

C

Hans-Georg
Schumann

FÜR **KIDS**



Auf der CD:

Alle Beispieldateien sowie
eine vollständige
Entwicklungsumgebung
für Windows und Linux

bhv

mit 
CD



Vorwort

Es sind schon wahre Wunderwerke, die Computer! Man trifft diese Dinge überall, sie können allerhand Sachen, man kann damit Texte schreiben, Bilder malen und spielen.

In Wirklichkeit aber sind sie stohdumm. Denn sie machen nur das, was ihnen ein Programm sagt. Ist das ein Grafikprogramm, dann lassen sich mit dem Computer Bilder bearbeiten. Ist das ein Actionspiel, dann lässt sich der Computer mit einem Joystick oder den Tasten steuern und bewegt irgendwelche Objekte über den Bildschirm.

Programme werden von Leuten erstellt, die sich in einer Programmiersprache auskennen, sich Nächte um die Ohren schlagen und nur in der Nähe ihres Computers schlafen. So sagt man, aber das muss ja nicht stimmen.

Was heißt denn Programmieren?

Wenn du aufschreibst, was ein Computer tun soll, nennt man das *Programmieren*. Das Tolle daran ist, dass du selbst bestimmen kannst, was getan werden soll. Lässt du dein Programm laufen, macht der Computer die Sachen, die du ausgeheckt hast. Natürlich wird er dann dein Zimmer nicht aufräumen und dir auch keine Tasse Kakao ans Bett bringen. Aber kannst du erst mal programmieren, kannst du den Computer sozusagen nach deiner Pfeife tanzen lassen.

Allerdings passiert es gerade beim Programmieren, dass der Computer nicht so will, wie du es gerne hättest. Meistens ist das ein Fehler im Programm. Er kann aber auch irgendwo anders im Computer oder im Betriebssystem liegen. Das Dumme bei Fehlern ist, dass sie sich gern so gut verstecken, dass die Suche danach schon manchen Programmierer zur Verzweiflung gebracht hat.

Vielleicht hast du nun trotzdem Lust bekommen. Dann brauchst du ja nur noch eine passende *Entwicklungsumgebung* und schon kann's losgehen.



Was ist eine Entwicklungsumgebung?

Um ein Programm zu erstellen, musst du erst mal etwas eintippen. Das ist wie bei einem Brief oder einer Geschichte, die man schreibt. Das Textprogramm dafür kann sehr einfach sein, weil es ja nicht auf eine besondere Schrift oder Darstellung ankommt. So was wird *Editor* genannt.

Ist das Programm eingetippt, kann es der Computer nicht einfach lesen und ausführen. Jetzt muss es so übersetzt werden, dass der PC versteht, was du von ihm willst. Weil er aber eine ganz andere Sprache spricht als du, muss ein Dolmetscher her.

Du programmierst in einer Sprache, die du verstehst, und der Dolmetscher übersetzt es so, dass es dem Computer verständlich wird. So was heißt dann *Compiler* (ausgesprochen: Kompailer).

Schließlich müssen Programme überarbeitet, verbessert, wieder getestet und weiterentwickelt werden. Dazu gibt es noch einige zusätzliche Hilfen. Daraus wird dann ein ganzes System, die *Entwicklungsumgebung*.

In welcher Sprache wird programmiert?

Leider kannst du nicht so programmieren, wie dir der Schnabel gewachsen ist. Eine *Programmiersprache* muss so aufgebaut sein, dass möglichst viele Menschen in möglichst vielen Ländern einheitlich damit umgehen können.

Weil in der ganzen Welt Leute zu finden sind, die wenigstens ein paar Brocken Englisch können, besteht auch fast jede Programmiersprache aus englischen Wörtern. Es gab auch immer mal Versuche, z. B. in Deutsch zu programmieren, aber meistens klingen die Wörter dort so künstlich, dass man lieber wieder aufs Englische zurückgreift.

In diesem Buch hast du es mit der Programmiersprache C (gesprochen wie der Zeh) zu tun. Sie ist seit vielen Jahren sehr weit verbreitet und nicht allzu schwer zu erlernen. Daneben gibt es noch einige andere Sprachen wie z. B. C++ (gesprochen: Zeh-Plusplus), Java und C# (gesprochen: Zeh-Sharp).

Noch immer sind die meisten Programme in C geschrieben. Und kaum eine Sprache bietet so viele Freiheiten, was allerdings auch verhängnisvoll sein kann, wenn man was falsch macht. C ist auch deswegen besonders inte-



ressant, weil sie die Mutter der drei oben genannten Sprachen ist, sie alle sind von C abgeleitet.

Und das Betriebssystem?

