

Lernzettel

Problemlösungskette: Von der Problemstellung
zur Abstraktion und Modellbildung

Universität:	Technische Universität Berlin
Kurs/Modul:	Einführung in die Informatik
Erstellungsdatum:	September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Einführung in die Informatik

Lernzettel: Problemlösungskette: Von der Problemstellung zur Abstraktion und Modellbildung

(1) Ziel und Kontext. Die Problemlösungskette beschreibt den Weg von einer konkreten Problemstellung hin zu einer abstrakten Beschreibung und einem Modell, auf dessen Basis eine Implementierung erstellt wird. Der Fokus liegt auf dem Übergang von realen Anforderungen zu einer modellierten Struktur, die sich in einer Programmiersprache umsetzen lässt.

(2) Vorgehensschritte.

- Verstehen der Problemstellung: Ziele, Randbedingungen, Einschränkungen identifizieren.
- Abstraktion: Relevante Aspekte herausfiltern, Unwichtiges ausblenden.
- Modellbildung: Entwurf von Modellen (Daten- und Verhaltensmodell), Auswahl passender Abstraktionsebenen.
- Formalisierung: Modelle als Strukturen (Klassen, Typen) und Algorithmen darstellen.
- Validierung: Modell mit Beispielen prüfen, Anforderungen verifizieren.
- Implementierung: Modell in eine Programmiersprache überführen, OO-Prinzipien berücksichtigen.

(3) Abstraktion – Typische Strategien.

- Abstrakte Datentypen (ADTs) definieren Schnittstellen, Implementierung verbergen.
- Objektorientierte Abstraktion: Klassen bündeln Daten und Verhalten.
- Abstrakte Klassen und Interfaces: Gemeinsame Merkmale trennen von konkreter Implementierung.
- Modelltypen: Strukturmodell vs. Verhaltensmodell; Zustands- vs. Interaktionsmodell.

(4) Modellbildung – Grundlagen.

- Datenmodell: Welche Objekte existieren? Welche Attribute haben sie? Beziehungen?
- Verhaltensmodell: Welche Operationen/Methoden existieren? Wie interagieren Objekte?
- Zustand und Übergänge: Welche Zustände? Welche Ereignisse ändern den Zustand?

(5) Von der Abstraktion zur Implementierung.

- Mapping: Abstraktionen zu Klassen, Attributen und Methoden.
- Vererbung und Polymorphie: Wiederverwendung und Flexibilität im OO-Design.
- Schnittstellen/Abstrakte Klassen: Trennung von Spezifikation und Implementierung.

(6) Kurzes Beispiel. Problemstellung: Eine einfache Aufgabenliste mit Datum und Priorität soll verwaltet werden. **Abstraktion:** Objekte Aufgabe mit Attributen wie Titel, Deadline, Priorität; Verhalten: erledigen(), verschieben(). **Modellbildung:** Klassen: Aufgabe(titel, deadline, priorität); Beziehungen: eine Aufgabe gehört zu einem Termin. **Implementierungshinweis:** Nutze OO-Prinzipien: Kapselung, Vererbung, Schnittstellen.