Ulrich Kaiser, Christoph Kecher

C/C++

Galileo Press

Auf einen Blick

Inhalt

1 Einige Grundbegriffe

Einem Kochbuch entnehmen wir das folgende Rezept zur Herstellung eines sogenannten Pharisäers:

Zutaten:
1/2 l heißer Kaffee
1/4 l Sahne
2 Essl. Zucker
4 Schnapsgläser 54%iger Rum (8 cl)

Zubereitung:
Den Kaffee aufbrühen und warmhalten. 4 Tassen mit heißem Wasser vorwärmen. Inzwischen die Sahne steif schlagen. Das Wasser aus den Tassen gießen, die Tassen abtrocknen und in jede Tasse 1–2 Teelöffel Zucker geben. Je 1 Schnapsglas Rum darüber gießen und mit dem Kaffee auffüllen. Die Schlagsahne als Haube auf jede Tasse Pharisäer setzen.

Das Rezept gliedert sich in zwei Teile. Im ersten Teil werden die erforderlichen Zutaten genannt, und im zweiten Teil wird ein Verfahren beschrieben, nach dem man aus den Zutaten das gewünschte Getränk herstellen kann. Die beiden Teile sind wesentlich verschieden und gehören doch untrennbar zusammen. Ohne Zutaten ist die Zubereitung nicht durchführbar, und ohne Zubereitung bleiben die Zutaten ungenießbar. Zu beachten ist auch, dass sich der Autor bei der Erstellung des Rezepts einer bestimmten Fachsprache (Essl., cl, Sahne steif schlagen, aufbrühen) bedient. Ohne diese Fachsprache wäre die Anleitung wahrscheinlich weitschweifiger, umständlicher und vielleicht sogar missverständlich. Die Verwendung einer Fachsprache setzt allerdings voraus, dass sich Autor und Leser des Rezepts zuvor (ausgesprochen oder unausgesprochen) auf eine gemeinsame Terminologie verständigt haben.

Wir übertragen dieses Beispiel in unsere Welt – die Welt der Datenverarbeitung:

- Die Zutaten für das Rezept sind die Daten bzw. **Datenstrukturen**, die wir verarbeiten wollen.
- Die Zubereitungsvorschrift ist ein **Algorithmus**[1], der festlegt, wie die Daten zu verarbeiten sind.

1. Dieser Begriff geht zurück auf Abu Jafar Muhammad Ibn Musa Al-Khwarizmi, der als Bibliothekar des Kalifen von Bagdad um 825 ein Rechenbuch verfasste und dessen Name in der lateinischen Übersetzung von 1200 als »Algorithmus« angegeben wurde.

- Das Rezept ist ein **Programm**, das alle Datenstrukturen (Zutaten) und Algorithmen (Zubereitungsvorschriften) zum Lösen der gestellten Aufgabe (Erstellen des Gerichts) enthält.
- Die gemeinsame Terminologie, in der sich Autor und Leser des Rezepts verständigen, ist eine **Programmiersprache**, in der das Programm geschrieben ist. Die Programmiersprache muss dabei in der Lage sein, alle bezüglich der Zutaten und der Zubereitung bedeutsamen Informationen zweifelsfrei zu übermitteln.
- Die Küche ist die technische Infrastruktur zur Umsetzung von Rezepten in schmackhafte Gerichte und ist vergleichbar mit einem **Computer**, seinem **Betriebssystem** und den benötigten **Entwicklungswerkzeugen**.
- Der Koch übersetzt das Rezept in einzelne Arbeitsschritte in der Küche. Üblicherweise geht ein Koch in zwei Schritten vor. Im ersten Schritt bereitet er die Zutaten einzeln und unabhängig voneinander vor (z. B. Kartoffeln kochen), um die Einzelteile dann in einem zweiten Schritt zusammenzufügen und abzuschmecken. In der Datenverarbeitung sprechen wir in diesem Zusammenhang von **Compiler** und **Linker**.
- Das fertige Gericht ist das **lauffähige Programm**, das vom Benutzer (Esser) angewandt (verzehrt) werden kann.

