

Einstieg in die Programmierung von Computern – Teil I

Mit der neuen mehrteiligen Serie „Einstieg in die Programmierung von Computern“ soll ein fundamentales Verständnis für die Funktionsweise der computerisierten Welt geschaffen werden. Eine moderne und leicht verständliche Programmiersprache wie z. B. „Java“ zu erlernen, ist dafür bestens geeignet. Mit dem ersten Teil gibt Autor Thomas Burgard einen Einstieg in die komplexe Welt der Computerprogrammierung mit Java.

Wofür wird eine Programmiersprache benötigt?

Das Wort „Programmiersprache“ ist aus den beiden Wörtern „Programm“ und „Sprache“ zusammengesetzt und drückt damit aus, dass ein Programm in einer bestimmten Sprache geschrieben oder, besser gesagt, „programmiert“ wird. Das Wort „Programm“ wiederum drückt aus, dass sogenannte „Algorithmen“ den Ablauf bestimmen. Algorithmen kommen aus der Mathematik bzw. Informatik und stellen Lösungen bzw. Vorschriften für Lösungen von mathematischen bzw. informationstechnischen Problemen dar. Man kann sich einen Algorithmus auch als eine Art Kochrezept oder Anleitung für eine bestimmte Sache vorstellen. Ein Algorithmus in der Informatik besitzt also folgende Eigenschaften (Quelle: „Wikipedia“):

- Ein Algorithmus bearbeitet eine bestimmte Menge von definierten „Objekten“ (z. B. ein Kunden-Objekt).
- Ein Algorithmus führt mit den Objekten bestimmte Operationen durch (z. B. Speichern der Kunden-Objekte in eine Datenbank, Entfernen von Kunden-Objekten aus der Datenbank usw.).
- Die Objekte besitzen einen wohl definierten Anfangszustand und Endzustand.
- Ein Algorithmus muss in einem endlichen Text eindeutig beschreibbar sein (Finitheit).
- Jeder Schritt des Algorithmus muss tatsächlich ausführbar sein (Ausführbarkeit).
- Ein Algorithmus darf zu jedem Zeitpunkt nur endlich viel Speicherplatz benötigen (Dynamische Finitheit).
- Ein Algorithmus darf nur endlich viele Schritte benötigen (Terminierung).

ANZEIGE



BLACK LASER-SINTERN

ENTSCHEIDUNG FÜR QUALITÄT UND SERVICE: ■ Legierungen ■ Galvanotechnik ■ Discs / Fräser ■ Lasersintern

■ Experten für CAD/CAM und 3Shape

+49 (0) 40 / 86 07 66 · www.flussfisch-dental.de

since 1911
FLUSSFISCH

Darüber hinaus wird der Begriff Algorithmus in praktischen Bereichen oft auf die folgenden Eigenschaften eingeschränkt:

- Der Algorithmus muss bei denselben Voraussetzungen das gleiche Ergebnis liefern (Determiniertheit).
- Die nächste anzuwendende Regel im Verfahren ist zu jedem Zeitpunkt eindeutig definiert (Determinismus).

Es gilt: Ein Algorithmus besteht aus einzelnen und sequenziell (nacheinander) ausführbaren Anweisungen, die der Mikroprozessor in einer endlichen Zeit abarbeiten kann.

Mit einer Programmiersprache ein Programm schreiben

Ein Programm (oft auch als Software bezeichnet) besteht demnach aus mehreren Anweisungen, die zusammengefasst einen

bestimmten Algorithmus darstellen und beschreibt, wie ein Problem zu lösen ist. Die Anweisungen müssen ganz genau mittels Wortschatz und Grammatikregeln (Syntax) festgelegt werden. Jede Programmiersprache besitzt seinen eigenen Wortschatz und Sprachsyntax, die ein Mensch lernen muss, um damit ein ausführbares Programm zu schreiben.

