

Computerprogrammierung für Anfänger

Claus D. Volko

Programmieren heißt, dem Computer zu sagen, was er tun soll. Im Prinzip ist es also nicht viel anders als einen Computer zu bedienen. Aber wenn man einen Computer bedient, benutzt man nur vorgefertigte Anwendungsprogramme. Beim Programmieren arbeitet man in einem Texteditor oder in einer integrierten Entwicklungsumgebung (integrated development environment, IDE). Ein Computerprogramm ist im Prinzip eine Textdatei, die in einer Programmiersprache verfasst ist. Der Computer kann diese Textdatei verarbeiten und folgt den darin enthaltenen Kommandos. Auf diese Weise kann man den Computer dazu veranlassen, Eingaben, zum Beispiel von Tastatur oder Maus, zu lesen, diese zu verarbeiten und die Ergebnisse auszugeben, zum Beispiel in Form von Text oder Grafik.

Einfachste Programmiersprachen wie BASIC sind rein imperativ, das heißt, die Programme bestehen aus einer Reihe von Befehlen, die nacheinander (sequenziell) vom Computer abgearbeitet werden. Ein typisches BASIC-Programm sieht zum Beispiel so aus:

```
10 CLS
20 PRINT "Hallo Welt!"
```

Am Zeilenanfang steht eine Zeilennummer. Die Programmzeilen werden in der Reihenfolge dieser Nummern abgearbeitet. Dieses einfache Programm löscht den Bildschirm und gibt anschließend den Text "Hallo Welt!" aus.

BASIC-Programme können auch konditionale Statements enthalten. Das heißt, man kann Bedingungen auswerten und je nachdem, ob sie erfüllt sind oder nicht, andere Befehle ausführen. Dazu verwendet man in BASIC die IF-Anweisung. Ein Beispiel:

```
10 CLS
20 PRINT "Drücke 1, wenn du 'Hallo Welt!' lesen willst, oder eine andere Taste, wenn
du 'Gute Nacht!' lesen willst."
30 INPUT a$
40 IF a$ = "1" THEN 70
50 PRINT "Gute Nacht!"
60 END
70 PRINT "Hallo Welt!"
```

Die ursprünglichen BASIC-Dialekte unterstützen nur Sprünge zu bestimmten Programmzeilen. Dadurch entsteht im Extremfall das, was im Jargon der Programmierer als "Spaghetticode" bezeichnet wird: ein sehr unübersichtliches Programm mit vielen Verzweigungen, bei dem man sich nicht auskennt, wo welcher Programmteil genau beginnt. Deshalb hat man in späteren BASIC-Dialekten sowie in anderen Programmiersprachen wie zum Beispiel Pascal Elemente der strukturierten Programmierung eingebaut. Es gibt nicht nur Sprünge, sondern auch Kontrollstrukturen, durch die das Programm besser gegliedert wird und übersichtlicher bleibt. Das obige Programm könnte zum Beispiel in QBasic folgendermaßen umgeschrieben werden:

```
CLS
PRINT "Drücke 1, wenn du 'Hallo Welt!' lesen willst, oder eine andere Taste, wenn du
'Gute Nacht!' lesen willst."
INPUT a$
IF a$ = "1" THEN
    PRINT "Hallo Welt!"
ELSE
```

```
        PRINT "Gute Nacht!"
    END IF
```

In fortgeschrittenen Programmiersprachen wie C sind Programme anders aufgebaut. In C bestehen Programme aus einzelnen Funktionen. Eine Funktion ist im Prinzip ein abgekapselter Programmteil, der einen Wert zurückgibt. Es gibt auch Funktionen vom Datentyp void, die keinen Wert zurückgeben – solche Funktionen werden in anderen Programmiersprachen auch als Prozeduren bezeichnet.

Die Hauptfunktion heißt in C main. Diese Funktion wird als erste ausgeführt.

Hier ein Beispiel in der Programmiersprache C:

```
#include <stdio.h>
#include <string.h>

void GuteNacht()
{
    printf("Gute Nacht!");
}

void HalloWelt()
{
    printf("Hallo Welt!");
}

char Entscheidung()
{
    char input;
    printf("Drücke 1, wenn du 'Hallo Welt!' lesen willst, oder eine andere Taste,
wenn du 'Gute Nacht!' lesen willst.");
    input = getchar();
    return input;
}

void main()
{
    if (Entscheidung() == '1')
    {
        HalloWelt();
    }
    else
    {
        GuteNacht();
    }
}
```

In den Programmiersprachen, die heutzutage tatsächlich professionell eingesetzt werden, geht es noch einen Schritt weiter. Diese Programmiersprachen sind objektorientiert. Das bedeutet, dass jede Funktion einer Klasse zugeordnet ist. Man bezeichnet diese Funktionen dann auch als Methoden. Von jeder Klasse lassen sich Objekte ableiten, auf die man dann im Programm zugreifen kann.

Im Prinzip ähnelt eine Klasse dem, was in älteren Programmiersprachen als Struktur bezeichnet wurde. Eine Struktur ist ein benutzerdefinierter Datentyp. Ein einfaches Beispiel:

Die Struktur "Kunde" kann aus dem Namen, dem Geburtsdatum und der Wohnadresse einer Person bestehen. Eine Variable, die von diesem Typ ist, kann zum Beispiel "Claus Volko, 8. Oktober 1983, Hungereckstraße" lauten.

Eine Klasse ist wie eine Struktur, aber erweitert darum, dass sie auch eigene Funktionen (Methoden) haben kann. Die Klasse "Kunde" könnte zum Beispiel eine Methode enthalten, mit der die personenbezogenen Daten auf dem Bildschirm ausgegeben werden.

Ein einfaches Programm in der objektorientierten Programmiersprache C# lautet:

```
Kunde claus = new Kunde();
claus.Name = "Claus Volko";
claus.Geburtsdatum = "8. Oktober 1983";
claus.Wohnadresse = "Hungereckstraße";

claus.AusgabeDaten();

public class Kunde
{
    public string Name;
    public string Geburtsdatum;
    public string Wohnadresse;

    public void AusgabeDaten()
    {
        Console.WriteLine("Name: " + Name);
        Console.WriteLine("Geburtsdatum: " + Geburtsdatum);
        Console.WriteLine("Wohnadresse: " + Wohnadresse);
    }
}
```

Wer professionell programmieren möchte, muss das objektorientierte Paradigma beherrschen, also wissen, wie man in einer objektorientierten Programmiersprache Programme schreibt. Zu dieser Gattung von Programmiersprachen gehören zum Beispiel C#, C++ oder auch Java.

Wer nur hobbymäßig programmieren möchte, kann auch mit einer einfacheren Programmiersprache anfangen. Für die Programmiersprache QBasic habe ich einen Kurs geschrieben, der für Anfänger sehr gut geeignet ist. Er kann unter dieser Internetadresse erreicht werden:

http://www.googlemeier.de/BASIC_Kurs_VOLKO.pdf

Claus D. Volko, Wien, im Juni 2023