

Programmeren voor Dummies

5e editie

Spiek
Brief

Tips voor het schrijven van programma's

- ✓ Gebruik duidelijke namen voor variabelen.
- ✓ Gebruik de juiste gegevenstypen.
- ✓ Gebruik zo veel mogelijk eenvoudige algoritmen en gegevensstructuren.
- ✓ Wees niet zuinig met commentaartekst (of schrijf zulke duidelijke code dat commentaar overbodig is).
- ✓ Maak modulaire programma's door een groot programma op te delen in meerdere kleine programma's die zo veel mogelijk op zichzelf staan.
- ✓ Zoek de grenzen van je programma op door extreem hoge en lage getallen of andere extreme invoer op te geven.
- ✓ Kies geschikte algoritmen en gegevensstructuren voor je programma.
- ✓ Verwijder alle overbodige opdrachten en variabelen.
- ✓ Zorg er allereerst voor dat een programma goed werkt; ga het pas daarna optimaliseren om het sneller en compacter te maken. Vergeet niet dat een monsterlijk groot en traag programma dat werkt altijd nog beter is dan een klein, snel programma dat niet werkt.

Gangbare lusstructuren

Zie bijlage A voor meer lusstructuren.

Liberty BASIC

```
WHILE voorwaarde  
  opdrachten  
WEND
```

```
FOR i = begin TO einde  
  opdrachten  
NEXT i
```

C++

```
while (voorwaarde)  
{  
  opdrachten  
}
```

```
for (initialisatie; voorwaarde; stap)  
{  
  opdrachten  
}
```

FreeBASIC

```
WHILE voorwaarde  
  opdrachten  
WEND
```

```
FOR i = begin TO einde  
  opdrachten  
NEXT i
```

LiveCode

```
repeat while voorwaarde  
  opdrachten  
end repeat
```

```
repeat with i=begin to einde  
  opdrachten  
end repeat
```

Veelgebruikte wiskundige operatoren

Wiskundige bewerking	Bijbehorend symbool	Wiskundige bewerking	Bijbehorend symbool
Optelling	+	Modulo	mod
Aftrekking	-	Vermenigvuldiging	*
Deling	/ (voorwaartse slash)	Machtsverheffing	^
Integerdeling	\ (achterwaartse slash)		

Inhoud in vogelvlucht

Over de auteur	xvii
Dankwoord	xviii
<i>Inleiding</i>	1
<i>Deel I: Een computer programmeren</i>	7
Hoofdstuk 1: Beginnen met programmeren	9
Hoofdstuk 2: Alles over programmeertalen	19
Hoofdstuk 3: Een programma schrijven	37
Hoofdstuk 4: De gereedschapskist van een computerprogrammeur	49
<i>Deel II: De bouwstenen van programma's</i>	63
Hoofdstuk 5: Aan de slag	65
Hoofdstuk 6: De structuur van een computerprogramma	73
Hoofdstuk 7: Variabelen, constanten en commentaar	89
Hoofdstuk 8: Getallen kraken en spelen met strings	103
Hoofdstuk 9: Beslissingen, beslissingen... ..	117
Hoofdstuk 10: In herhaling vallen	137
Hoofdstuk 11: Een programma verdelen in subprogramma's	151
Hoofdstuk 12: Dingen opslaan in arrays	167
Hoofdstuk 13: Spelen met objectgeoriënteerd programmeren	181
<i>Deel III: Programmeren voor gevorderden</i>	195
Hoofdstuk 14: Zoek- en sorteeralgoritmen	197
Hoofdstuk 15: Programma's debuggen	221
Hoofdstuk 16: Programmacode optimaliseren	233
Hoofdstuk 17: Een gebruikersinterface maken	245
<i>Deel IV: Programmeren voor internet en smartphones</i>	261
Hoofdstuk 18: Spelen met HTML	263
Hoofdstuk 19: Interactieve webpagina's met Java en JavaScript	289
Hoofdstuk 20: Mobiele apps maken	307
<i>Deel V: Het deel van de tientallen</i>	327
Hoofdstuk 21: Tien nuttige bronnen voor programmeurs	329
Hoofdstuk 22: Tien potentiële programmeercarières	341
Bijlage A: Gangbare lus- en beslissingsstructuren	351
Bijlage B: Gratis compilers en interpreters	355
Bijlage C: Gebruikelijke programmeertermen	357
<i>Index</i>	385

Inleiding



Iedereen kan met een computer leren programmeren. Programmeren vereist geen hoog IQ of een aangeboren talent voor toegepaste wiskunde. Het programmeren van een computer vereist twee dingen: de wil om dingen te leren en vooral veel geduld en doorzettingsvermogen.