Es gilt: Programmiersprachen sind abstrakte Sprachen, mit denen Programmieranweisungen (Algorithmus) geschrieben werden können, die ein Mikroprozessor dann sequenziell abarbeiten kann. Die Anweisungen folgen einem bestimmten Wortschatz (Anweisungen, Funktionen, ...) und einer bestimmten Sprachsyntax (Syntax dieser Anweisungen, Funktionen, ...), die jede Programmiersprache selbst festlegt.

Was bedeutet „höhere Programmiersprache“?

Eine höhere Programmiersprache ist eine abstrakte Sprache, mit der unabhängig von einem Computersystem Programme geschrieben werden können. Der Programmierer soll sich nicht um die Details eines Computersystems kümmern, sondern nur um die Problemlösung auf einer abstrakten Ebene.

Was bedeutet „Maschinen- bzw. Assemblersprache“?

Im Gegensatz zu höheren Programmiersprachen gibt es außerdem die „Maschinensprachen“ (auch als Assemblersprachen bezeichnet), bei denen der Programmierer den internen Auf-

bau der Mikroprozessorarchitektur genau kennen muss. Assemblersprachen sind Programmiersprachen, die die Maschinensprache einer Mikroprozessorarchitektur in eine für den Menschen lesbare Form abstrahieren. Jede Mikroprozessorarchitektur hat somit eine eigene Assemblersprache.

Der Compiler

Die mit einer Programmiersprache geschriebenen Programme müssen vor der Ausführung in einem Mikroprozessor „com-

piert“ werden. Der Compiler

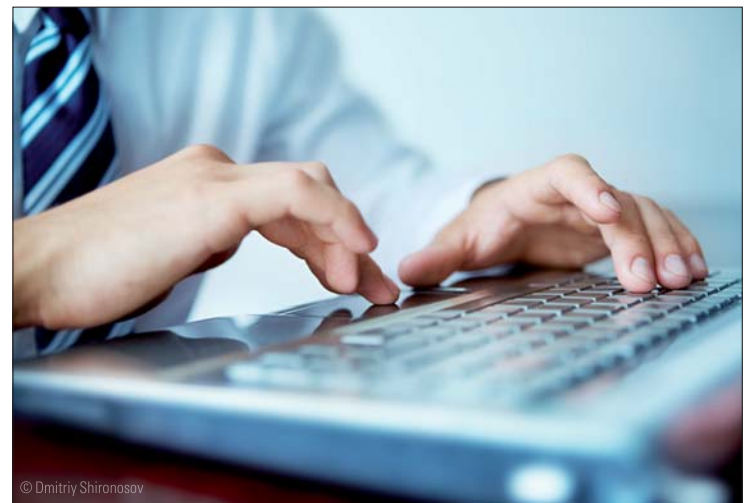
compiliert die Anweisungen in einen sogenannten „Maschinencode“, der dann direkt vom Prozessor verstanden und ausgeführt werden kann.

Es gilt: Programm → Compiler → Ausführung

Da wir nun wissen, wofür Programmiersprachen verwendet werden, können wir nun schauen, welche Arten von höheren Programmiersprachen eingesetzt werden (ich werde hier jedoch nur die zwei interessantesten

Arten von Programmiersprachen beschreiben):

1. *Imperative (prozedurale) Programmiersprachen:* Hier bestehen die Programme aus einer Aneinanderreihung von Anweisungen bzw. Operationen, die bestimmte Daten verbreiten. Das Hauptaugenmerk ist das „Variablenkonzept“, nach dem Eingabewerte in Variablen (Speicherzellen) abgelegt und dann weiterverarbeitet werden. Beispiele: BASIC, C, FORTRAN, PASCAL, COBOL, ADA
2. *Objektorientierte Programmiersprachen:* Objektorien-



tierte Programmiersprachen sind moderne und leistungsfähige Sprachen und behandeln die Informationen als sogenannte „Objekte“ (Objektorientierung). Objekte sind wie im richtigen Leben aktiv handelnde Einheiten, die miteinander kommunizieren können (ein Objekt ruft z. B. eine Nachricht eines anderen Objektes auf). Objekte stellen Eigenschaften (Attribute) und Operationen (Methoden) zur Verfügung, dabei charakteri-

