

# Seite 10. UML-Aktivitätsdiagramm-Trainer

## UML-Aktivitätsdiagramm-Trainer

Dieser interaktive Trainer gehört zur Theorie-Seite:

UML-Aktivitätsdiagramm

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

Im Fokus stehen:

Startknoten  
Aktionen  
Entscheidungsraute  
Guards  
Kontrollflüsse  
Merge-Knoten  
Endknoten

### Was wird trainiert?

Bereich	Bedeutung
<b>Start erkennen</b>	Der Ablauf beginnt mit einem gefüllten Kreis
<b>Aktionen eintragen</b>	Arbeitsschritte als abgerundete Rechtecke
<b>Entscheidung erkennen</b>	Verzweigung als Raute
<b>Guards eintragen</b>	Bedingungen wie <code>[ja]</code> und <code>[nein]</code> an den Pfeilen
<b>Merge-Knoten erkennen</b>	Alternative Wege werden wieder zusammengeführt
<b>Ende erkennen</b>	Kreis mit gefülltem Punkt
<b>Ablaufreihenfolge verstehen</b>	Pfeile zeigen den Kontrollfluss

### Interaktiver UML-Aktivitätsdiagramm-Trainer

<https://trainer.ulrich-wiki.com/uml-aktivitaetsdiagramm-trainer.html?v=3>

UML-Aktivitätsdiagramm-Trainer im Vollbild öffnen

---

## Merksatz für den Trainer

Startknoten = gefüllter Kreis  
Aktion = abgerundetes Rechteck  
Entscheidung = Raute  
Guard = Bedingung am Pfeil  
Merge = Raute zur Zusammenführung  
Endknoten = Kreis mit gefülltem Punkt  
Kontrollfluss = Pfeilrichtung im Ablauf

---

## Beispiel: Produkt speichern

Aufgabenstellung:

Ein Benutzer gibt Produktdaten ein.  
Das System prüft die Eingabe.  
Wenn die Eingabe gültig ist, wird das Produkt gespeichert.  
Wenn die Eingabe ungültig ist, wird eine Fehlermeldung angezeigt.  
Danach kehrt das System ins Menü zurück.

Mögliche Lösung:

```
Start
↓
Produkt Daten eingeben
↓
Eingabe prüfen
↓
Eingabe gültig?
├─ [ja] → Produkt speichern
└─ [nein] → Fehler anzeigen
↓
Merge
↓
Zurück ins Menü
↓
Ende
```

## Warum ist die Raute wichtig?

Eine Entscheidung wird im UML-Aktivitätsdiagramm als **Raute** dargestellt.

Nicht als Rechteck.

Aktion = abgerundetes Rechteck

Entscheidung = Raute

Merge = Raute

Der Unterschied liegt im Ablauf:

Symbol	Bedeutung
<b>Entscheidung</b>	ein Eingang, mehrere Ausgänge
<b>Merge</b>	mehrere Eingänge, ein Ausgang

## Warum stehen **[ja]** und **[nein]** an den Pfeilen?

Die Bedingungen heißen **Guards**.

Sie beschreiben, wann ein bestimmter Weg genommen wird.

Beispiel:

Eingabe gültig?

└─ [ja] → Produkt speichern

└─ [nein] → Fehler anzeigen

Wichtig:

[ja] und [nein] gehören an die ausgehenden Kontrollflüsse der Entscheidung.

## Typische Aufgaben im Trainer

Der Trainer enthält Aufgaben zu:

Produkt speichern

Login prüfen

Preis prüfen

Dabei musst du jeweils erkennen:

Welche Aktion kommt zuerst?  
Welche Prüfung folgt?  
Wie lautet die Entscheidungsfrage?  
Welche Guards gehören an die Pfeile?  
Welche Aktion gehört zum Ja-Zweig?  
Welche Aktion gehört zum Nein-Zweig?  
Wo werden die Wege wieder zusammengeführt?

## Typische Fehler

Fehler	Warum falsch?
Entscheidung als Rechteck	Eine Entscheidung muss als Raute dargestellt werden
Guards fehlen	Es ist nicht klar, welcher Pfad wann gilt
[ja] und [nein] stehen an falscher Stelle	Guards gehören an die Kontrollflüsse
Merge-Knoten fehlt	Alternative Wege werden unsauber zusammengeführt
Pfeile zeigen in falsche Richtung	Ablaufreihenfolge wird falsch
Aktion zu lang formuliert	Aktionen sollten kurz und eindeutig sein
Start- oder Endknoten fehlt	Ablauf wirkt unvollständig

## Mini-Testfragen

### 1. Wie wird der Startknoten dargestellt?

Der Startknoten wird als gefüllter Kreis dargestellt.



### 2. Wie wird eine Aktion dargestellt?

Eine Aktion wird als abgerundetes Rechteck dargestellt.

Beispiel:

Produkt speichern

### 3. Wie wird eine Entscheidung dargestellt?

Eine Entscheidung wird als Raute dargestellt.

Beispiel:

Eingabe gültig?

### 4. Was sind Guards?

Guards sind Bedingungen an den Kontrollflüssen.

Beispiele:

[ja]  
[nein]  
[gültig]  
[ungültig]

### 5. Was ist ein Merge-Knoten?

Ein Merge-Knoten führt alternative Ablaufwege wieder zusammen.

Er wird ebenfalls als Raute dargestellt.

mehrere Eingänge → Merge → ein Ausgang

### 6. Was ist der Unterschied zwischen Entscheidung und Merge?

Eine Entscheidung teilt den Ablauf.

Ein Merge führt alternative Wege wieder zusammen.

Entscheidung = ein Eingang, mehrere Ausgänge

Merge = mehrere Eingänge, ein Ausgang

## 7. Wie wird der Endknoten dargestellt?

Der Endknoten wird als Kreis mit gefülltem Punkt dargestellt.



## 8. Wo gehören `[ja]` und `[nein]` hin?

Sie gehören an die ausgehenden Pfeile der Entscheidungsraute.

Eingabe gültig?

└─ [ja] → Speichern

└─ [nein] → Fehler anzeigen

## 9. Warum ist ein Merge nach einer Entscheidung sinnvoll?

Wenn beide alternativen Wege danach wieder gemeinsam weiterlaufen, führt ein Merge-Knoten diese Wege sauber zusammen.

Beispiel:

[ja] → Speichern ↴

Merge → Zurück ins Menü

[nein] → Fehler ↴

## 10. Was zeigt der Kontrollfluss?

Der Kontrollfluss zeigt die Reihenfolge der Aktionen.

Er wird als Pfeil dargestellt.

## Nächste Seite

Danach geht es weiter mit:

UML-Anwendungsfalldiagramm

---

Revision #2

Created 28 May 2026 10:57:10 by Admin

Updated 28 May 2026 12:22:24 by Admin