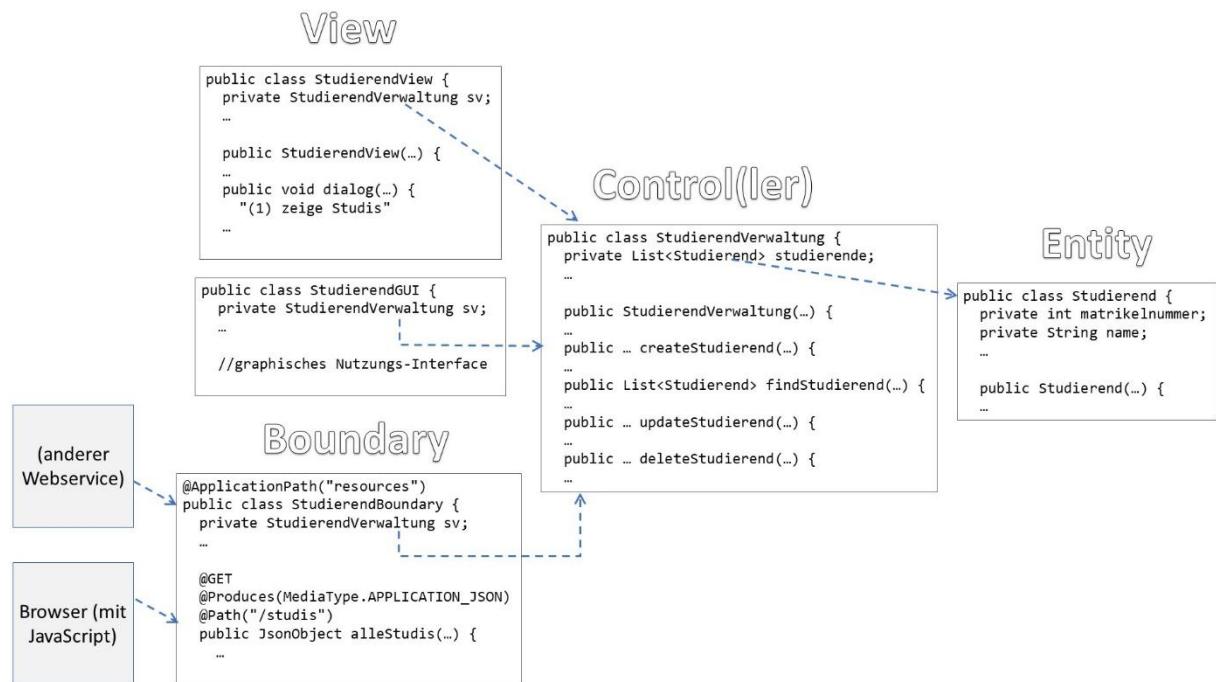


## Fragen, Antworten, Kommentare zur aktuellen Vorlesung

Frage: In den Programmieraufgaben sind die Klassen und Methoden genau benannt, die entwickelt werden sollen. Wann lernen wir welche Klassen man überhaupt mit welchen Methoden schreiben soll.

Generell ist die Idee, dass Sie durch typische Programmieraufgaben kontinuierlich mehr und mehr in der Lage sind, über eigene Klassen und Methoden nachzudenken. Dies passiert in der Vorlesung in kleinen Schritten und wird in nachfolgenden Veranstaltungen, insbesondere der Programmierung, Datenbanken (Modellierungsanteil) und Objektorientierte Analyse und Design (Design-Anteil) sowie Software-Architektur wesentlich vertieft.

Sie sollten bis jetzt gelernt haben, dass es für jedes relevante Datenobjekt eigene Klassen gibt (gerne Entities genannt) und dass jede Entity auch andere über Referenzen nutzen kann. Die Klasse Kreis nutzt die Klasse Punkt. Wir steigen mit den Listen in das Thema Verwaltung von Objekten ein. Es gibt typischerweise sogenannte Controller-Klassen, die Sammlungen anderer Objekte verwalten, das heißt insbesondere Objekte erzeugen (create/add), passende Objekte suchen (read/find), Objekte verändern (update) und Objekte löschen (delete, ergibt „CRUD“). Es kann eine oder meist mehrere Controller-Klassen in Projekten geben. Beim Säulendiagramm ist die Säule eine Entität und etwas irritierend die Säulendiagramm-Klasse ein Controller, wobei damit der Name SaeulenController technisch schöner wäre. Als Controller hat die Klasse recht wenige eigene Eigenschaften, dafür aber viele Referenzen zu verwalteten Objekten. Der dritte wichtige Klassentyp, gerne Boundary oder auch View genannt, regelt den Zugriff von außen auf den Controller. Als View gibt es bei uns die Text-Ein-Und-Ausgabe über eine Dialogklasse.



Würde die Säulendiagrammbearbeitung über das Web angeboten werden, würden Veränderungen z. B. über HTTP aufgerufen (z. B. REST-basiert) und der Begriff Boundary genutzt werden. Dieses, eher ins 4. Semester passende Konzept ist in der Abbildung skizziert. Es ist zu beachten, dass es

typischerweise von jeder Klassenart mehrere, also z. B. Controller mit verschiedenen Aufgaben, geben kann.

Unabhängig von den Details lernt man Programmieren nur durch Programmieren. Es ist die Standardempfehlung zumindest in der veranstaltungsfreien Zeit eigene Programmierprojekte durchzuführen. Sie können über solche Projekte und Ergebnisse auch mit mir reden. Personen ohne Programmiererfahrung sollten dabei erst bei Java bleiben, anderen sei der Einstieg in andere Programmiersprachen angeraten (für TI-Studierende z. B. Go, Rust, C).

Als eine Beispielidee kann versucht werden irgendein sehr einfaches Spiel mit dem Interaktionsbrett umzusetzen, dies ist bereits mit dem bisherigen Wissen möglich und wird in Richtung des Interaktionsbretts noch auf Aufgabenblatt 9 erweitert. Zwei kleine Beispielprogramme sollten neben diesem Dokument verlinkt sein. Da die Überprüfung der Performance eines Rechners nicht trivial ist, können die Programme deutlich zu schnell oder zu langsam laufen.

Hinweis: Die Beispielklausur ist online, sie zeigt die generelle Struktur der Klausur. Aufgaben 6 und 7 können auf alle Vorlesungsinhalte angepasst werden. Wer die Klausur noch nicht sehen möchte, sei darauf hingewiesen, dass es sehr sehr sinnvoll ist, die Messreihe-Aufgaben zunächst auf Papier zu lösen.