Einführung Programmiersprachen

16. September 2012

Inhaltsverzeichnis

1 Was ist Programmierung

Einfach erklärt: Man schreibt dem Computer auf was er tun soll und dieser führt dann die Befehle aus.

Ausführlicher erklärt: Grundsätzlich unterscheidet man zwischen Systemprogrammierung und Anwenderprogrammierung.

Systemprogramme sind die Programme, die für den Betrieb des Computers benötigt werden, wie z.B. das Betriebssystem.

Mit Anwenderprogrammen lassen sich vielfältige Probleme mit Hilfe eines Computers lösen, wie z.B. Briefe erstellen, Bilder bearbeiten oder im Internet surfen. Um programmieren zu können, benötigt man eine Programmiersprache, eine Schnittstelle zur Maschinensprache und natürlich einen Computer, auf dem das Programm laufen kann.

Eine Programmiersprache dient der Formulierung der exakten Beschreibung eines Algorithmus, in einer dem Computer indirekt verständlichen Weise, so dass er diesen verarbeiten kann. Ein Algorithmus ist eine Folge eindeutiger und ausführbarer Anweisungen. Aus zulässigen Eingabedaten werden dabei nach endlich vielen Verarbeitungsschritten bestimmte Ausgabedaten bereitgestellt.

Algorithmen sind der Grundbaustein eines jeden Programms.

Das Ziel der Programmierung sind Programme, die dem Anwender und Nutzer die Arbeit erleichtern, Schritte zu automatisieren oder bestimmte Aufgaben ganz abzunehmen. Die Programmierung ist beendet, wenn das Programm fehlerfrei läuft. Softwareentwicklung dagegen ist weitaus mehr als Programmierung. Heutzutage gehört dazu nicht nur, das Schreiben eines Programms, was die gewünschten Funktionen und Arbeiten erfüllt, sondern auch das Schreiben einer Dokumentation, das Reagieren auf eventuell auftauchende Fehler und Probleme beim Anwender ("Bugs") und schließlich auch die Weiterentwicklung des Programms.

2 Die Entwicklung der heutigen Programmiersprachen

Programmiersprachen formulieren Algorithmen und Datenstrukturen so, dass der Rechner das resultierende und übersetzte (kompilierte) Programm ausführen kann. Es gibt heute bereits mehr als tausend Programmiersprachen, wobei allerdings nur etwa 20 weiter verbreitet sind.

`http://en.wikipedia.org/wiki/Hello_world_program_examples`
`http://www.lob.de/pdf/helloworld.pdf`
`http://99-bottles-of-beer.net/`
`http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`

Es gibt viele Möglichkeiten Programmiersprachen einzuteilen. So läßt sich eine Programmiersprache zum Beispiel nach ihrer Maschinennähe, nach ihrer Generation, nach ihrer Problembezogenheit oder nach ihrer konzeptionellen Denkweise (Programmierparadigma) einteilen. Nicht alle Programmiersprachen lassen sich eindeutig einer dieser Klassen zuordnen.

Für einen ersten Überblick eignet sich die Einteilung nach der Maschinennähe einer Programmiersprache. Die kommenden Beispiele zeigen eindrucksvoll die Unterschiede. Man unterscheidet heute 3 Hauptgruppen von Programmiersprachen in unterschiedlicher Höhe (Entfernung) vom Computer:

1. **Maschinensprachen**
2. **Assemblersprachen**
3. **Problemorientierte (höhere) Programmiersprachen**

In der Anfangszeit der Mikrorechentechnik wurden Programme fast ausschließlich in Maschinen- oder Assemblercode erstellt und getestet. Nach dem Erscheinen der ersten Betriebssysteme und verbesserter Hilfsmittel für die Programmentwicklung, vor allem von Übersetzern (Compilern) für höhere Programmiersprachen, waren Ende der siebziger Jahre die Voraussetzungen für eine Programmentwicklung geschaffen, die dem Niveau der Rechentechnik entsprach. Die Orientierung auf wenige international verbreitete Betriebssysteme förderte die Kompatibilität und damit den Austausch von Programmen.

