

2.1 Funktionen/Top-Down-Methode

2.1.1 Problem

Bei der *Erstellung* der bisherigen Programme traten einige Probleme auf:

- Die Programme werden schnell sehr umfangreich und unübersichtlich. Auch kleine Aufgabenstellungen erfordern oft viel Aufwand zu ihrer Lösung.
- Aber kaum ein Mensch kann mehr als 100 Zeilen vollständig im Kopf behalten oder mehr als zehn Variablen. Menschlicher Geist und Computer scheinen einfach nicht kompatibel.
- Schon bei kurzen Programmen können einfach dadurch, dass bestimmte Ausnahmen eintreten (Überlauf, Division durch null, Rundungsfehler), Programmfehler passieren. In der Folge stürzt das Programm ab (nicht so schlimm) oder man erhält – ohne es zu merken – falsche Ergebnisse (viel schlimmer!).

Trotzdem gibt es in der Praxis auch funktionierende Programme mit Hunderttausenden von Zeilen. Wie kann man das selbst auch hinkriegen?

2.1.2 Top-Down-Strategie

Eine mögliche Lösung für dieses Problem ist die Top-Down-Strategie.

- Eine nicht sofort lösbare Aufgabe wird in Unteraufgaben aufgeteilt. Es wird quasi die Arbeit an die Unteraufgaben delegiert.
- Jede dieser Unteraufgaben wird wieder in Teilaufgaben aufgeteilt.
- Dieses Vorgehen wird so lange wiederholt, bis man auf Probleme stößt, die sofort und ohne viel Nachdenken lösbar sind (so genannte triviale Probleme).
- Nun werden alle diese Aufgaben der untersten Ebene gelöst.
- Zum Schluss werden sie wieder zusammengefügt zur Gesamtlösung.

Diese Vorgehensweise wird oft auch als Methode der schrittweisen (bzw. sukzessiven) Verfeinerung, als *Bausteintechnik* oder als *Modultechnik* bezeichnet, die einzelnen Teilaufgaben heißen dann Bausteine oder Module.

In der Elektrotechnik ist diese Bausteintechnik ebenso weit verbreitet: Größere Anlagen werden in Schränke unterteilt, die Schränke in Einschübe (z.B. in 19-Zoll-Technik), die Einschübe enthalten Steckkarten.¹

Diese Vorgehensweise hat manche Vorteile (in der Elektrotechnik wie auch in der Programmierung):

- Jeder Baustein kann einzeln für sich geprüft und für gut befunden werden.
- Aus der Anzahl der fertigen Bausteine kann man den Fortschritt des Projektes abschätzen.
- Vielleicht kann man den einen oder anderen Baustein in einem anderen Projekt wiederverwenden (*Software reuse*).
- Man kann fehlende Bausteine zunächst durch so genannte *Dummies* (Baustein-Hüllen ohne Funktion) ersetzen und damit sehen, ob das Gesamtsystem so funktionieren kann.²

¹Das Kraftwerk-Kennzeichensystem KKS z.B. basiert auf einer solchen Unterteilung.

²Mit diesen Dummies kann man z.B. die graphische Benutzeroberfläche eines Programmes simulieren, ohne dass hinter den einzelnen Menüpunkten bereits die volle Funktionalität vorhanden ist.

2.1.3 Beispiel

Ein Programm soll einen Brief schreiben. Im Zeitalter der Serienbriefe ist das Normalität. Zuerst teilen wir die Aufgabe auf. Ein Brief besteht allgemein aus:

- Briefkopf
- Hauptteil
- Anhang

In einem zweiten Schritt kann man das Schreiben des Briefkopfes aufteilen. Ein Briefkopf besteht aus:

- Absender
- Datum
- Anschrift
- Betreff
- Bezug

Genauso kann man den Hauptteil und den Anhang aufteilen. Es entsteht ein Baumdiagramm aus allen Teilen, die für das Verfassen des Briefes gebraucht werden (Abbildung 1). Bei größeren

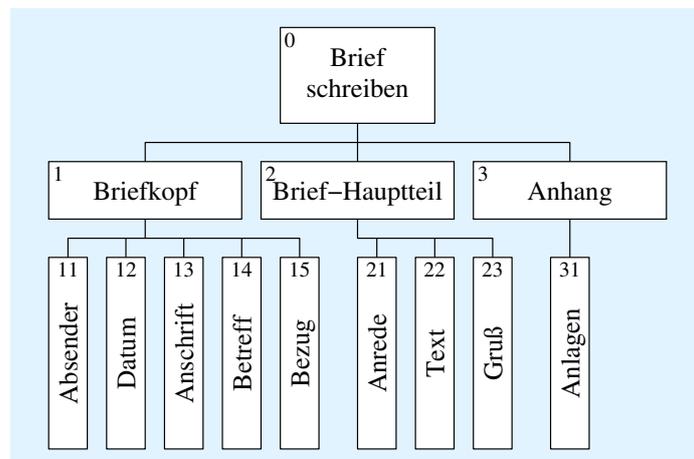


Abbildung 1: Teilaufgaben zu: Schreiben eines Briefes

Projekten können die einzelnen Bausteine wie in der Abbildung nummeriert werden, das verbessert ihre Auffindbarkeit.