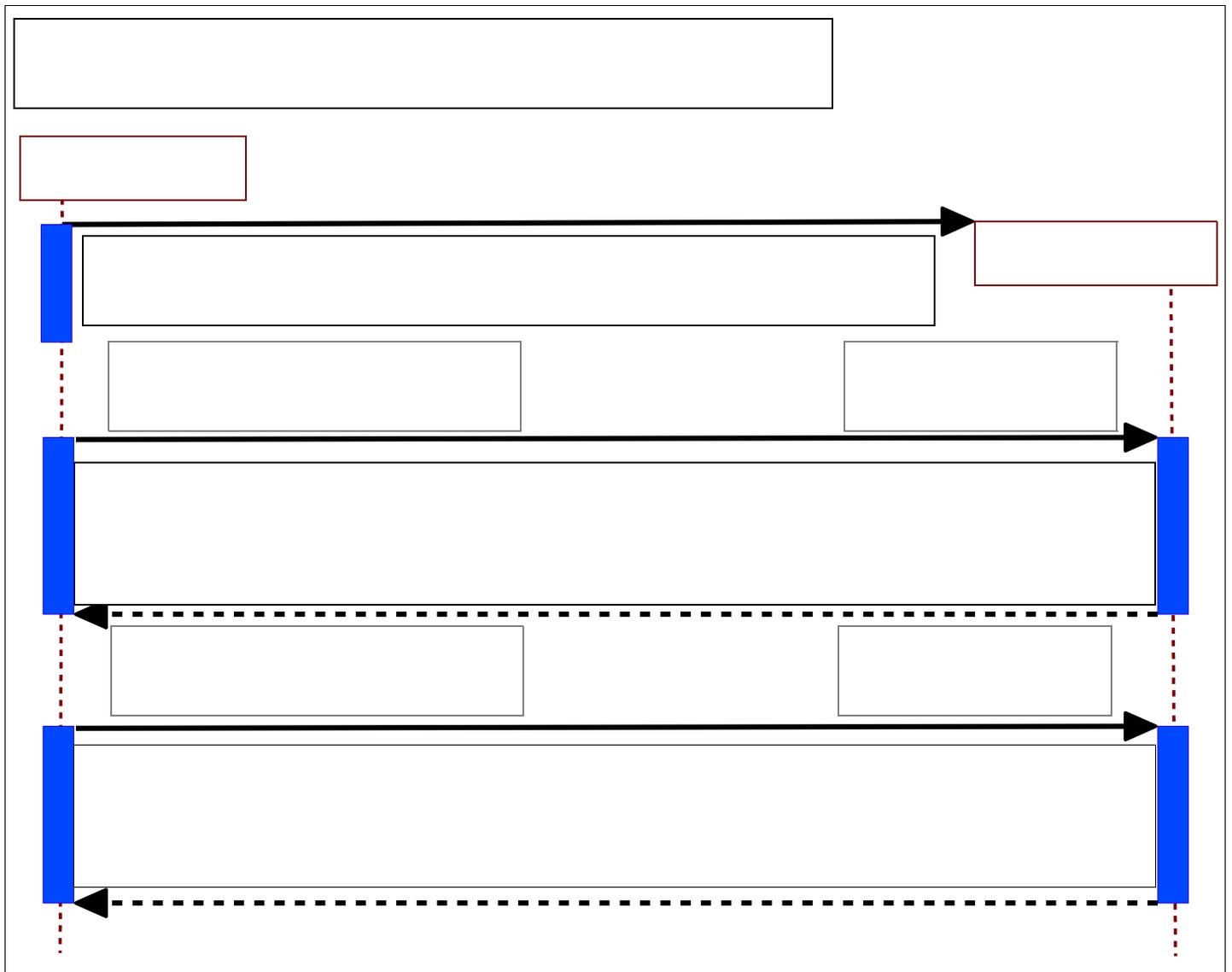
Nachricht , Verhaltensweise, enthaltene Tätigkeit
(Methodenaufruf)

Antwort , Verhaltensweise, enthaltene Tätigkeit
(z.B. Rückgabe eines Attributs)



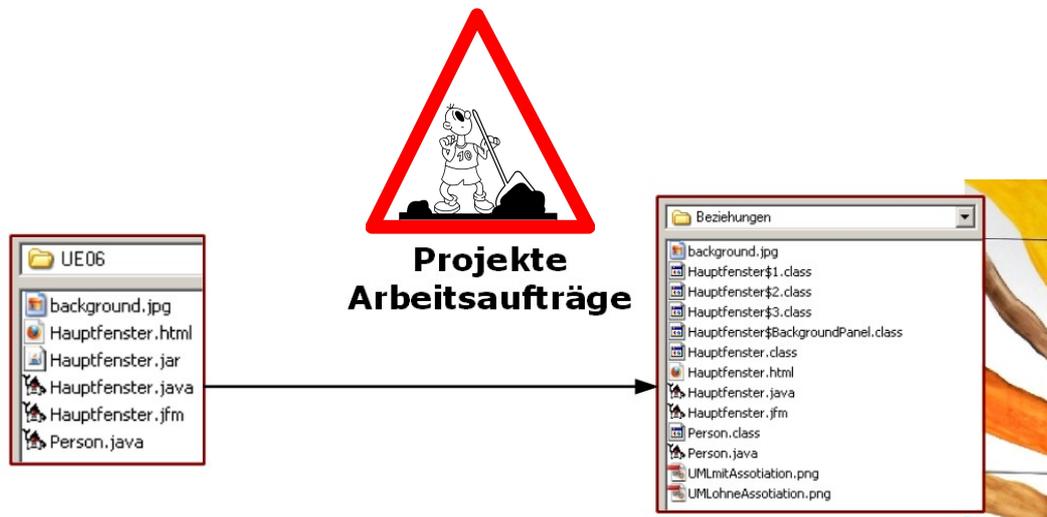
Schritt 13: Erlegen und übertragen Sie das Sequenzdiagramm für das Ereignis Anmelden

