

LU01.A10 - Anforderungen deklarativ formulieren

Formulieren Sie Schritt-für-Schritt-Anforderungen so um, dass sie den gewünschten **Endzustand** beschreiben – und in der Gegenrichtung deklarative Anforderungen in eine konkrete Handlungsabfolge.

Ausgangslage

Anforderungen werden im Betrieb oft als Handlungsanleitung formuliert: „Zuerst öffnest du ..., dann prüfst du ..., dann schreibst du ...“. Das ist **imperativ**. Wer so schreibt, legt die Lösung bereits fest – und verbaut sich möglicherweise die bessere.

Eine **deklarative** Anforderung beschreibt stattdessen das Ergebnis: Woran erkennt man, dass die Aufgabe erledigt ist? Sie lässt offen, wie das System dorthin gelangt.

Faustregel

Deklarative Anforderungen enthalten keine Verben wie „durchlaufen“, „öffnen“, „zwischenspeichern“, „hochzählen“. Sie enthalten Sätze der Form: „... *enthält genau ...*“, „... *ist sortiert nach ...*“, „... *gilt für alle ...*“.

Teil A: imperativ → deklarativ

Formulieren Sie jede der folgenden Anforderungen in **ein bis zwei Sätzen** als Beschreibung des Endzustands. Nennen Sie ausserdem, welche Wörter der Vorlage typisch imperativ waren.

1. „Öffne die Datei `kunden.csv`. Lies sie Zeile für Zeile ein. Prüfe bei jeder Zeile, ob die Postleitzahl mit einer 8 beginnt. Wenn ja, hänge die Zeile an eine neue Datei `zuerich.csv` an. Schliesse am Schluss beide Dateien.“
2. „Setze einen Zähler auf 0. Gehe alle Bestellungen durch. Wenn eine Bestellung mehr als 500 Franken kostet, erhöhe den Zähler um 1. Gib den Zähler am Ende aus.“
3. „Nimm die erste Note. Merke sie dir als Maximum. Gehe alle weiteren Noten durch und ersetze das Maximum, sobald du eine grössere Note findest. Gib zum Schluss das Maximum aus.“
4. „Lege eine leere Liste an. Durchlaufe alle Lernenden. Wenn ein Lernender in allen drei Fächern mindestens eine 4 hat, füge seinen Namen der Liste hinzu. Sortiere die Liste am Ende alphabetisch.“
5. „Warte, bis die Datei eingetroffen ist. Prüfe die Dateigrösse. Wenn sie grösser als 0 ist, verschiebe sie in den Ordner `verarbeitet`. Andernfalls verschiebe sie nach `fehler` und schreibe einen Eintrag ins Logfile.“

Teil B: deklarativ → imperativ

Hier ist die Anforderung bereits deklarativ formuliert. Beschreiben Sie in nummerierten Schritten, was ein Programm konkret tun muss, um diesen Endzustand herzustellen.

1. „Die Ausgabeliste enthält für jede Abteilung genau einen Eintrag mit dem Namen der Abteilung und der Anzahl Mitarbeitender, absteigend nach Anzahl sortiert.“
2. „Nach dem Import gibt es keine zwei Kundendatensätze mit derselben E-Mail-Adresse. Bei Dubletten bleibt der zuletzt geänderte Datensatz erhalten.“

Teil C: Reflexion

1. Bei welcher der fünf Anforderungen aus Teil A war die Umformulierung am schwierigsten? Woran lag das?
2. Anforderung 5 aus Teil A enthält etwas, das sich **nicht** rein deklarativ als Endzustand beschreiben lässt. Was ist es, und warum?
3. Welchen Vorteil hat eine deklarative Anforderung für die Person, die das Programm später schreibt? Und welchen Nachteil?

Beispiel zur Orientierung

Imperativ (Vorlage)	Deklarativ (Umformulierung)
„Gehe alle Artikel durch, addiere Menge mal Preis auf eine Summenvariable und gib sie am Schluss aus.“	„Der ausgewiesene Lagerwert entspricht der Summe aus Menge mal Preis über alle Artikel.“

Typisch imperative Wörter in der Vorlage: „*gehe durch*“, „*addiere auf*“, „*am Schluss*“ – sie beschreiben die Reihenfolge der Arbeitsschritte, nicht das Ergebnis.

Abgabe

Geben Sie ein Dokument mit den Teilen A, B und C in Moodle ab.

[M323-LU01](#), [M323-B1I](#), [M323-B1A](#)

© Kevin Maurizi

From:
<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu01/aufgaben/anforderungendeklarativ>

Last update: 2026/08/18 10:20