Programmeren is een handigheid, zoals bergbeklimmen, tapdansen of polsstokspringen. Sommige mensen zijn er van nature beter in dan anderen, maar iedereen kan het uiteindelijk met de nodige oefening onder de knie krijgen. Dit is ook de reden waarom veel kinderen en tieners al op jonge leeftijd de status van *wizzkid* hebben bereikt: ze zijn niet geniaal, maar hebben gewoon veel tijd besteed aan het ontwikkelen van een nieuwe vaardigheid. Verder zijn ze niet bang om fouten te maken, want fouten maken is nu eenmaal een belangrijk onderdeel van dingen leren.

Misschien heb je er wel eens aan gedacht hoe het zou zijn om zelf programma's te kunnen schrijven. Nou, neem van ons aan dat het niet saai of eentonig is! Programmeren kan erg leuk zijn, maar soms ook frustrerend of vervelend. Bovenal is het tijdrovend. Speciaal om deze laatste redenen is dit boek uitgebracht, zodat je leert hoe je met de minste problemen en het meeste plezier computers leert programmeren.

Of je het nu wilt leren voor je plezier, een nieuwe carrière in de IT wilt beginnen of misschien ook je huidige werk eenvoudiger wilt maken: beschouw dit boek als je persoonlijke gids in de avontuurlijke wereld van programmeren.

We moeten je wel gelijk een illusie armer maken: ook dit boek zal je niet binnen enkele dagen omtoveren in een softwaregoeroe. Wel kan het je op een eenvoudige manier duidelijk maken hoe programma's werken, wat de sterke en zwakke kanten zijn van de verschillende programmeertalen en hoe je zelf kunt beginnen met het schrijven van programma's.

Voor wie dit boek bedoeld is

Dit boek is een goede aankoop als je wilt begrijpen hoe het programmeren met computers in zijn werk gaat, zonder echter in de eindeloze details van een programmeertaal verstrikt te raken. Als je eenmaal begrijpt hoe programmeren in het algemeen gaat, kun je veel gemakkelijker leren werken met programmeertalen met cryptische namen zoals C++ en Java. Koop dit boek vooral als je een van de volgende dingen wilt weten:

- ✓ hoe computerprogramma's werken;
- ✓ wat de overeenkomsten tussen de programmeertalen zijn;
- ✓ hoe je programma's schrijft voor meerdere verschillende besturingssystemen, zoals Windows, Mac OS en Linux;
- ✓ of je je volgende programma het beste kunt schrijven in Visual Basic, C++, Perl, SmallTalk, C# of een andere programmeertaal.

Net zoals iedere andere vaardigheid kun je ook programmeren alleen leren door veel te oefenen. Dit boek helpt je de benodigde praktijkervaring op te doen door op de website bij dit boek naar verschillende compilers te linken, zodat je programma's kunt leren schrijven op iedere computer die werkt met Windows, Mac OS X of Linux.

De drie voornaamste programmeertalen die in dit boek aan de orde komen zijn BASIC, C++ en een scripttaal met de naam LiveCode.

BASIC is speciaal ontworpen om beginners te leren programmeren; in de loop van dit boek leer je dan ook BASIC-programma's schrijven in FreeBASIC, een variant van deze taal. Ook maak je kennis met Liberty BASIC.

FreeBASIC biedt de programmeertaal BASIC in zijn puurste en simpelste vorm, zodat je de beginselen van het programmeren kunt leren zonder verzand te raken in de technische details.

Dit boek biedt ook programmavoorbeelden in C++, de populairste programmeertaal die op dit moment in gebruik is. Als je ooit van plan bent professioneel te gaan programmeren, moet je op zijn minst vertrouwd zijn met de manier waarop C++ werkt.

Als laatste maak je kennis met een niet-traditionele programmeertaal met de naam LiveCode, waarmee je de computer opdrachten geeft door iets wat lijkt op gewoon Engels. Scripttalen zoals LiveCode zijn zo ontworpen dat zowel het programmeren als het doorgronden van een programma zo eenvoudig mogelijk wordt. Met dergelijke talen is het oplossen van ingewikkelde programmeervraagstukken vaak eenvoudiger dan met traditionele talen zoals BASIC en C++.

Ook kunnen we nog opmerken dat LiveCode is gebaseerd op AppleScript, de programmeertaal die gebruikt wordt om taken op Apple-computers te automatiseren. Dit betekent dat je via LiveCode ook meteen vertrouwd raakt met AppleScript.