3 Übersicht der Programmiersprachen

3.1 Maschinensprachen

Der Maschinencode (Maschinensprache) ist das Endergebnis der Kompilierung einer höheren Programmiersprache wie z.B. C oder Assembler. Dieser aus Einsen und Nullen (Binärzeichen) bestehende Code ist die einzige Sprache, die ein Computer verstehen kann. Er bildet somit die Grundlage aller Programmiersprachen, welche nur als Hilfsmittel dienen, dass der Mensch in einer einfachen Weise die vom Computer zu lösenden Probleme formulieren kann. Alle verwendeten mathematischen Operationen werden auf eine sehr begrenzte Anzahl von Grundoperationen zurückgeführt. Die Programme sind an ein bestimmtes Rechnermodell gebunden und nicht auf andere Plattformen portierbar. Auch die benutzbaren Ein- und Ausgabegeräte sind vorgegeben.

In Maschinensprache geschriebene Programme sind äußerst schwer lesbar und auch für den erfahrenen Programmierer oft unverständlich. Sie bieten dafür den Vorteil, dass sie schneller verarbeitet werden können, weil eine spezifischere Problemformulierung möglich ist, da man nicht auf vorgegebene und umfangreiche Formulierungshilfenßurückgreifen muss.

Listing 1: Hello World in Maschinencode stark verkürzt

```
1 B4 09 8D 16 0D 01 CD 21 B8 00 4c CD 21 48 65 EF
2 6C 45 F2 00 1C C1 FF 32 48 A2 EF 56 FF F0 F4 00
```

3.2 Assemblersprachen

Assemblersprachen sind symbolische Darstellungen der Maschinencodes und damit ebenso maschinennah und schnell zu verarbeiten. Dadurch sind sie sehr effizient und außerdem benötigen sie nur wenig Speicherplatz. Der Nachteil ist allerdings, dass sie, genau wie die Maschinensprachen, an einen bestimmten Rechnertyp gebunden sind.

Auch Assemblersprachen sind noch recht schwer verständlich und wenig übersichtlich, erleichtern aber immerhin die Programmierung in Maschinensprache (rein binäre Darstellung).

Listing 2: Hello World in Assembler x86 Windows 32-bit

```
1  ; This program displays "Hello, World!" in a windows messagebox
2  ; and then quits.
3          .486p
4          .model  flat,STDCALL
5  include  win32.inc
6
7  extrn           MessageBoxA:PROC
8  extrn           ExitProcess:PROC
9
10 .data
11 HelloWorld db "Hello, World!",0
12 msgTitle db "Hello world program",0
13
14 .code
15 Start:
16         push    MB_ICONQUESTION + MB_APPLMODAL + MB_OK
17         push    offset msgTitle
18         push    offset HelloWorld
19         push    0
20         call    MessageBoxA
21         push 0
22         call ExitProcess
23 ends
24 end Start
```

3.3 Höhere Programmiersprachen (High Level Language)

Höhere Programmiersprachen sind die heute am meisten verwendeten Programmiersprachen. Sie basieren auf der Maschinensprache und werden durch einen Compiler oder Interpreter in diese übersetzt (compiliert).

Programme in einer höheren Programmiersprache sind oft plattform- und rechnerunabhängig und können übertragen werden. Sie haben einen klaren und strukturierten Aufbau und sind relativ leicht nachvollziehbar. Die höheren Programmiersprachen ermöglichen eine problemnahe und für den Anwender transparente Darstellung, die den Forderungen nach Anwenderfreundlichkeit und Verständlichkeit entgegenkommt. Da dem Computer an sich die Programme unverständlich sind, müssen sie immer erst compiliert werden. Sie belegen mehr Speicherplatz und sind in der Ausführung langsamer als vergleichbare reine Maschinencodes.

Listing 3: Hello World in c

```c
// this programm writes hello world to the commandline

#include <stdio.h>

int main(){
        printf("hello world\n");
        return 0;
}
```

Listing 4: Hello World in python

```python
// this programm writes hello world to the commandline

print "hello world"
```

4 Quellen

Verwendte Links:

http://www.bernd-leitenberger.de/entwicklung-der-programmiersprachen.shtml
http://en.wikipedia.org/wiki/Hello_world_program_examples
http://www.lob.de/pdf/helloworld.pdf
http://99-bottles-of-beer.net/
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

5 Fragen zum Text

1. Nach welchem Kriterium haben wir Programmiersprachen eingeteilt.
2. Welche Haupgruppen haben wir dabei unterschieden und worin liegt der Unterschied zwischen diesen Gruppen?
3. Was ist der Unterschied zwischen einer kompilierten Sprache wie C oder C++ und einer interpretierten Sprache wie Java oder Python?
4. Was macht ein Compiler?