

Seite 14. UML-Sequenzdiagramm-Trainer

Dieser interaktive Trainer gehört zur Theorie-Seite:

UML-Sequenzdiagramm

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

Im Fokus stehen:

Teilnehmer
Lebenslinien
Nachrichten
Rückgaben
Reihenfolge
Aktivierungsbalken
parallele Prozessbereiche

Was wird trainiert?

Bereich	Bedeutung
Teilnehmer erkennen	beteiligte Akteure, Objekte oder Systemteile
Lebenslinien verstehen	gestrichelte Linien unter den Teilnehmern
Nachrichten eintragen	durchgezogene Pfeile als Aufrufe
Rückgaben unterscheiden	gestrichelte Pfeile als Antworten
Reihenfolge prüfen	Ablauf wird von oben nach unten gelesen
Parallelbereich erkennen	zwei parallele Linien markieren parallele Abläufe
Beschriftungen sauber setzen	Methodenaufrufe und Rückgaben eindeutig benennen

Interaktiver UML-Sequenzdiagramm-Trainer

<https://trainer.ulrich-wiki.com/uml-sequenzdiagramm-trainer.html?v=1>

[UML-Sequenzdiagramm-Trainer im Vollbild öffnen](https://trainer.ulrich-wiki.com/uml-sequenzdiagramm-trainer.html?v=1)

Merksatz für den Trainer

Teilnehmer stehen oben.
Lebenslinien laufen nach unten.
Zeit läuft von oben nach unten.
Nachrichten sind durchgezogene Pfeile.
Rückgaben sind gestrichelte Pfeile.
Aktivierungsbalken zeigen Verarbeitung.
Parallele Abläufe müssen sichtbar markiert werden.

Wichtige fachliche Einordnung

Im Trainer verwenden wir für den parallelen Bereich eine **IHK-nahe vereinfachte Darstellung mit zwei parallelen Linien**.

Das bedeutet:

zwei parallele Linien = paralleler Ablaufbereich

Streng nach UML kann Parallelität auch als `par`-Combined-Fragment mit Rahmen dargestellt werden.

Für dein IHK-Lernen ist wichtig:

Du sollst erkennen:
Hier laufen zwei Prozesse parallel oder unabhängig voneinander ab.

Beispiel: Login prüfen

Aufgabenstellung:

Ein Benutzer gibt Login-Daten ein.
Die LoginGUI ruft den LoginService auf.
Der LoginService fragt das UserRepository ab.
Danach werden Dashboarddaten und Benutzerkontext parallel geladen.
Zum Schluss wird die Startseite angezeigt.

Mögliche Lösung:

Teilnehmer:

Benutzer

loginGUI:LoginGUI

loginService:LoginService

userRepository:UserRepository

dashboard:Dashboard

Ablauf:

Benutzer → loginGUI: loginDatenEingeben(name, passwort)

loginGUI → loginService: pruefeLogin(name, passwort)

loginService → userRepository: findeBenutzer(name)

userRepository --> loginService: benutzer

loginService --> loginGUI: loginStatus

Parallelbereich:

Prozess A:

loginGUI → dashboard: ladeDashboard()

dashboard --> loginGUI: dashboardDaten

Prozess B:

loginGUI → loginService: ladeBenutzerKontext()

loginService --> loginGUI: benutzerKontext

Abschluss:

loginGUI → Benutzer: startseiteAnzeigen()

Nachricht oder Rückgabe?

Darstellung	Bedeutung
durchgezogener Pfeil	Aufruf / Nachricht
gestrichelter Pfeil	Rückgabe / Antwort
senkrechte gestrichelte Linie	Lebenslinie
schmaler Balken auf Lebenslinie	Aktivierungsbalken

Darstellung	Bedeutung
zwei parallele Linien	paralleler Ablaufbereich

Teilnehmer richtig benennen

Ein Teilnehmer kann ein Akteur oder ein Objekt sein.

Beispiele:

```
Benutzer
loginGUI:LoginGUI
loginService:LoginService
userRepository:UserRepository
dashboard:Dashboard
```

Wichtig:

```
Benutzer = Akteur
loginGUI:LoginGUI = Objekt / Instanz mit Klasse
```

Warum steht die Zeit von oben nach unten?

Ein Sequenzdiagramm wird vertikal gelesen.

```
oben = früher
unten = später
```

Das bedeutet:

```
Die erste Nachricht steht oben.
Spätere Nachrichten stehen weiter unten.
```

Typische Fehler im Sequenzdiagramm

Fehler	Warum falsch?
Teilnehmer fehlen	unklar, wer beteiligt ist
Lebenslinien fehlen	Ablauf ist nicht als Sequenzdiagramm erkennbar

Fehler	Warum falsch?
Rückgaben als normale Nachrichten gezeichnet	Antwort wirkt wie neuer Aufruf
Nachrichten in falscher Reihenfolge	Ablauf wird fachlich falsch
Pfeile überlagern sich	Diagramm ist schlecht lesbar
paralleler Bereich nicht markiert	Gleichzeitigkeit wird nicht sichtbar
zu lange Texte auf Pfeilen	Diagramm wird unübersichtlich
Klassen statt konkreter Teilnehmer falsch verwendet	Sequenzdiagramm zeigt konkrete Kommunikation

Mini-Testfragen

1. Wofür verwendet man ein UML-Sequenzdiagramm?

Ein Sequenzdiagramm zeigt den zeitlichen Nachrichtenaustausch zwischen Teilnehmern oder Objekten.

Merksatz:

Wer ruft wen wann auf?

2. Wie liest man ein Sequenzdiagramm?

Von oben nach unten.

oben = früher
unten = später

3. Was ist eine Lebenslinie?

Eine Lebenslinie ist die gestrichelte senkrechte Linie unter einem Teilnehmer.

Sie zeigt, dass der Teilnehmer während des Ablaufs existiert.

4. Wie wird eine normale Nachricht dargestellt?

Als durchgezogener Pfeil zwischen zwei Teilnehmern.

Beispiel:

```
loginGUI → loginService: pruefeLogin()
```

5. Wie wird eine Rückgabe dargestellt?

Als gestrichelter Pfeil zurück.

Beispiel:

```
loginService --> loginGUI: loginStatus
```

6. Was zeigt ein Aktivierungsbalken?

Ein Aktivierungsbalken zeigt, dass ein Teilnehmer gerade aktiv ist oder etwas verarbeitet.

7. Was bedeuten zwei parallele Linien in unserem Trainer?

Sie markieren einen parallelen Ablaufbereich.

Das bedeutet:

```
Mehrere Prozesse können parallel oder unabhängig voneinander ablaufen.
```

8. Was ist der Unterschied zwischen Nachricht und Rückgabe?

Eine Nachricht ruft etwas auf.

Eine Rückgabe liefert ein Ergebnis zurück.

```
Nachricht = Aufruf
```

```
Rückgabe = Antwort / Ergebnis
```

9. Warum dürfen Pfeile und Texte nicht überlappen?

Weil sonst unklar wird, welcher Pfeil zu welcher Nachricht gehört.

Ein Sequenzdiagramm muss sauber lesbar bleiben.

10. Was ist ein typischer IHK-Fehler?

Ein häufiger Fehler ist, die zeitliche Reihenfolge falsch darzustellen.

Wichtig:

Die Reihenfolge ergibt sich von oben nach unten.

Nächste Seite

Danach geht es weiter mit:

Mock-up / Benutzeroberflächen-Entwurf

Revision #2

Created 28 May 2026 12:43:56 by Admin

Updated 28 May 2026 12:48:38 by Admin