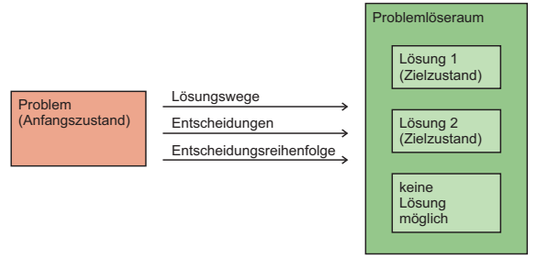
Inhaltsübersicht

In 10 Schritten vom Problem zur Lösung

Um zu einem gegebenen Problem eine geeignete Problemlösung in Form eines Programms zu finden, sollten folgende Hinweise beachtet werden:



Es wird von folgenden **Voraussetzungen** ausgegangen:

- Das gegebene Problem ist algorithmisch, d.h. in Form eines Programms, lösbar.
- Es stehen zur Problemlösung *nur* strukturierte und prozedurale Programmierkonzepte einschließlich der Datenabstraktion zur Verfügung.
- Das Problem und die gewünschte Lösung sind verstanden und ausreichend spezifiziert.
- Die Problemlösung besteht aus wenigen Seiten Programmquellcode.

In der Regel sollte folgende **Entscheidungsreihenfolge** eingehalten werden:

1. Prüfen, ob ein ähnliches oder **vergleichbares Problem** einschließlich der Problemlösung bereits bekannt ist. Wenn ja, dann die Lösung übernehmen und unter Umständen anpassen.
2. Prüfen, ob ein **allgemeineres Problem** einschließlich der Problemlösung bereits bekannt ist. Wenn ja, dann überlegen, ob das zu lösende Problem als Sonderfall der allgemeinen Problemlösung behandelt werden kann.
3. Bei komplexen Daten überlegen, welche **Datenstruktur** oder welche Datenstrukturen für die Problemlösung geeignet sind (z.B. Felder, siehe Vom Problem zur Lösung: Teil 3, S. 202). Mögliche Alternativen durchspielen.
4. **Algorithmen** konzipieren, die das Problem lösen. Liegt eine Datenstruktur oder liegen mehrere Datenstrukturen vor, dann prüfen, ob die konzipierten Algorithmen die Datenstruktur(en) optimal »bearbeiten«.
5. Die Algorithmen in **Prozeduren und Funktionen** unterteilen, so dass sie jeweils nur ein Problem lösen bzw. eine Aufgabe erledigen und im Umfang überschaubar sind (siehe Vom Problem zur Lösung: Teil 4, S. 267).
6. Bei der Konzeption von Prozeduren und Funktionen darauf achten, dass die Anzahl und Art der **Parameter** optimal gewählt wird, um eine allgemeine und flexible Lösung zu gewährleisten.
7. Prüfen, ob eine **rekursive** Lösung besser als eine iterative ist oder nicht.
8. Für jede Prozedur und Funktion zuerst die geeigneten **Kontrollstrukturen** (Wiederholung, Auswahl) (siehe Vom Problem zur Lösung: Teil 2, S. 164) ermitteln und anschließend die **einfachen Anweisungen** (siehe Vom Problem zur Lösung: Teil 1, S. 95) programmieren.
9. Prüfen, ob eine **Datenabstraktion** benötigt wird, d.h. die Datenstruktur muss noch zur Verfügung stehen, wenn einzelne Algorithmen in Form von Prozeduren und/oder Funktionen beendet sind.
10. Folgende Prüfungen sind durchzuführen:
    a. Gibt es alternative Lösungen?
    b. Kann die Lösung verallgemeinert werden?
    c. Ist die Lösung erweiterbar?
    d. Ist die Lösung defensiv programmiert?
    e. Wurden geeignete Sprachelemente verwendet?

Es empfiehlt sich folgende **iterative Vorgehensweise** (siehe Vom Problem zur Lösung: Teil 4, S. 267):

1. Zuerst Erstellung einer umgangssprachlichen oder semiformalen Lösungsbeschreibung.
2. Dann Umsetzung in die formale Syntax der gewählten Programmiersprache.

Iterativ bedeutet, dass schrittweise eine Lösungsbeschreibung erstellt, programmiert und getestet wird, anschließend der nächste Schritt der Lösung erstellt, programmiert und getestet wird usw.