

Model-View-Controller-Paradigma/Controller

Wechseln zu: [Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#) **nur teilweise**:

Korrektheit: 4 (größtenteils überprüft)	Umfang: 3 (einige wichtige Fakten fehlen)	Quellenangaben: 4 (fast vollständig vorhanden)	Quellenarten: 4 (sehr gut)	Konformität: 4 (sehr gut)
---	---	--	--------------------------------------	-------------------------------------

Diese Bewertungen beziehen sich auf alle im nachfolgenden Menü genannten Artikel gleichermaßen.

MVC-Paradigma: [Model \(Data\) | View | Controller](#)

MVC-Pattern: [Singletons](#) | [Dependency Injection](#) | [Observers](#)

MVCS-Paradigma: [Service](#)

MVCS-Pattern: [Singletons](#) | [Dependency Injection](#) | [Observers](#)

LDVCS-Paradigma: [Logic](#)

VCLSD-Pattern: [Singletons](#) | [Dependency Injection](#) | [Observers](#)

Inhaltsverzeichnis

- [1 Definition \(Kowarschick \(MMProg\)\)](#)
- [2 Anmerkungen](#)
- [3 Beispiele](#)
- [4 Quellen](#)
- [5 Siehe auch](#)

1 Definition ([Kowarschick \(MMProg\)](#))

Controller (engl. [controllers](#)) dienen zur Steuerung einer [MVC-](#), [MVCS-](#) oder [LDVCS-Anwendung](#). Dazu nimmt ein Controller Eingaben aus verschiedensten Quellen entgegen (z.B. Sensor-Daten oder Daten, die ein Benutzer über eine beliebige Benutzer-Schnittstelle wie eine Tastatur oder eine Maus eingibt) und leitet diese bereinigt und normalisiert an ein [Modell](#) bzw. – im Falle von LDVCS-Anwendungen – an ein [Datenmodul](#) weiter.

Ein MVC- bzw. ein MVCS-Controller kann weitere Aufgaben übernehmen:

Umsetzung der Komponentenlogik

Verarbeitung von Systemsignalen, wie z.B. einer Systemuhr (z.B. „Spielzeit ist abgelaufen“)

Ein MVC-Controller kann darüber hinaus noch eine Aufgabe übernehmen:

Kommunikation mit externen Datenquellen (zum Zweck der Datensynchronisation)

Der MVC-Prozess

Der MVCS-Prozess

Der LDVCS-Prozess

2 Anmerkungen

Controller von MVC-Anwendungen und [Controller von MVCS-Anwendungen](#) unterscheiden sich nur in einer Hinsicht: MVC-Controller könnten für die Kommunikation mit der Außenwelt zuständig sein, allerdings werden dafür normalerweise [Modelle](#) eingesetzt. MVCS-Modelle sind dagegen nie für die Kommunikation mit der Außenwelt zuständig. Diese Aufgabe übernehmen [MVCS-Service-Module](#).

[Controller von LDVCS-Anwendungen](#) sind weder für die Kommunikation mit der Außenwelt noch für die Komponentenlogik zuständig. Für diese Aufgaben gibt es [LDVCS-Service-Module](#) bzw. [LDVCS-Logic-Module](#).

3 Beispiele

Tastatur-Controller fangen Tastatur-Ereignisse ab und ermitteln, welches Zeichen der Benutzer mit der gewählten Tastenkombination eingegeben hat. Das so ermittelte Zeichen kann dann weiterverarbeitet werden. Die Ermittlung des vom Benutzer eingegebenen Zeichens kann, wenn die zugehörige Programmiersprache dies nicht sauber unterstützt (wie z.B. [ActionScript 3](#)), sehr aufwendig sein, da die Tastencodes nichts über die Zeichen aussagen, die auf der Tastatur aufgedruckt sind – die Zuordnung zwischen Tastencode und Tasteninhalte hängt vom gewählten Tastaturlayout ab.

Device-Controller fangen die Ereignisse von bestimmten [Devices](#), wie USB-Geräten, Infrarot-Steuerungen, Mäusen etc. ab und verarbeiten diese weiter.

Data-Input-Controller fangen die Daten, die ein Benutzer über eine View (wie z.B. ein HTML-Formular) eingibt, ab und verarbeiten diese weiter.

Etc. pp.

4 Quellen

Kowarschick (MMProg): [Wolfgang Kowarschick](#); Vorlesung „Multimedia-Programmierung“; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; [Quellengüte](#): 3 (Vorlesung)

5 Siehe auch

[Logic](#)

[Model-View-Controller-Paradigma](#) (insb. die dort angegebenen Quellen)

Kategorien:

[MVC](#)

[Objektorientierte Programmierung](#)

[Glossar](#)

Diese Seite wurde zuletzt am 22. September 2017 um 13:53 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).



