



CODERS.BAY

MVC

- Klassendesign wird in drei verschiedene Kategorien unterteilt
 - **Model**

Für jede "Entität" in der Datenbankstruktur gibt es eine eigene Model-Klasse
z.B.: eine Model-Klasse für User = Tabelle user
Model-Klassen sind eng gekoppelt an die Datenbankstruktur
sorgen dafür, dass Daten aus der Datenbank geladen und gespeichert werden
 - **View**

Darstellung der Daten verantwortlich
sind Templates um die Website mittels HTML und CSS auszugeben
keine Logik implementiert, d.h. es erfolgen keine Datenbankabfragen und keine Überprüfungen von Werten
kümmern sich bloß um die Darstellung der Anwendung
 - **Controller**

Daten (der Model) mit Präsentationsschicht (Views) verheiratet sind
hier werden Logiken für Anwendungen implementiert
Bsp.: Controller bekommt von View die Login-Daten, diese überprüft der Controller ob Benutzer existiert und das Passwort korrekt ist. Falls alle Eingaben korrekt waren -> Controller veranlasst die Darstellung der nächsten Seite

- klare Trennung zwischen
 - Verarbeitung von Daten
 - Darstellung der Daten im View
 - Anwendungslogik im Controller
- bei Änderungen von Datenbankstrukturen (z.B. Spaltenname wird umbenannt) müssen diese Änderungen nur in der Model-Klasse geändert werden
Anwendungslogik (Controller) und Darstellungsschicht (View) bleiben meist unberührt
- Trennung macht es deutlich leichter mit mehreren Personen an ein und der selben Webanwendung zu arbeiten
- für jede größere Anwendung nützlich und sinnvoll
- Implementierung von Controller-Klassen kann kompliziert sein, sofern man auf kein MVC-Framework zurückgreift



DAS MODEL



- Zugriff auf Datenbanktabellen
- pro Datenbanktabelle ein entsprechende Model-Klasse, die die Anfrage auf diese Datenbanktabelle steuert



DIE VIEW



- wie werden Daten dargestellt
- sind HTML Templates in die Daten reingeladen werden
- viele PHP-Frameworks (z.B.: Laravel) bieten eine Template-Engine an, mit der die Darstellung der Daten von der Logik für die Datenverarbeitung getrennt werden kann
bei Laravel ist es `.blade`



DER CONTROLLER



- implementiert Anwendungslogik
- z.B.: neuer Benutzer registrieren
 - entsprechende Daten aus dem Formular abfragen
 - neuer Datenbankeintrag erstellen
 - ggf. Willkommensmail senden
 - Benutzer anzeigen, dass Registrierung erfolgreich war