Nur, welche Rolle spielen **wir** in diesem Szenario? Sollte für uns kein Platz vorgesehen sein? Wir suchen uns die interessantesten Aufgaben aus:

- Wir sind Autoren, die sich neue, schmackhafte Gerichte für unterschiedliche Anlässe ausdenken und Rezepte bzw. Kochbücher mit den besten Kreationen veröffentlichen.
- Gelegentlich probieren wir auch einen Pharisäer, um uns an unseren eigenen Schöpfungen zu berauschen und um festzustellen, ob die Speise gelungen ist.

Was müssen wir lernen, um unsere Rolle ausfüllen zu können?

- Wir müssen die Sprache beherrschen, in der Rezepte formuliert werden.
- Wir müssen einen Überblick über die üblicherweise verwendeten Zutaten, deren Eigenschaften und Zubereitungsmöglichkeiten haben.
- Wir müssen einen Vorrat an Standard-Zubereitungsverfahren bzw. Rezepten abrufbereit im Kopf haben.
- Wir müssen wissen, welche Zutaten oder Verfahren miteinander harmonieren und welche nicht.
- Wir müssen wissen, was in einer Küche üblicherweise an Hilfsmitteln vorhanden ist und wie bzw. wozu diese Hilfsmittel verwendet werden.

- Bei anspruchsvolleren Gerichten müssen wir wissen, in welcher Reihenfolge und mit welchem Timing die Einzelteile zuzubereiten sind und wie die einzelnen Aufgaben verteilt werden müssen, damit alles zeitgleich serviert werden kann.
- Wir müssen auch wissen, worauf ein potentieller, späterer Esser Wert legt und worauf nicht. Dies ist besonders wichtig, wenn wir Rezepte für einen ganz besonderen Anlass erstellen.

Letztlich wollen wir komplette Festmenüs und deren Speisefolge komponieren und benötigen dazu eine Mischung aus Phantasie, Kreativität, logischer Strenge, Ausdauer und Fleiß, wie sie auch ein guter Komponist oder Architekt benötigt.

Zurück zu den Grundbegriffen der Informatik. Wir haben informell die Begriffe

- Datenstruktur,
- Algorithmus und
- Programm

eingeführt und dabei bereits erkannt, dass diese Begriffe untrennbar zusammengehören und eigentlich nur unterschiedliche Facetten ein und desselben Themenkomplexes sind.

- Algorithmen arbeiten auf Datenstrukturen. Algorithmen ohne Datenstrukturen sind leere Formalismen.
- Datenstrukturen benötigen Algorithmen, die auf ihnen operieren und sie damit zum »Leben« erwecken.
- Programme realisieren Datenstrukturen und Algorithmen. Algorithmen und Datenstrukturen sind zwar ohne Programme denkbar, aber viele Datenstrukturen und Algorithmen wären ohne Programmierung allenfalls von akademischem Interesse.

In einem ersten Wurf versuchen wir, die Begriffe »Algorithmus«, »Datenstruktur« und »Programm« einigermaßen exakt zu erfassen.

1.1 Algorithmus

Um unsere noch sehr vage Vorstellung von einem Algorithmus zu präzisieren, starten wir mit einer Definition:

> Ein **Algorithmus** ist eine endliche Menge von genau beschriebenen Anweisungen, die unter Benutzung von vorgegebenen Anfangsdaten in einer genau festgelegten Reihenfolge auszuführen sind, um die Lösung eines Problems in endlich vielen Schritten zu ermitteln.

Bei dem Begriff »Algorithmus« denkt man heute sofort an »Programmierung«. Das war nicht immer so. In der Tat gab es Algorithmen schon lange, bevor man auch nur entfernt an Programmierung dachte. Bereits im antiken Griechenland wurden Algorithmen zur Lösung mathematischer Probleme formuliert, so zum Beispiel der Euklidische Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen oder das sogenannte Sieb des Eratosthenes zur Bestimmung aller Primzahlen unterhalb einer vorgegebenen Schranke.[2]

Ihnen ist der Algorithmus zur schrittweisen Berechnung des Quotienten zweier Zahlen schon lange vertraut. Um beispielsweise `84` durch `16` zu dividieren, geht man wie folgt vor:

```
84 : 16 = 5,25
80
 40
 32
  80
  80
   0
```

Wenn wir versuchen, dieses Verfahren im Detail zu beschreiben, finden wir alle in der obigen Definition genannten Elemente wieder:

Problem:

Berechne den Quotienten zweier natürlicher Zahlen!

