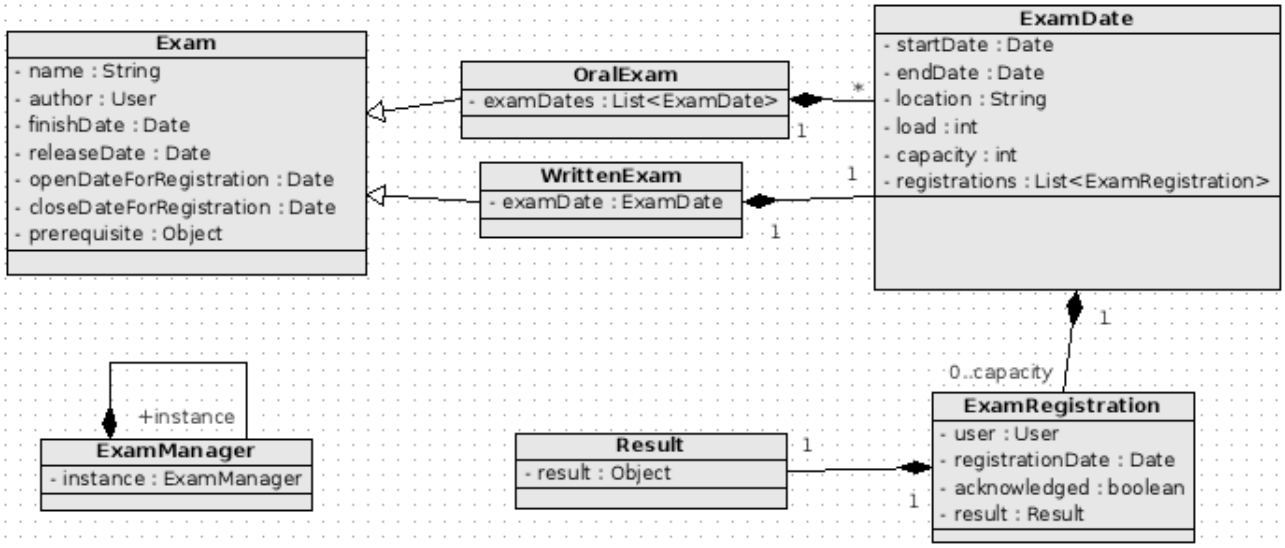
Aufgabenblatt 5

(Entwurfsbeschreibung)

1. Statisches Modell

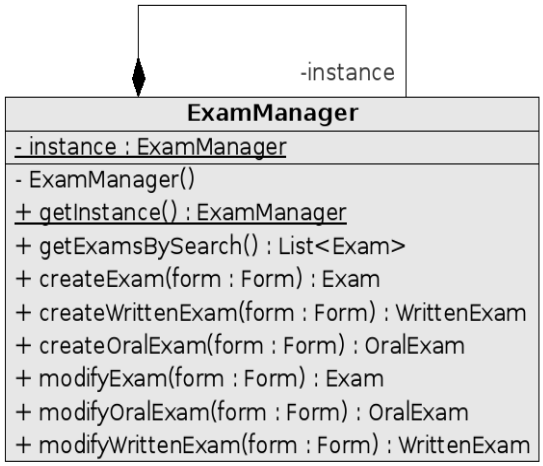
1.1. Klassendiagramm: innere Logik



Das Klassendiagramm beinhaltet ausschließlich die Komponenten der logischen Schicht der *paX*-Extension. Die Klasse *Exam* ist abstrakt. Dadurch wird gewährleistet, dass die Art einer Prüfung immer schon mit der Klasse definiert ist. Dies erlaubt auch ein einfaches Hinzufügen von weiteren Prüfungsarten in einer späteren (möglichen) Ausbaustufe. Das Diagramm zeigt auch die Mengen und Typen der Daten, die mit einer Prüfung (OralExam oder WrittenExam) assoziiert werden.