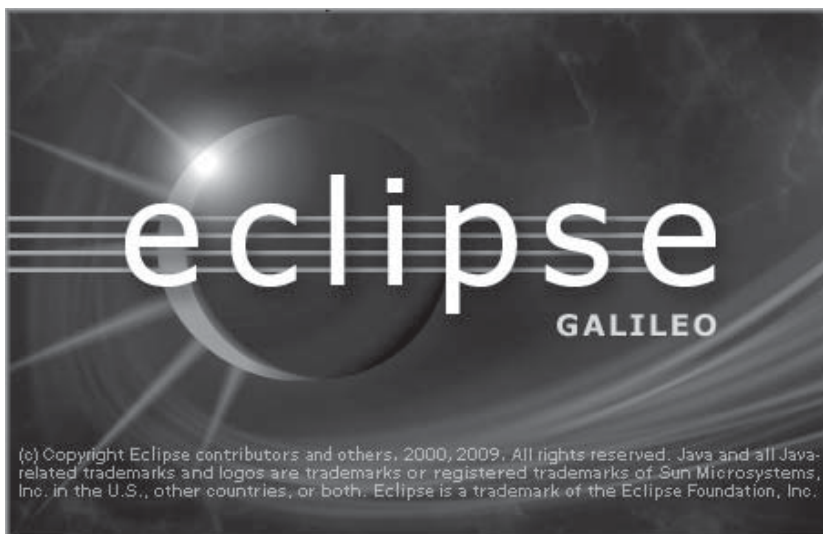
Das Betriebssystem sorgt dafür, dass der Betrieb deines Computers möglichst reibungslos ablaufen kann. Es übernimmt u.a. die Verwaltung des Arbeitsspeichers und der Geräte, die mit dem Computer verbunden sind: von Monitor und Tastatur bis zu Festplatten und anderen Laufwerken.

Fast jeder Computer arbeitet heute mit dem Betriebssystem *Windows* oder *Linux*. C ist plattformunabhängig. Was heißt das? Du kannst unter *Windows* in C programmieren und dieselben Programme auch unter *Linux* zum Laufen bringen – und umgekehrt.

Eine Einschränkung gibt es da allerdings: Die Unabhängigkeit vom Betriebssystem gilt natürlich nur für C-Systeme, die sich eng an einen Standard halten, der allgemein anerkannt ist. Für C gilt der so genannte ANSI- oder ISO-Standard, der in der letzten Version die Nummer 99 trägt – kurz ANSI 99 bzw. ISO 99.



Um in C zu programmieren und die erstellten Projekte auch zu testen, benutzen wir mit *Eclipse* eine Entwicklungsumgebung, die einfach zu handhaben und schon für die Sprache Java weit verbreitet ist (dazu gibt es mit *Java mit Eclipse für Kids* ein Buch aus dieser Reihe).



Auf der CD findest du je eine Eclipse-Version für Linux und für Windows, die beide schon auf C vorbereitet sind. Außerdem gibt es weitere »Zutaten«
© des Titels »C für Kids« (ISBN 978-3-8266-8670-2)

2009 by Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidelberg
Nähere Informationen unter: <http://www.it-fachportal.de/8670>



wie MinGW, ein C-System für Windows, und GTK, einen grafischen Werkzeugkasten für Linux und für Windows.

Während eigentlich alle Linux-Systeme von Haus aus schon ein C-System integriert haben, muss das bei Windows erst nachgerüstet werden. Hier ist es MinGW.

Aller Anfang ist schwer

Ich würde lügen, wenn ich behaupte, dass Programmieren einfach ist. Immerhin musst du eine neue Sprache lernen, die nicht nur aus ein paar englischen Wörtern besteht. So manches wirst du erst nach einiger Übung verstehen. Nicht selten kommt es vor, dass du die Lust verlierst, weil einfach gar nichts klappen will. Das Programm tut etwas ganz anderes, du kannst den Fehler nicht finden, und du fragst dich: Wozu soll ich eigentlich programmieren lernen, wo es doch schon genug Programme gibt?

Trotzdem meine ich: Es ist nicht nur einen Versuch wert, es lohnt sich, es immer wieder zu probieren. Hast du erst mal »Blut geleck«, dann lässt es dich so schnell nicht wieder los ...

Und was bietet dieses Buch?

Über eine ganze Reihe von Kapiteln verteilt lernst du

- ◇ ausführliche Grundlagen von C kennen
- ◇ mit Eclipse unter Windows und Linux umzugehen
- ◇ mit einem grafischen Toolkit (GTK) zu arbeiten – das gehört nicht zum C-Standard, doch das Kennenlernen lohnt sich, zumal es unverändert unter Windows und Linux funktioniert

Im *Anhang* gibt es dann noch eine ganze Menge an Informationen und Hilfen. Auch für deine Eltern und Lehrer, aber vor allem für dich!



Einleitung

Wie arbeitest du mit diesem Buch?

