

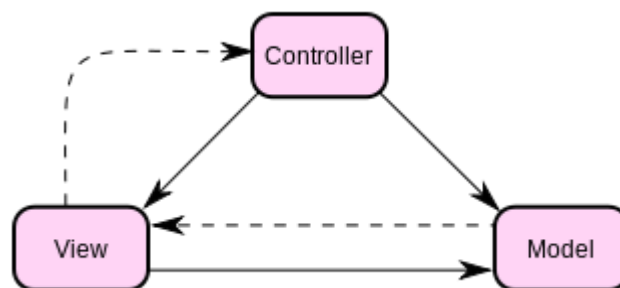
MVC

DEFINITION

Model View Controller (MVC, englisch für Modell-Präsentation-Steuerung) ist ein Muster zur Unterteilung einer Software in die drei Komponenten Datenmodell (englisch model), Präsentation (englisch view) und Programmsteuerung (englisch controller). (Wikipedia)

MVC ist eine Art zu Programmieren. Sie gibt 3 Kategorien vor welche beschreiben, welche Files bzw. Klassen für welchen Teil der Software zuständig ist.

KOMPONENTEN



CONTROLLER

Die Controller Klassen verwalten die Model und View Klassen.

Meistens stellen sie auch den Einstiegspunkt einer Funktion dar. Wenn in einer MVC Applikation eine Aktion durchgeführt wird, bedeutet dies, dass eine Funktion einer Controller Klasse aufgerufen wird.

Im Controller werden Daten des Models geladen und für die View aufbereitet.

Elemente eines Controllers	Keine Elemente eines Controllers
<ul style="list-style-type: none">- Daten vom Model laden- Daten um modellieren- Laden einer View	<ul style="list-style-type: none">- Angaben, wie Daten dargestellt werden (HTML)- Laden von Daten aus der Datenbank (SQL)

MODEL

Die Model Klassen sind die Schnittstelle zur persistenten Datenspeicherung.

Sie abstrahieren den Datenspeicher einer Applikation soweit, dass es für den Controller irrelevant wird, ob die Daten in einer MySQL, MSSQL oder als Text Files gespeichert werden.

Die Model Klassen sind die einzigen Klassen welchen es erlaubt ist, auf diese Daten zuzugreifen und sie zu speichern. In bestimmten Fällen ist das Model auch für die Validierung der Daten zuständig.

Elemente eines Models	Keine Elemente eines Models
<ul style="list-style-type: none"> - Daten laden - Daten speichern - Datenkonsistenz prüfen - (Validierungen vornehmen) 	<ul style="list-style-type: none"> - Daten verändern -

... ES GIBT FÜR ALLES EINE AUSNAHME!

Angenommen, man speichert für alle seine Daten einen Zeitstempel sowie die ID des Benutzers welche diese Daten zuletzt geändert hat. Es kann nun Sinn ergeben, diese Daten in der Model Klasse selbst zu setzen.

VIEW

Die View ist für die Darstellung der im Controller geladenen Daten zuständig.

In den meisten MVC Implementierungen spricht man nicht von View Klassen, sondern von Template Files. Jedoch werden auch in diesen Implementierungen die Template Files von einer einzigen View Klasse geladen und mit Daten versorgt.

Elemente einer View	Keine Elemente einer View
<ul style="list-style-type: none"> - Daten darstellen - Daten formatieren 	<ul style="list-style-type: none"> - Daten verändern - Daten speichern

VORTEILE

MODULARER AUFBAU

Wie erwähnt ist der Controller der Einstiegspunkt einer MVC Applikation und der Teil welcher die anderen beiden Komponenten verwaltet („kontrolliert“).

Daraus ergibt sich einer der grossen Vorteile von MVC. Der Controller kann, abhängig von Benutzereingaben und Parametern, die View und das Model dynamisch laden. So kann eine Funktion im Controller welche z.B. Personendaten anzeigen soll, wählen ob er eine HTML View oder eine JSON View laden soll.

SEPARATION OF CONCERNS

„Trennung der Sorgen“ ist ein Programmierprinzip welches besagt, dass unterschiedliche Komponenten einer Software eine klar definierte Aufgabe (die „Sorgen“) haben. MVC erfüllt dieses Prinzip, da alle Komponenten einen klar definierten Zweck haben.

AUFGABENTRENNUNG

Jeder Entwickler hat unterschiedliche Stärken. Setzt man ein System wie MVC ein, so können alle Entwickler an jedem Teil arbeiten welcher ihnen liegt.

Der Frontend Spezialist kann sich voll und ganz auf die schöne Darstellung der Daten konzentrieren. Er muss nur wissen, welche Daten er vom Controller Spezialisten zur Verfügung gestellt bekommt. Der Datenbankprofi kann die Aufgabe übernehmen, die Daten redundant und konsistent auf mehrere DB Systeme zu speichern – er muss dem Controller nur angeben mit welcher Funktion er Daten laden und Daten speichern kann.