Doordat je kennismaakt met drie verschillende talen, BASIC, C++ en LiveCode, leer je vanzelf ook de verschillende stijlen en manieren van aanpak kennen om problemen op te lossen. Ook merk je wat de voordelen en beperkingen zijn van elk van deze talen, zodat je uiteindelijk in staat bent de taal te kiezen die het geschiktst is om een bepaald probleem op te lossen.

De opbouw van dit boek

Dit boek bestaat uit vijf delen, zodat je snel kunt vinden wat je zoekt. Elk van de delen behandelt een bepaald onderwerp op het gebied van programmeren. Zodra je iets wilt weten, blader je gewoon door het boek totdat je het deel hebt gevonden met het gezochte onderwerp.

Deel I: Programmeren met een computer

Het programmeren met computers mag misschien lijken op zwarte magie, maar dat is slechts schijn. In dit boek worden de bekende mythen over programmeren aan de kaak gesteld, laten we precies zien hoe computerprogramma's werken en leggen we uit waarom programmeren lang niet zo moeilijk is als de meeste mensen denken.

In dit deel lees je ook hoe de kunst van het programmeren zich heeft ontwikkeld, waarom er zoveel verschillende programmeertalen zijn en hoe bijna elk programma is gebaseerd op vaste, gemakkelijk te onthouden principes, zodat je bijna meteen aan de slag kunt met het programmeren met je eigen computer.

Deel II: De bouwstenen van programmeertalen

Ook al bestaan er vele honderden programmeertalen, al deze talen hebben bepaalde fundamentele overeenkomsten. In dit deel gaan we in op de simpele bouwstenen waaruit programma's zijn opgebouwd, ongeacht de gebruikte taal.

Je vindt hier allerlei voorbeelden van hoe telkens dezelfde problemen in verschillende programmeertalen worden opgelost, zodat je al snel leert de basisstructuren van programma's te herkennen. Natuurlijk kun je deze voorbeelden ook op jouw eigen computer uitproberen.

Deel III: Geavanceerde programmeerprincipes

Een simpel programma leren schrijven is één ding, maar het wordt wat ingewikkelder als je het programma efficiënter wilt maken, fouten wilt opsporen en daadwerkelijk interactie met gebruikers wilt laten plaatsvinden. Het is immers de bedoeling dat mensen het programma gebruiken! In dit deel zie je hoe programmeurs hun software verfijnen en eenvoudig bruikbaar maken (en wat de gevolgen kunnen zijn als ze dat niet doen).

Deel IV: Programmeren voor internet en smartphones

Internet is inmiddels een onmisbaar element geworden bij computergebruik en smartphones zijn niet meer weg te denken. Daarom laten we je

hier zien hoe de verschillende talen voor internet en smartphones werken. Je maakt kennis met HTML (de taal die het uiterlijk van webpagina's bepaalt), JavaScript, Java en PHP en leest over de mobiele besturings-systemen Android, iOS en Windows Phone.

Ook zie je hier hoe programmeurs al die fraaie webpagina's maken die formulieren kunnen tonen en met gebruikers kunnen communiceren. Deze informatie is nuttig als je zelf interactieve webpagina's wilt ontwerpen.

Deel V: Het deel van de tientallen

In dit deel proberen we je te laten wennen aan het idee dat je geld kunt verdienen met programmeren. Daarom vind je hier onder meer een overzicht van veel bekende bedrijven en hun softwareproducten. Ook vind je hier verwijzingen naar allerlei handige softwaregereedschappen en nog meer programmacode waarop je kunt studeren.

Hoe je dit boek gebruikt

De bedoeling van dit boek is duidelijk maken hoe programmeren in zijn werk gaat, echter zonder direct tot in detail in te gaan op één programmeertaal. Het belangrijkste is dat programmacode er na het lezen van dit boek niet langer uit ziet als een onbegrijpelijke toverformule of het typewerk van een dronken aap, maar als iets wat je (misschien met enige moeite) kunt ontcijferen; het is ook de bedoeling dat je gaandeweg zelf iets op een toetsenbord kunt inkloppen wat het daglicht kan verdragen. Dit boek maakt je niet alleen wegwijs, maar kan ook van pas komen als je op zoek bent naar een manier om een bepaald probleem op te lossen.