sierte Programmiersprachen sind moderne und leistungsfähige Sprachen und behandeln die Informationen als sogenannte „Objekte“ (Objektorientierung). Objekte sind wie im richtigen Leben aktiv handelnde Einheiten, die miteinander kommunizieren können (ein Objekt ruft z. B. eine Nachricht eines anderen Objektes auf). Objekte stellen Eigenschaften (Attribute) und Operationen (Methoden) zur Verfügung, dabei charakteri-

ANZEIGE

Dubliersilikone



LOGOSIL FL16/18 N türkis
Hochwertiges additionsvernetzendes Dubliersilikon mit niedriger Endhärte **Shore A16/18**.

Nur für Kuvettendublierung empfohlen.

- Hohe Zeichnungsgenauigkeit
- Modelle leicht entformbar durch geringe Endhärte
- Für Dosierautomaten geeignet

N = normalhärtend (30/45 Min.)

LOGOSIL FL24 N rosé

Hochwertiges additionsvernetzendes Dubliersilikon mit allen Voraussetzungen für die Klebeband- und Kuvettentechnik.

Endhärte **Shore A24**

- Einfach und sparsam zu dosieren
- Gute dünnfließende Konsistenz
- Für Dosierautomaten geeignet

N = normalhärtend (30/45 Min.)



Das „schnellste“ unter den Dubliersilikon

da nach 10 Min. abformbar!

LOGOSIL FL24 S gelb

Ein additionsvernetzendes und schnell ausvulkanisierbares Dubliersilikon, speziell für eilige Arbeiten in der Klebeband- und Kuvetten-Technik entwickelt.

Endhärte **Shore A24**

- Bereits nach 10 Minuten entformbar
- Hohe Reißfestigkeit und weiter Reißwiderstand
- Gute dünnfließende Konsistenz
- Für Dosierautomaten geeignet

S = schnellhärtend, bereits nach 10 Min. abformbar

Daueraktion LOGOSIL

3 Einheiten nach Wahl bestellen + 1 Einheit kostenlos (33% gespart)



Zum Testen:
LOGOSIL FL16/18 N türkis
LOGOSIL FL24 N rosé
LOGOSIL FL24 S gelb

2 x 1 Kg Komp. A+B nach Wahl

Sonderpreis 49,90 €

Bei Nachkauf von 2 x 6 Kg wird der

Sonderpreis gutgeschrieben. Aktion gültig bis 31.12.2012

Je 2 x 6 Kg Komp. A+B

LOGOSIL FL16/18 N türkis 268,- €
LOGOSIL FL24 N rosé 299,- €
LOGOSIL FL24 S gelb 308,- €

zzgl. MwSt. • Preisänderungen vorbehalten

sieren die Attribute das Objekt und die Methoden stellen die Kommunikationsschnittstelle nach außen dar. Beispiele: JAVA, C++, PHP, SMALL-TALK, EIFEL

In unserer Serie wird die Programmiersprache Java verwendet!

Warum Java?

Java ist eine moderne und sehr leistungsfähige Programmiersprache, die den objektorientierten Ansatz optimal umgesetzt hat (eine rein objektorientierte Sprache) und zudem in vielen Bereichen und Anwendungen eingesetzt wird. Was „Objekte“ bzw. „Objektorientierung“ genau bedeutet, wird später in der Serie erklärt.