Hinweis: Arbeiten Sie im Team an der Lernstation





PHASE 4.1.: DOKUMENTATION / PRÄSENTATION



PHASE 1: ASSOZIATION – Leittext – Teamarbeit (15 min)

PHASE 2: EREIGNIS-METHODEN – Leittext – Teamarbeit (20 min)

PHASE 3: INTERAKTIONSDIAGRAMM – Lernstation (15 min)

PHASE 4.1: DOKUMENTATION (10 min)

PHASE 4.2: PRÄSENTATION – Team-Konferenz (10 min)

Autor: Christine Janischek
Thema: Beziehungen, Ereignisse und Aktionen

Kaufmännische
Schule
Wangen

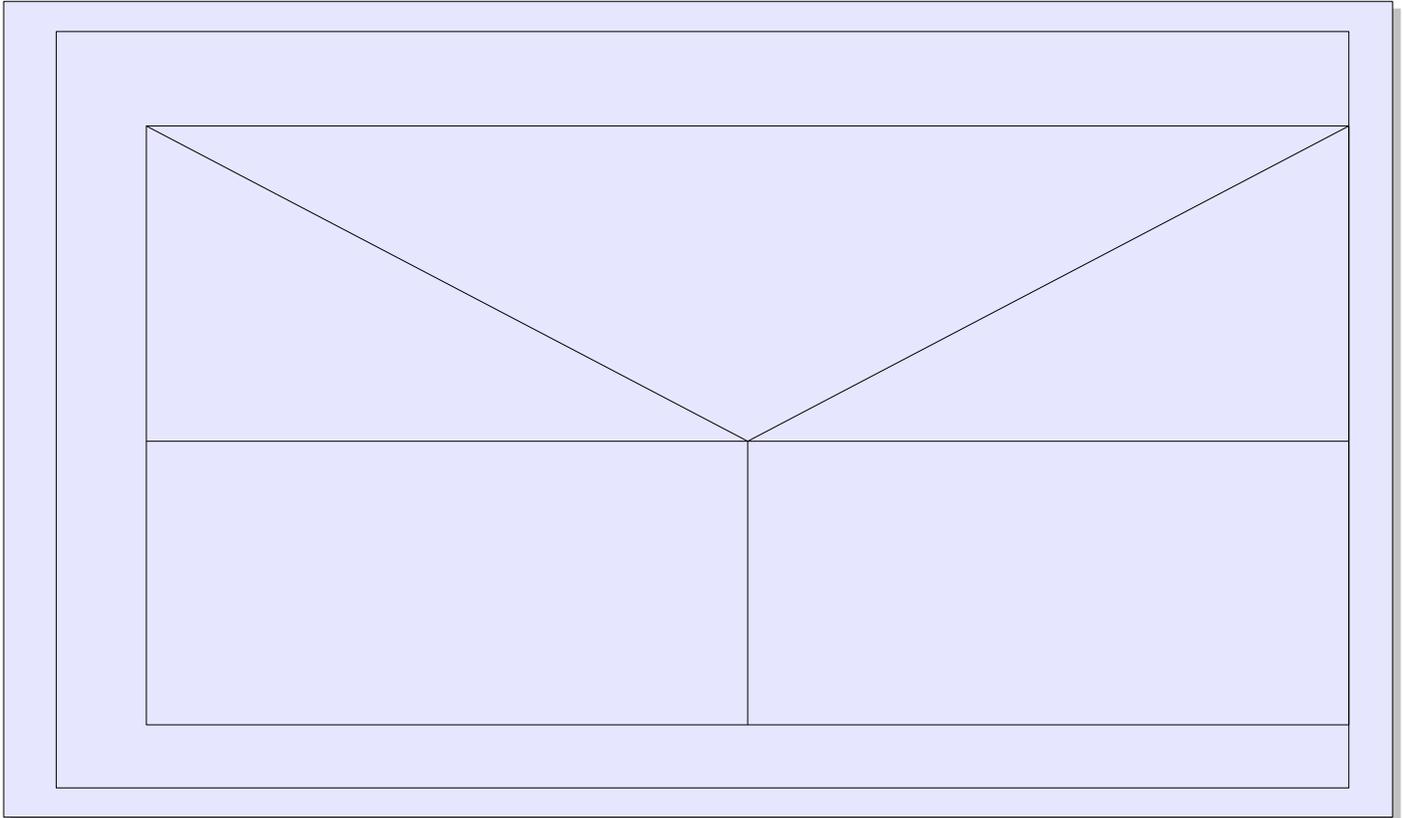
Wege zeigen, öffnen, gehen

Dokumentation und Notizen:



PHASE 4.2.: Zusatzaufgabe

Erstellen Sie ein *Struktogramm* für die Ereignis-Methode `btAnmelden_`



Kontrollfragen:



1. Welche Schritte wären notwendig, um *die Suche nach dem Namen* zu ermöglichen?
2. Welches Attribut der Fachklasse signalisiert dem Hauptfenster-Objekt ob die Anmeldung bzw. die Suche nach dem Namen erfolgreich war?
3. Wofür benötigt man die `equals()`-Methode (siehe Leittext der letzten Stunde)?