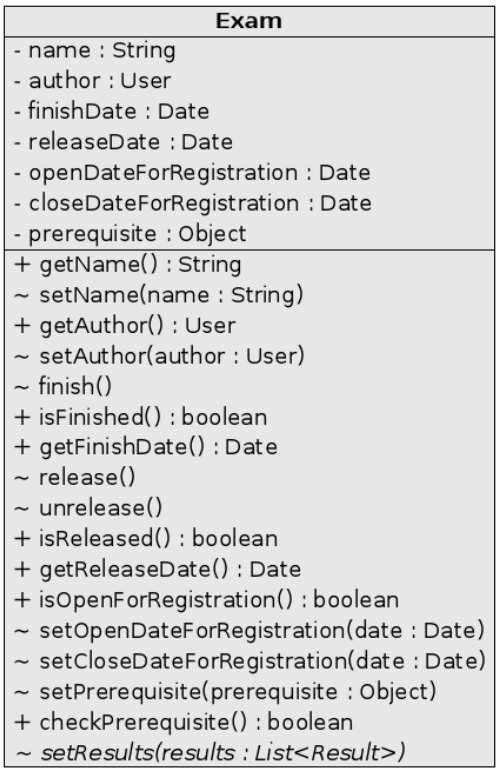
1.2. Klassen im Detail

1.2.1. ExamManager



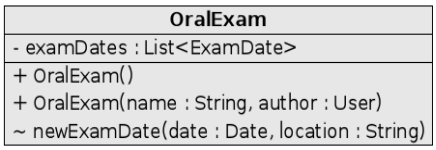
Der *ExamManager* ist lediglich ein Hilfsmittel zum Verwalten der Prüfungen. Er beinhaltet eine *mächtige* Suchfunktion, die sich im Aufbau des Methodenkopfes an den *OLAT*-Suchfunktionen orientiert. (Signatur wegen Übersichtlichkeit weggelassen)
Der Manager ist so konzipiert, dass er die ausgefüllte *Form* übergeben bekommt und anhand dessen die gewünschte Aktion ausführt. Er ist außerdem ein Singleton damit Synchronisierungsverfahren einfacher durchgeführt werden können und somit die Stabilität des Programms gewährleistet wird.

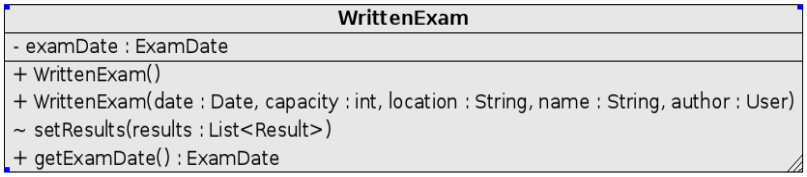
1.2.2. Exam



Die Klasse Exam wurde abstrakt gehalten, da die Unterschiedlichkeiten zwischen mündlicher und schriftlicher Prüfung zu groß sind, um sie in einer einzigen Klasse vereinen zu können.
Nicht zu sehen ist das Erben aus PersistentObject. Damit wurde gewährleistet bestimmte Standarteigenschaften, welche für Datenbanken gelten und bereits auf PersistentObject abgebildet wurden, implementiert werden.
Ihre Methoden entsprechen den zu erledigenden Aufgaben auf Datenebene, d.h. es werden Manipulationen der Daten durchgeführt um eine bestimmte Aufgabe zu erledigen und deren Ergebnis in den modifizierten Daten widerzuspiegeln.

1.2.3. OralExam & WrittenExam





Die Implementierungen der *Exam* Klasse unterscheiden sich in der jetzigen Form nur in den in der Arten der Prüfungstermine. So kann eine mündliche Prüfung (*OralExam*) mehrere Prüfungstermine haben, um mehrere Studenten prüfen zu können. Im Gegensatz dazu hat eine schriftliche Prüfung (*WrittenExam*) nur ein Prüfungsdatum.
Dementsprechend verändern sich auch die Methoden zum Holen der Daten und des Verwaltungsaufwandes um Instanzen dieser Klassen persistent zu machen und gültig zu halten.

1.2.4. ExamDate

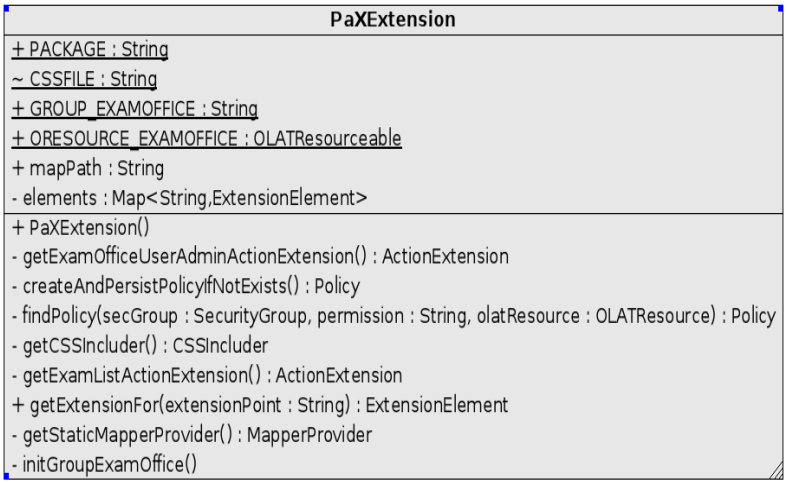
Diese Klasse representiert genau ein Termin einer Prüfung. Ein Termin besteht aus Ort, Anfangs- und Enddaten, sowie aus den angemeldeten Studenten (Benutzern) und die maximal möglichen Anmeldungen.