Die Vorteile und Eigenschaften von Java:

- Java ist eine rein objektorientierte Programmiersprache.
- Java ist sehr leicht zu erlernen und eignet sich deshalb optimal für Anfänger.
- Die in Java entwickelten Programme laufen sehr stabil.
- Java ist „Open Source“ und somit kostenfrei.
- Java-Programme können auf allen gängigen Plattformen (Windows, Mac, Linux, Unix, ...) ablaufen. Der Java-Quellcode wird in einen Bytecode umgewandelt, der von der JVM (Java Virtual Machine) interpretiert wird. Daher muss nur die JVM für die jeweilige Plattform bzw. Betriebssystem angepasst sein. Die JVM ist ein Bestandteil der Java-Laufzeitumgebung (engl. JRE Java Runtime Environment). Java ist also „plattformunabhängig“!
- Java unterstützt die Programmierung von verteilten Systemen und Netzen.
- Java unterstützt die parallele Programmierung, die auch als Nebenläufigkeit oder Multi-Threading bezeichnet wird. Multi-Threading bedeutet, dass mehrere Programmvorgänge gleichzeitig (also parallel) ausgeführt werden können. Diese Eigenschaft wird nicht von vielen anderen Programmiersprachen unterstützt.
- Java-Programme werden erst zur Laufzeit übersetzt und an-

schließend ausgeführt (interpretiert). Java ist somit eine Interpretersprache.

- Mit Java können sichere und robuste Programme entwickelt werden, da sie keine „Zeiger“ (ein Zeiger ist ein direkter Verweis auf eine Speicherstelle im Hauptspeicher des Rechners) wie in C++ besitzen. Man kann sich ja leicht vorstellen, dass Zeiger zu großen Problemen führen können, wenn diese auf nicht versorgte Speicherbereiche referenzieren.
- Java ist durch eine sehr umfangreiche Klassenbibliothek (wird später in der Serie erklärt) optimal für die Client-Server-Programmierung (z.B. in der Webprogrammierung) geeignet.
- Für Java existieren sehr viele Zusatzprodukte und Programmier-Werkzeuge, dabei sind viele Produkte „Open Source“ und kostenfrei.
- Java ist zukunftssicher, da weltweit viele Unternehmen und Programmierer diese Sprache einsetzen. Außerdem wird Java von dem Unternehmen „Oracle“ ständig weiterentwickelt und kann eine sehr große „Community“ vorweisen, die äußerst aktiv ist.

Die Java-Plattform

Java bildet zusammen mit der umfangreichen Klassenbibliothek, der JRE und verschiedenen Werkzeugen die sogenannte „Java-Plattform“. Die Java-Plattform gibt es in den folgenden Ausprägungen:

- als Java Standard Edition **Java SE**. Für den generellen Einsatz auf PCs, Servern oder ähnlichen Geräten.
- als Java Enterprise Edition **Java EE** (baut auf Java SE auf). Für Webanwendungen geeignet.
- und als Java Micro Edition **Java ME** (baut auf Java SE auf). Für mobile Endgeräte geeignet.

In unserer Serie werden wir uns aber nur mit der Java SE beschäftigen!

Das Java Development Kit (JDK)

Mit dem Java Development Kit (kurz JDK) als Teil der Java-Plattform kann der Programmierer Java SE Applikationen programmieren und bekommt für die Entwicklung verschiedene Werkzeuge und Hilfsprogramme wie z. B. der Java-Compiler „javac“ in die Hand.

Das Java Runtime Environment (JRE)

Die Java Runtime Environment (JRE), also der Java Laufzeitumgebung, bündelt lediglich die Java Virtuelle Maschine (JVM) mit den notwendigen Klassenbibliotheken. Die JRE wird für

die Ausführung eines Java-Programmes benötigt.

Der Weg von der Programmerzeugung zur Ausführung

Prinzipiell besitzen alle Java Quellcode-Dateien die Dateiendung .java. Quellcode-Dateien sind diejenigen Dateien, die der Programmierer mit einem Texteditor schreibt. Die Quellcode-dateien werden anschließend mit dem Java-Compiler in sogenannte Bytecode-Dateien kompiliert. Die vom Java-Compiler erzeugten Bytecode-Dateien besitzen die Dateiendung .class. Diese können von der JRE auf einem Zielrechner interpretiert und ausgeführt werden. Um das Java-Programm zu starten, muss eine Bytecode-Datei eine startbare Klasse (wird später in der Serie erklärt) sein. Die JVM setzt dabei den Bytecode in den Maschinencode des jeweiligen Prozessors auf dem Zielrechner um. Im Folgenden nochmals die Reihenfolge von der Programmerzeugung bis zum Programmstart:

1. Java Quellcode-Datei(en) mit der Dateiendung .java mit einem Texteditor schreiben.
2. Java Quellcode-Datei(en) in Bytecode compilieren. Zu jeder Java Quellcode-Datei wird eine Bytecode-Datei mit der Dateiendung .class vom Compiler generiert. Dabei muss eine startbare Klasse vorhanden sein.
3. Das compilierte Java-Programm (sind in der Regel mehrere Java Bytecode-Dateien) über die startbare Klasse mit der JRE zur Ausführung bringen.

Ausblick zum zweiten Teil der Serie

Im nächsten Teil II wollen wir die Voraussetzungen auf dem Computer zum Programmieren in Java schaffen. Wir schreiben das erste kleine Programm und lernen die grundlegenden Werkzeuge für die Java-Programmierung kennen. Wir lernen die ersten einfachen Datentypen und Anweisungen in Java kennen, erzeugen eine „startbare Klasse“ und führen anschließend das selbst geschriebene Java-Programm aus. **ZT**

ZT Autor



Thomas Burgard entwickelt Applikationssoftware und professionelle Internetauftritte für Unternehmen.

ZT Adresse

Thomas Burgard Dipl.-Ing. (FH)
Softwareentwicklung & Webdesign
Bavariastraße 18b
80336 München
Tel.: 089 540707-10
Fax: 089 540707-11
info@burgardsoft.de
www.burgardsoft.de
burgardsoft.blogspot.com
twitter.com/burgardsoft

ANZEIGE

Hedent

Hedent Inkosteam

Platzsparend durch Wandhalterung, Arbeitsflächen bleiben frei.

Dampföuse am Handstück auswechselbar. Dampföusen 1 mm, 3 mm zusätzlich erhältlich.

Durch direkten Anschluss an die Wasserleitung entfällt das lästige Nachfüllen von Hand. Der Dampfstrahl bleibt durch die vollautomatische Niveauregulierung konstant.

Inkoquell 6 ist ein Wasseraufbereitungsgerät zur Versorgung von Geräten mit kalkfreiem Wasser bei automatischer Kesselfüllung. Einfache Installation. Das Gerät verhindert durch Teildemineralisierung des Leitungswassers Kalkablagerungen im Dampfstrahlgerät. Robuste Bauweise aus Edelstahl sichert eine einwandfreie Funktion des Gerätes über einen langen Zeitraum. Besonders wirtschaftlich. Einfache Regenerierung durch den Anwender.

Inkosteam Ein leistungsstarkes Hochdruckdampfstrahlgerät für den täglichen Einsatz im Labor und allen Arbeitsbereichen, wo hartnäckiger Schmutz auf kleinstem Raum zu entfernen ist. Das Inkosteam arbeitet mit einer Dampfstufe. Das **Inkosteam II** arbeitet mit zwei Dampfstufen. Normaldampf und Nassdampf. Der Schmutz wird nicht nur gelöst, sondern fließt durch die hohe Spülwirkung gut ab. Durch den zweistufigen Fußschalter lässt sich der Dampf jederzeit individuell anpassen. Keine umständlichen Schalterfunktionen. Kessel und Heizung sind aus hochwertigem Edelstahl gefertigt. Hohe Zuverlässigkeit und Betriebssicherheit.

Inkoquell 6 Wassereenthärter
Gerät schützt Ihr Dampfstrahlgerät vor Verkalkung.

Hedent GmbH

Obere Zeil 6 – 8
D-61440 Oberursel/Taunus
Germany

Telefon 06171-52036
Telefax 06171-52090

info@hedent.de
www.hedent.de

Weitere Produkte und Informationen finden Sie auf unserer Homepage!