Lees een stukje, en als je denkt dat je het begrepen hebt, probeer je het gewoon uit op de computer. Speel ook gerust wat met de code – er zal ongetwijfeld van alles verkeerd gaan, maar dat kan doorgaans geen kwaad. Met een beetje geduld en geluk krijg je al snel een goed idee wat programmeren eigenlijk inhoudt.

Open deuren intrappen

We hoeven het eigenlijk niet meer te zeggen: je hebt pas echt wat aan dit boek als je kunt beschikken over een computer. Je kunt immers ook niet je rijbewijs halen met alleen maar een schriftelijke cursus! Het beste kun je hiervoor een computer hebben met Windows of Mac OS X als besturingssysteem, al is Linux ook goed bruikbaar.

In dit boek wordt speciale opmaak gebruikt om bepaalde informatie te benadrukken. De items die je moet typen zijn gezet in een *niet-proportioneel lettertype*. Alle code en de berichten op het scherm worden in

hetzelfde lettertype weergegeven. Wanneer ik een nieuwe term definieer, is dat woord *cursief* gedrukt.

De pictogrammen in dit boek

We gebruiken enkele pictogrammen om de aandacht te vestigen op handige tips, belangrijke informatie of technische details (die je waarschijnlijk net zo snel ook weer vergeet). Let op de volgende pictogrammen:



Dit pictogram wordt gebruikt bij handige informatie die je veel werk en tijd kan besparen (zolang je dit tenminste onthoudt, natuurlijk).



Hier vertellen we je iets belangrijks, wat je zeker niet wilt vergeten.



Dit pictogram waarschuwt voor fouten, valkuilen en andere ellende die op de loer ligt.



Dit pictogram vind je als een bepaald stuk programmacode op de website bij dit boek is terug te vinden. Als je slim bent, kopieer je de code dan ook gewoon vanaf internet, zodat je niet alles hoeft over te typen.



Dit pictogram wijst op informatie die aardig is om te weten, maar verder niet erg belangrijk is. Alles rondom dit pictogram mag je dus meteen weer vergeten! (Maar als je echt ambitieus bent, stamp je natuurlijk alle technische informatie die je kunt vinden in je hoofd, zodat je je uiteindelijk kunt meten met andere techneuten van wereldklasse.)

De website bij dit boek

De enige manier om te leren programmeren is zelf programma's te gaan schrijven en daarvoor heb je natuurlijk een compiler nodig – en wel minstens eentje voor elke taal die je wilt gebruiken. Daarom vind je op de website bij dit boek links naar de programmeertalen, compilers en andere hulpmiddelen die in dit boek worden gebruikt. Ook vind je op de webpagina een download met de code van een aantal in dit boek besproken listings.

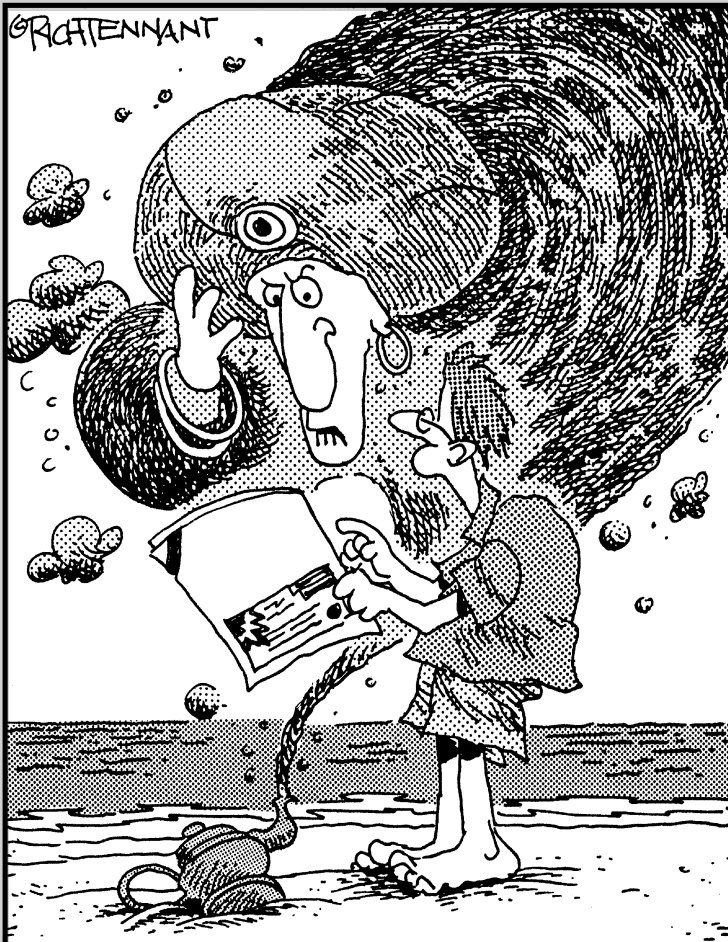
Je vindt deze website op www.dummies.nl/downloads.

