

LERNEN LEICHTER GEMACHT



Programmieren lernen

für
dummies[®]



Programmiersprachen
und ihre Unterschiede

Benötigte Werkzeuge:
Compiler, Entwicklungs-
umgebung & Co.

Programmierkonzepte:
Tiefere Einblicke in
Java und PHP

Daniel Lorig

IN DIESEM TEIL ...

Erfahren Sie, wie Computer funktionieren, was Programmierung ist und welche Möglichkeiten es gibt, programmieren zu lernen

Erhalten Sie einen Überblick über verschiedene Programmiersprachen sowie grundlegende Konzepte, die in vielen Sprachen vorhanden sind

Erfahren Sie, was sich heutzutage überhaupt alles (abgesehen von gewöhnlichen Computern) programmieren lässt

Kapitel 1

Von der Idee zum Programm

IN DIESEM KAPITEL

Wie ergänzen sich Mensch und Maschine?

Welche Schritte werden beim Erstellen eines Programms durchlaufen?

Dieses Kapitel skizziert den ganzen Weg, der bei der Erstellung eines Programms beschritten werden muss: von der Idee über den Algorithmus bis hin zur schlussendlichen Ausführung des fertigen Programms.

Mensch vs. Maschine

Computer wurden erschaffen, um Menschen beim Rechnen zu helfen. Heute geht das mit einem handlichen Taschenrechner oder einer App auf dem Smartphone, während früher einfache Rechner schnell einmal ganze Räume ausgefüllt haben.

Mit der Zeit wurden Rechner immer kleiner – und gleichzeitig leistungstärker. Die heutigen modernen Computer unterstützen den Menschen nicht nur bei einfachen Berechnungen, sondern bei so ziemlich allen Problemen des menschlichen Alltags und des Berufslebens:

- ✓ Wie viel Steuern muss man bei einem Brutto-Einkommen von 25.000 Euro zahlen? (Steuerberater)
- ✓ Wie viele Kalorien habe ich bei meinem heutigen Training verbrannt? (Sportler)
- ✓ Wie sieht die aktuelle Tabelle der Fußball-Bundesliga aus? (Fußballfan)
- ✓ Wenn ich ein Haus mit den angegebenen Materialien in der angegebenen Form baue (oder bauen lasse), ist das stabil oder stürzt es ein? (Architekt)
- ✓ Wie komme ich am schnellsten mit dem Auto vom Münchner Hauptbahnhof zum Rathaus von Wladiwostok? (Taxifahrer)

All diese Fragen ließen sich auch ohne die Hilfe einer Maschine klären. Für jede der Fragen kann der Mensch sich genau die Arbeitsschritte überlegen, die zur Beantwortung der Frage nötig wären. Diese müsste der Mensch dann selbst durchführen – was vermutlich

sehr viel langsamer und potenziell auch fehleranfälliger ist, als wenn ein Computer diese Aufgaben durchführt.

Mensch und Maschine haben also sehr unterschiedliche Stärken und Schwächen (siehe [Abbildung 1.1](#)). Während Menschen sehr kreativ sind und neue Lösungswege für Problemstellungen finden können, sind sie auch langsam und fehleranfällig bei der Durchführung. Computer dagegen können gestellte Aufgaben sehr schnell und zuverlässig lösen, führen aber nur exakt die vorgegebenen Schritte aus. Computer »denken« nicht mit, sondern arbeiten streng nach Vorschrift.



Abbildung 1.1: Mensch vs. Maschine

Um das beste Resultat zu erzielen, müssen Mensch und Maschine Hand in Hand arbeiten: Beide müssen ihre eigenen Stärken einbringen, um die Schwächen des jeweils anderen auszugleichen. Die optimale Strategie zur Lösung eines Problems ist demnach wie folgt:

1. Zunächst muss das zu lösende Problem vom Menschen genau spezifiziert werden: Welche Voraussetzungen sind gegeben und was genau soll herausgefunden werden? (*Problemstellung*)
2. Der Mensch findet heraus, welche Schritte zum Lösen des Problems vonnöten sind. (*Algorithmus*)
3. Der Mensch instruiert den Computer, sodass dieser später in der Lage ist, die einzelnen Schritte des Algorithmus eigenständig durchzuführen. (*Programmierung*)
4. Der Computer kann nun jederzeit dazu aufgefordert werden, die zuvor instruierten Schritte durchzuführen, und damit das ursprüngliche Problem lösen. (*Programmausführung*)

Bevor Sie einen Computer dazu verwenden können, ein Problem zu lösen, müssen Sie also erst einmal selbst wissen, wie sich das Problem generell lösen lässt. Das ist eine wichtige Voraussetzung für die Programmierung und in vielen Fällen nicht trivial. Das Finden eines geeigneten Algorithmus kann unter Umständen viel aufwendiger sein als die eigentliche Programmierung selbst.



Auch als Programmierer genügt es nicht, einfach zum Computer zu sagen: »Löse dieses Problem mal für mich.« Das wird den Rechner nicht sonderlich beeindruckend oder motivieren, sondern er wird die Frage schlichtweg ignorieren.