1.2.5. ExamRegistration

Diese Klasse enthält die Informationen, welche mit einer Prüfungsanmeldung assoziiert werden. Dazu gehören der angemeldete Student (Benutzer), die Anmeldungsbestätigung (oder Gültigkeit), das Ergebnis und der Anmeldezeitpunkt.

1.2.6. PaXExtension

Die PaXExtension ist die Hauptverwaltungsklasse der paX-Erweiterung. Diese implementiert das Extension-Interface und dementsprechend die dafür notwendigen Methoden.
Im Konstruktor wird mit Hilfte des ExtensionPoint Konzepts, die konkreten ExtensionElements erstellt und der Liste an Extensions hinzugefügt.
Weitere Methoden zum erweitern des OLAT Rollensystems sind hier implementiert.

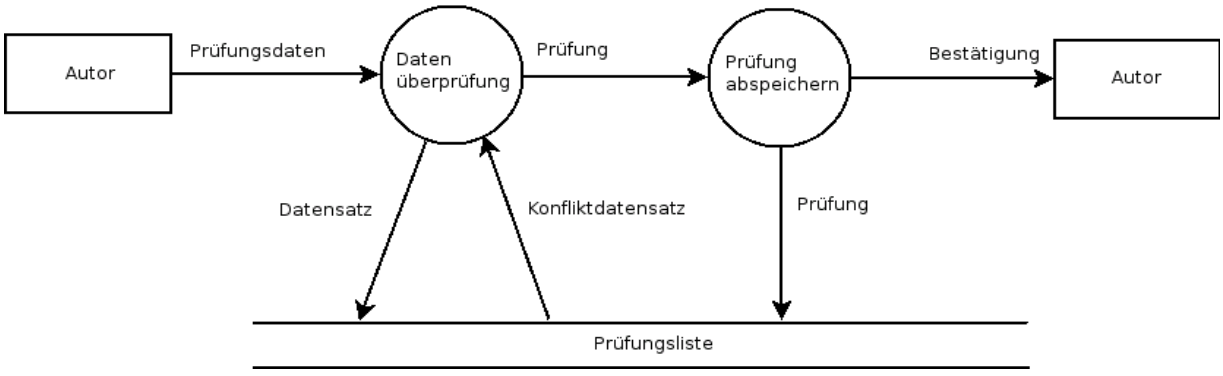


1.2.7. Weitere Klassen

Aus dem OLAT GUI-Framework leiten sich mehrere Dutzend Klassen ab, welche meist nur wenige Zeilen Code enthalten und zur Steuerung und Modellierung des Views genutzt werden. Diese Klassen werden während der Implementierungsphase erschlossen, da sie nach konkreten Vorgaben konstruiert werden müssen.