Anfangsdaten:

`z` = Zähler `(z ≥ 0)`,
`n` = Nenner `(n > 0)` und
`a` = Anzahl der zu berechnenden Nachkommastellen.[3]

Anweisungen:

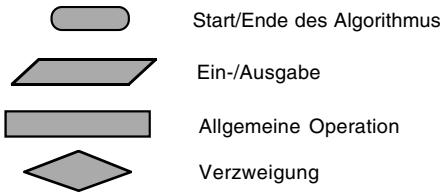
1. Bestimme die größte ganze Zahl `x` mit `nx ≤ z`! Dies ist der Vorkomma-Anteil der gesuchten Zahl.
2. Zur Bestimmung der Nachkommastellen fahre wie folgt fort:
   2.1 Sind noch Nachkommastellen zu berechnen (d.h. `a > 0`)? Wenn nein, dann beende das Verfahren!
   2.2 Setze `z = 10(z-nx)`!
   2.3 Ist `z = 0`, so beende das Verfahren!

2. Eukleides von Alexandria (um 300 vor Chr.) und Erathostenes von Kyrene (um 200 vor Chr.).
3. Anfänglich ist a die Anzahl der zu berechnenden Nachkommastellen. Im Verfahren verwenden wir a als die Anzahl der **noch** zu berechnenden Nachkommastellen. Wir werden den Wert von a in jedem Verfahrensschritt herunterzählen, bis a = 0 ist und keine Nachkommastellen mehr zu berechnen sind.

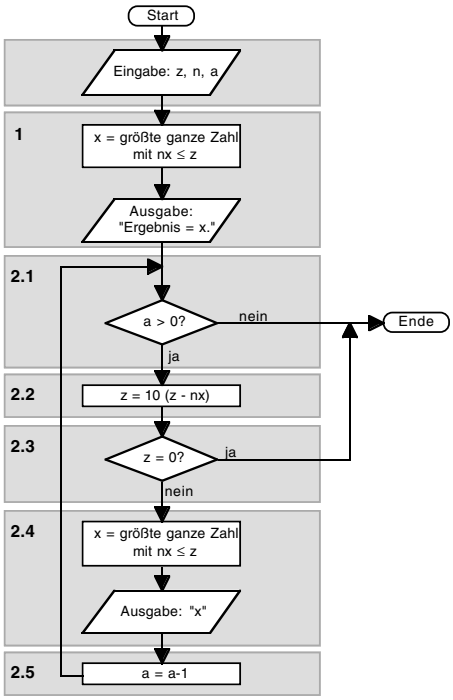
2.4 Bestimme die größte ganze Zahl `x` mit `nx` $\leq$ `z`! Dies ist die nächste Ziffer.

2.5 Jetzt ist eine Ziffer weniger zu bestimmen. Vermindere also den Wert von `a` um `1` und fahre anschließend bei 2.1 fort!

Die einzelnen Anweisungen und ihre Abfolge können wir uns durch ein sogenanntes **Flussdiagramm** veranschaulichen. In einem solchen Diagramm werden alle beim Ablauf des Algorithmus möglicherweise vorkommenden Wege unter Verwendung bestimmter Symbole grafisch beschrieben. Die dabei zulässigen Symbole sind in einer Norm (DIN 66001) festgelegt. Von den zahlreichen in der Norm festgelegten Symbolen wollen wir hier nur einige einführen:



Mit diesen Symbolen können wir den zuvor nur sprachlich beschriebenen Algorithmus auch grafisch darstellen, wenn wir zusätzlich die Abfolge der einzelnen Operationen durch Richtungspfeile kennzeichnen.



Index

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z