Natürlich könnten Sie einen digitalen Assistenten wie Google Now, Siri, Cortana oder Alexa nutzen und diesen nach der Lösung Ihres Problems fragen (beispielsweise nach der Tabelle der Fußball-Bundesliga). Diese Assistenten sind aber auch nur Programme, die so entworfen wurden, dass sie natürliche Sprache teilweise interpretieren und bestimmte, auf diese Weise formulierte Aufgaben lösen können. Hier hat Ihnen jemand die Schritte 1 bis 3 bereits abgenommen.

In diesem Fall sind Sie nur der Anwender des Programms. Stattdessen möchten Sie aber lernen, wie Sie selbst Programme erstellen können, die derartige Probleme zu lösen in der Lage sind. Oder nicht?

Einen Algorithmus entwickeln

Ein Algorithmus ist ganz allgemein eine eindeutige Handlungsvorschrift zum Lösen eines Problems. Da sehr unterschiedliche Arten von Problemen mit Computern gelöst werden, gibt es keine allgemeingültige Anleitung zum Erstellen eines Algorithmus.

Als Anfänger in der Programmierung werden Sie natürlich zunächst nur eher kleinere und einfachere Probleme lösen, die keine komplizierten Algorithmen benötigen. Dennoch macht es Sinn, sich jetzt schon einmal kurz mit der Erstellung von Algorithmen zu befassen.

Eine Problemstellung und der zugehörige Algorithmus zur Lösung sollten immer möglichst allgemein formuliert werden. Ein Algorithmus, der zwei beliebige Zahlen addieren kann, ist besser als ein Algorithmus, der immer nur $08 + 15$ rechnet.

Ein Algorithmus wird daher in der Regel einen oder mehrere Eingabe-*Parameter* erwarten und eines oder mehrere Ergebnisse liefern. Das Ergebnis ist die sogenannte *Ausgabe*. Der Algorithmus zur Addition von zwei Zahlen würde zum Beispiel die beiden zu addierenden Zahlen als Parameter erhalten und als Ergebnis die Summe liefern. Er kann so mehrmals mit unterschiedlichen Parametern ausgeführt werden, um unterschiedliche Summen zu berechnen.

Im Übrigen brauchen Sie nicht wirklich einen Algorithmus zur Addition von zwei Zahlen zu erstellen. Beim Erstellen eines Algorithmus dürfen Sie davon ausgehen, dass der Computer, der den Algorithmus später ausführen soll, einige gewisse »Grundkenntnisse« bereits mitbringt. Dazu zählen die vier Grundrechenarten: Das Summieren, Subtrahieren, Multiplizieren und Dividieren beherrscht jeder Computer aus dem Effeff.

Damit Sie eine erste Idee davon bekommen, wie ein Algorithmus aussieht, zeige ich Ihnen

hier ein kurzes Beispiel.

Problem: Wie ist die Gesamtsumme einer Reihe von Zahlen, die sich in einer Liste befinden?

Als Beispiel verwende ich hier ein relativ alltägliches Problem. Sie erhalten eine Liste von Zahlen und sollen deren Gesamtsumme bestimmen.

Diesem Problem werden Sie in Ihrem Alltag regelmäßig gegenüberstehen, zum Beispiel wenn Sie beim Kaufen an einer Kasse von der Kassiererin Wechselgeld erhalten oder wenn Sie beim Backen das Gesamtgewicht aller verwendeten Zutaten bestimmen wollen.

Das Problem erhält als Eingabe-Parameter also eine Liste von Zahlen, wobei die Anzahl der enthaltenen Elemente variabel ist. Das bedeutet, jedes Mal, wenn Sie das Problem lösen sollen, handelt es sich um eine unterschiedliche Liste mit einer unterschiedlichen Anzahl an Elementen.

Ich nenne diese Liste im Folgenden L . Sie soll n Elemente besitzen, die mit z_1 bis z_n bezeichnet werden: $L = [z_1, z_2, \dots, z_n]$

Wie würden Sie ein solches Problem manuell lösen, wenn Sie keinen Computer haben, der Ihnen dabei hilft?

- ✓ Eine Idee wäre, sich eine Zwischensumme zu merken und zu dieser nacheinander alle Elemente der Liste hinzuzuaddieren.
- ✓ Zunächst wäre die Zwischensumme 0. Dann würde das erste Element der Liste dazu addiert, danach das zweite und so weiter.
- ✓ Wenn auf diese Weise alle Elemente der Liste durchlaufen wurden, enthält die Zwischensumme die Gesamtsumme aller enthaltenen Elemente, also das gesuchte Endergebnis.

Im Groben ist dies bereits der Algorithmus, mit dem man die Summe von beliebigen Listen berechnen kann. Dieser muss jetzt nur noch etwas ausführlicher aufgeschrieben werden.

Zunächst fällt noch auf, dass für die Lösung des Problems eine Zwischensumme verwendet wurde. Innerhalb eines Algorithmus würde man dafür eine sogenannte *Variable* verwenden.

Eine Variable ist ein Behälter für eine veränderliche Größe, die während der Ausführung eines Algorithmus verwendet wird. In vielen Fällen werden Zahlen in Variablen abgelegt. In Variablen können aber auch Texte stehen. Sie können eine Variable mit einem Wert belegen und diesen Wert jederzeit wieder auslesen oder abändern. Innerhalb eines Algorithmus werden Variablen zum Beispiel verwendet, um Zwischenergebnisse abzuspeichern.

Sie können Variablen beliebige Namen geben, zum Beispiel `arxqt22` oder `hrmpf144mop`.