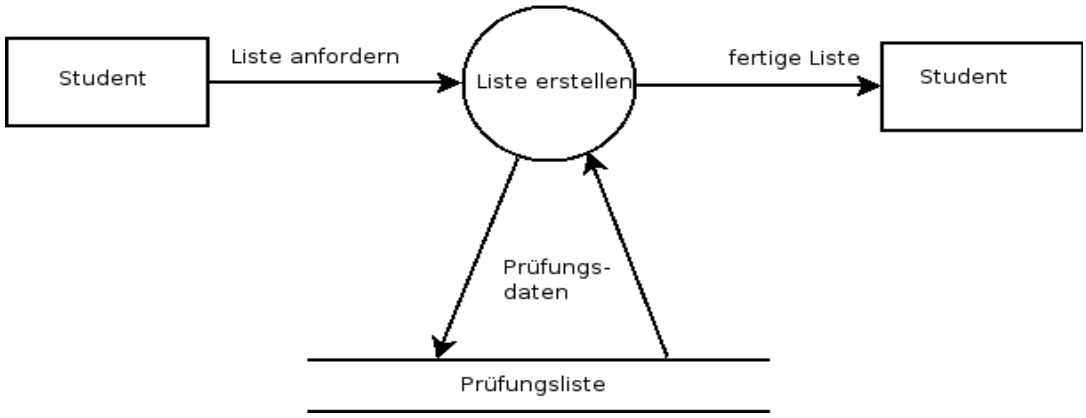
2. Dynamisches Modell

2.1. DFD[1] (Demokratischer Frauenbund Deutschlands)



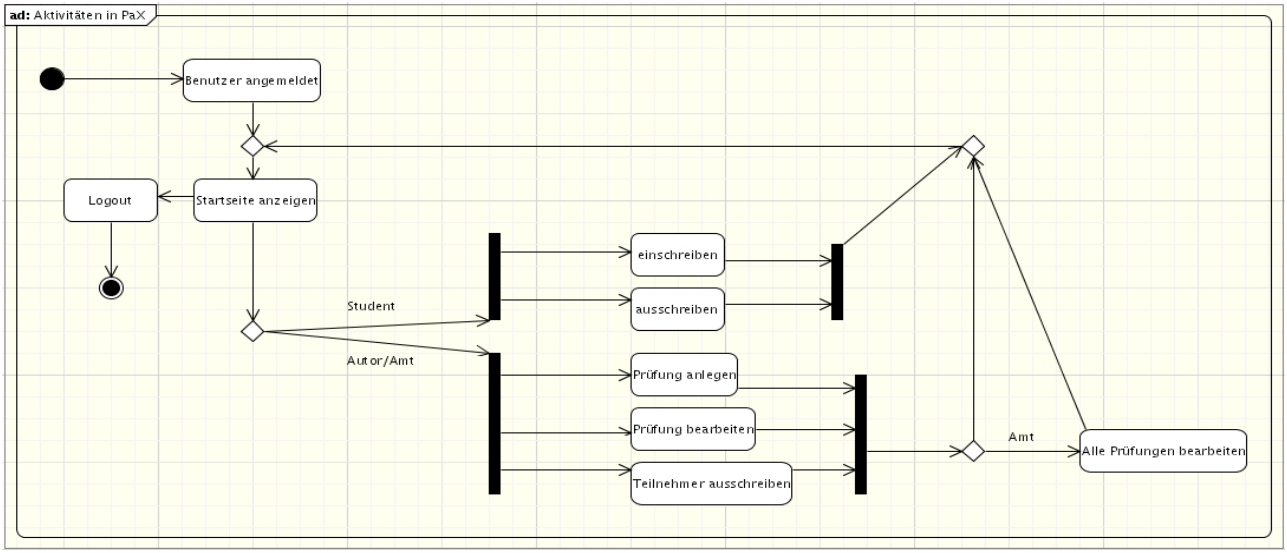
Dieses DFD verdeutlicht den Ablauf des Erstellens einer Prüfung durch einen Autor. Dieser gibt sämtliche prüfungsrelevanten Daten ein. Anschließend werden diese auf Validität überprüft (auch auf eventuelle Doppeleintragungen) und bei dessen Erfolg in das Prüfungsverzeichnis (*Prüfungsliste*) eingetragen.

[1]) Datenflussdiagramm



Hier wird durch ein DFD dargestellt, wie ein Student eine Prüfungsliste angezeigt bekommt. Die erstellte Liste wird dem Studenten dann angezeigt.

2.2. Aktivitätsdiagramm



Das Aktivitätsdiagramm zeigt alle in der paX-Erweiterung vorkommenden wesentlichen Abläufe. Voraussetzung für eine Benutzung der Erweiterung ist natürlich ein erfolgreicher Login in OLAT. Darauffolgend wird entsprechend der Rolle des Benutzers die Startseite der Erweiterung angezeigt. Von dieser sind dann sämtliche Funktionen (berechtigungsentsprechend) der Erweiterung zu erreichen.

2.3. Sequenzdiagramm

Folgendes Sequenzdiagramm stellt den allgemeinen Ablauf einer Benutzerinteraktion mit der Applikation dar. Dabei steht die paX-Extension ganz am Ende der Aufrufkette und stellt damit einen maximalen Abstraktions- und Modellierungsgrad dar. Die .init Aufrufe sind optional und werden nur einmal bei der Instanzierung des Servlets und der Extension ausgeführt. Dies geschieht im Normalfall nach dem ersten HTTP-Requests an das OLAT System, nachdem der Tomcat-Webserver gestartet wurde.