Grundsätzlich besteht dieses Buch aus viel, viel Text und einer Menge Abbildungen dazwischen. Natürlich habe ich mich bemüht, alles so zuzubereiten, dass daraus lauter gut verdauliche Happen werden. Damit das Ganze noch genießbarer wird, gibt es zusätzlich noch einige Symbole, die ich dir hier gern erklären möchte:

Arbeitsschritte

➤ Wenn du dieses Zeichen siehst, heißt das: Es gibt etwas zu tun. Damit kommen wir beim Programmieren Schritt für Schritt einem neuen Ziel immer näher.

Grundsätzlich lernt man besser, wenn man einen Programmtext selbst eingibt oder ändert. Aber nicht immer hat man große Lust dazu. Weil du alle Programme aus dem Buch auch auf der CD findest, ist hinter einem Programmierschritt auch der jeweilige Dateiname (z. B. HALLO1.C) angegeben. Wenn du also den Programmtext nicht abtippen willst, kannst du stattdessen diese Datei laden (zu finden auf der CD im Ordner PROJEKTE).

Fragen und Aufgaben

Am Ende eines Kapitels wirst du jeweils einige Fragen und Aufgaben entdecken. Diese Übungen sind nicht immer ganz einfach, aber sie helfen dir, noch besser zu programmieren. Lösungen zu den Fragen und Aufgaben findest du in verschiedenen Formaten auf der CD im Verzeichnis LOESUNG. Du kannst sie dir alle im Editor von Windows oder auch in deinem Textverarbeitungsprogramm anschauen. Oder du lässt sie dir ausdrucken und hast sie dann schwarz auf weiß, um sie neben den PC zu legen. (Die Programme zu den Aufgaben stehen im Ordner LOESUNG/AUFGABEN auf der CD.)

Wenn eine aktuelle Aufgabe die Änderung eines Projekts betrifft, steht direkt dahinter in Klammern mit einem Pfeil (→) versehen der Name der Datei, in der das jeweilige Programm zu finden ist (→ MATHE1.C).



Notfälle

Manchmal hast du etwas falsch gemacht oder etwas vergessen. Oder es wird gerade ganz knifflig. Dann fragst du dich, was du nun tun sollst. Hund Buffi bietet dir immer eine Lösungsmöglichkeit. Notfalls kannst du aber auch ganz hinten im *Anhang E* nachschauen, wo eine Reihe von häufigen Programmierpannen aufgeführt ist.



Wichtige Stellen im Buch

Hin und wieder findest du ein solch dickes Ausrufezeichen im Buch. Dann ist das eine Stelle, an der etwas besonders Wichtiges steht.



Wenn es um eine ausführlichere Erläuterung geht, tritt wieder Buffi in Erscheinung und schnuppert in seiner Kiste mit Tipps & Tricks.



Was brauchst du für dieses Buch?

Die CD zum Buch

Du findest ein Paket aus Eclipse und C-System als komplette Entwicklungsumgebung auf der beiliegenden CD. Außerdem als Dreingabe das GTK. (Das gilt für Windows, für Linux ist es nur Eclipse, weil die anderen Pakete entweder vorhanden sind oder sich über das Internet nachinstallieren lassen.)

Die *Beispielprogramme* in diesem Buch sind ebenfalls auf der CD gespeichert, falls du mal keine Lust zum Abtippen hast (→ PROJEKTE). Und auch die *Lösungen* zu den Übungsaufgaben sind dort zu finden (→ LOESUNG). Damit alles sowohl unter Linux als auch unter Windows funktioniert, gibt es nur die reinen Quelltexte. Wie man die in ein Projekt einfügt, erfährst du z. B. ausführlich in *Anhang D*.

Speichermedien

Auf jeden Fall ist ein Speichermedium wie ein USB-Stick sinnvoll, auch wenn du deine Programme auf der Festplatte unterbringen willst. Denn auf einem Extraspeicher sind deine Quelltexte zusätzlich sicher aufgehoben.

Was brauchst du für dieses Buch?



Bitte deine Eltern oder Lehrer um Hilfe: Sie sollen den *Anhang A* (und vielleicht auch noch *Anhang B/C* oder *D*) lesen. Dann können sie dir bei den ersten Schritten besser helfen.

Es wäre nützlich, wenn du schon einmal mit einem (einfachen) Textprogramm gearbeitet hast. Wenn nicht, probier doch mal den Editor von Windows oder Linux aus.

Wenn du dich lieber noch mal ausführlicher über den Umgang mit dem PC und mit Windows informieren willst, ehe du ins Programmieren einsteigst, ist es besser, sich erst mal ein grundlegendes Buch über Computer anzuschauen, z. B. aus der Reihe für Kids (*PCs für Kids*).



Fühlst du dich aber fit genug, dann lass uns beginnen!