

Logic-Data-View-Controller-Service-Paradigma/Logic

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 Quellen](#)
- [4 Siehe auch](#)

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

Ein Logikmodul (engl. [logic module](#)) einer [LDVCS-Anwendung](#) dient zur Realisierung der Anwendungslogik.

Diese Module werden i. Allg. durch externe Ereignisse (wie [Controlleraktionen](#) oder Änderungen am [Datenbestand](#)) aktiviert.

Wenn ein Logikmodul aktiviert wurde, kann es sowohl den Inhalt von beliebigen [LDVCS-Datenmodulen](#) modifizieren (gegebenenfalls nachdem es zuvor lesend auf den aktuellen Datenbestand zugegriffen hat), als auch beliebige [LDVCS-Services](#) aktivieren.

Der LDVCS-Prozess

2 Anmerkungen

Die Idee, [MVC-Modelle](#) explizit in zwei Komponenten, nämlich [LDVCS-Datenmodule](#) und LDVCS-Logikmodule aufzuteilen geht auf Wolfgang Kowarschick^[1] zurück.

Der Vorteil dieser Aufteilung ist vor allem darin zu sehen, dass im [MVC-Paradigma](#) nicht eindeutig festgelegt worden ist, welche der beiden Module „[Modell](#)“ oder „[Controller](#)“ eigentlich für die Realisierung der Anwendungslogik zuständig ist. Man findet für beide Möglichkeiten zahlreiche Beispiele.

3 Quellen

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

4 Siehe auch

[Model \(MVC\)](#)
[Controller \(MVC\)](#)
[Service \(VCLSD\)](#)

Kategorien:

[MVC](#)

[Objektorientierte Programmierung](#)

[Glossar](#)

Diese Seite wurde zuletzt am 22. September 2017 um 16:49 Uhr bearbeitet.
Inhalt verfügbar unter [CC BY-SA 4.0](#).

