

Seite 12. UML-Anwendungsfalldiagramm-Trainer

UML-Anwendungsfalldiagramm-Trainer

Dieser interaktive Trainer gehört zur Theorie-Seite:

UML-Anwendungsfalldiagramm

Hier übst du, aus einer Aufgabenbeschreibung ein korrektes UML-Anwendungsfalldiagramm abzuleiten.

Im Fokus stehen:

Akteure
Systemgrenze
Use Cases
Assoziationen
<<include>>
<<extend>>
Pfeilrichtung
Pflichtbestandteil vs. optionale Erweiterung

Was wird trainiert?

Bereich	Bedeutung
Akteure erkennen	externe Rollen außerhalb des Systems
Systemgrenze verstehen	Rechteck um die Use Cases
Use Cases benennen	Funktionen des Systems als Ellipsen
Assoziationen erkennen	einfache Linien zwischen Akteur und Use Case
<> erkennen	Pflichtbestandteil eines Use Cases
<> erkennen	optionale Erweiterung eines Use Cases
Pfeilrichtung prüfen	include und extend zeigen in unterschiedliche fachliche Richtungen

Interaktiver UML-Anwendungsfalldiagramm-Trainer

<https://trainer.ulrich-wiki.com/uml-anwendungsfalldiagramm-trainer.html?v=6>

Merksatz für den Trainer

Akteur = externe Rolle außerhalb des Systems
Use Case = Funktion des Systems
Systemgrenze = Rechteck um die Use Cases
Akteur-Use Case = einfache Linie ohne Pfeil
<<include>> = Pflichtbestandteil
<<extend>> = optionale Erweiterung
include zeigt zum eingebundenen Pflicht-Use-Case
extend zeigt zum Basis-Use-Case

Beispiel: Shop-System

Aufgabenstellung:

Ein Kunde kann Artikel suchen und eine Bestellung aufgeben.
Beim Aufgeben einer Bestellung muss eine Zahlung durchgeführt werden.
Vor der Bestellung kann optional ein Rabatt geprüft werden.
Ein Admin kann Artikel verwalten.

Mögliche Lösung:

System:
Shop-System

Akteure:
Kunde
Admin

Use Cases:
Artikel suchen
Bestellung aufgeben
Zahlung durchführen
Rabatt prüfen
Artikel verwalten

Beziehungen:

Kunde — Artikel suchen

Kunde — Bestellung aufgeben

Admin — Artikel verwalten

Bestellung aufgeben <<include>> Zahlung durchführen

Rabatt prüfen <<extend>> Bestellung aufgeben

Warum ist „Zahlung durchführen“ ein <>?

Bestellung aufgeben <<include>> Zahlung durchführen

Das bedeutet:

Wenn eine Bestellung aufgegeben wird,
muss die Zahlung durchgeführt werden.

Also ist die Zahlung ein **Pflichtbestandteil**.

Warum ist „Rabatt prüfen“ ein <>?

Rabatt prüfen <<extend>> Bestellung aufgeben

Das bedeutet:

Rabatt prüfen erweitert Bestellung aufgeben nur optional.

Zum Beispiel:

nur wenn ein Rabattcode vorhanden ist
nur wenn eine Aktion aktiv ist
nur wenn der Kunde berechtigt ist

Richtige Pfeilrichtung

Beziehung	Richtung
<>	vom Basis-Use-Case zum eingebundenen Pflicht-Use-Case
<>	vom optionalen Erweiterungs-Use-Case zum Basis-Use-Case

Merksatz:

include zeigt auf das, was immer gebraucht wird.
 extend zeigt auf das, was optional erweitert wird.

Typische Fehler

Fehler	Warum falsch?
Akteur innerhalb der Systemgrenze	Akteure stehen außerhalb
Use Case außerhalb der Systemgrenze	Use Cases gehören ins System
Pfeile bei Akteur-Verbindung	Akteur-Use Case ist normalerweise einfache Linie
include und extend vertauscht	Pflicht und Option werden falsch dargestellt
include-Pfeil falsch herum	include zeigt auf den Pflicht-Use-Case
extend-Pfeil falsch herum	extend zeigt auf den Basis-Use-Case
technische Methoden als Use Cases	Use Cases sollen Nutzersicht zeigen

Mini-Testfragen

1. Wo stehen Akteure im Anwendungsfalldiagramm?

Akteure stehen außerhalb der Systemgrenze.

2. Wie werden Use Cases dargestellt?

Use Cases werden als Ellipsen dargestellt.

Beispiel:

(Bestellung aufgeben)

3. Wie wird eine Akteur-Use-Case-Beziehung dargestellt?

Als einfache durchgezogene Linie ohne Pfeilspitze.

Beispiel:

Kunde — Bestellung aufgeben

4. Was bedeutet <>?

`<<include>>` bedeutet Pflichtbestandteil.

Beispiel:

Bestellung aufgeben `<<include>>` Zahlung durchführen

5. Was bedeutet <>?

`<<extend>>` bedeutet optionale Erweiterung.

Beispiel:

Rabatt prüfen `<<extend>>` Bestellung aufgeben

6. In welche Richtung zeigt include?

`<<include>>` zeigt vom Basis-Use-Case zum eingebundenen Pflicht-Use-Case.

Bestellung aufgeben → Zahlung durchführen

7. In welche Richtung zeigt extend?

`<<extend>>` zeigt vom optionalen Erweiterungs-Use-Case zum Basis-Use-Case.

Rabatt prüfen → Bestellung aufgeben

8. Warum ist Zahlung durchführen im Shop-Beispiel include?

Weil die Zahlung ein notwendiger Bestandteil der Bestellung ist.

Ohne Zahlung ist die Bestellung nicht vollständig.

9. Warum ist Rabatt prüfen im Shop-Beispiel extend?

Weil der Rabatt nur optional geprüft wird.

Nur wenn ein Rabattcode oder eine Rabattbedingung vorhanden ist.

10. Was zeigt ein Anwendungsfalldiagramm nicht?

Es zeigt nicht den inneren Programmablauf.

Nicht dargestellt werden:

if-Anweisungen
Schleifen
Methodenaufrufe
Datenbanktabellen

Dafür wären andere Diagramme besser geeignet.

Nächste Seite

Danach geht es weiter mit:

UML-Sequenzdiagramm

Updated 28 May 2026 12:19:04 by Admin