Deel I

Een computer programmeren

The 5th Wave

By Rich Tennant



'Kan ik je niet gewoon onmetelijke rijkdom geven of zo?'

In dit deel...

Op het eerste gezicht lijkt het misschien een bijna onoverkomelijke uitdaging om een computer te programmeren, dus laten we je in dit deel voorzichtig kennismaken met deze fascinerende tak van wetenschap. Als eerste zie je precies welk doel programma's hebben en hoe professionals ze maken.

Daarna lees je waarom er zoveel programmeertalen zijn en waarom sommige talen veel populairder zijn dan andere. Je maakt kennis met de verschillende gereedschappen die door programmeurs worden gebruikt om een programma van begin tot einde te ontwerpen, bewerken en verspreiden.

Als laatste ontdek je waar je rekening mee moet houden als je besluit zelf een programma te gaan schrijven. Je maakt kennis met de voor- en nadelen van de verschillende programmeertalen en je komt erachter waarom zelfs mensen met amper enige ervaring in het programmeren er toch in slagen een werkend programma te schrijven.

Na het lezen van dit deel van het boek heb je een aardig inzicht gekregen in hoe je een programma schrijft, welke achtereenvolgende stappen je daarvoor doorloopt en hoe je een idee voor een programma omzet in een daadwerkelijk functionerend product dat door anderen kan worden gebruikt. En wie weet? Misschien ben *jij* wel degene die met een goed idee en veel doorzettingsvermogen een gigantisch populair programma maakt, en zo het volgende softwarebedrijf met een miljoenenomzet sticht!

Hoofdstuk 1

Beginnen met programmeren

.....

In dit hoofdstuk:

- Leren wat programmeren inhoudt
 - Begrijpen hoe een programma werkt
 - Wat er komt kijken bij programmeren
-

Ongeacht wat je misschien hebt gehoord: een computer programmeren is niet zo moeilijk. Programmeren is een vaardigheid die iedereen onder de knie kan krijgen met genoeg geduld, oefening en cafeïnehoudende dranken.

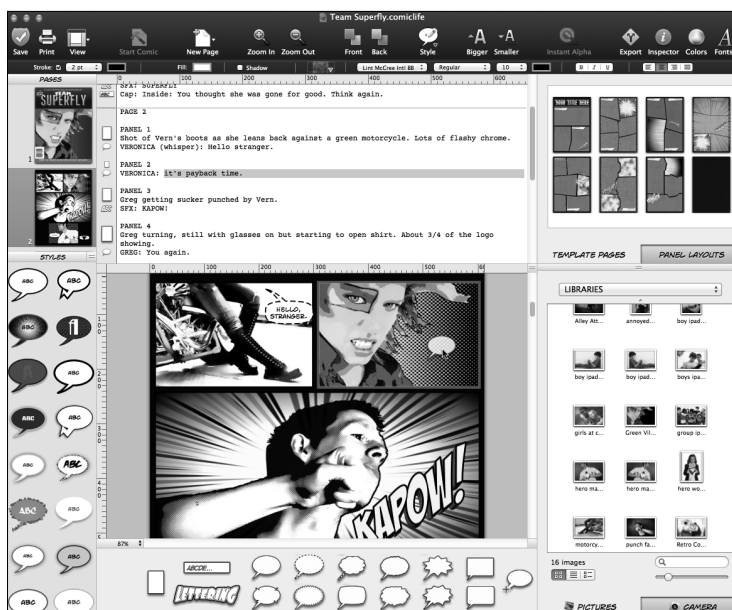
Computers mogen dan vreselijk ingewikkelde apparaten lijken, maar daar hoeft je je niet druk om te maken. Er zijn relatief weinig mensen die weten hoe een verbrandingsmotor precies werkt, maar verreweg de meeste mensen leren vroeg of laat toch autorijden. Programmeren is weinig anders: je kunt het met wat oefening leren, zonder dat je precies hoeft te weten hoe een computer werkt.

Waarom zou je leren programmeren?

De eerste vraag die vaak gesteld wordt, ligt voor de hand: ‘Waarom zou je met je computer willen leren programmeren?’ Het antwoord hangt natuurlijk af van wat je ermee wilt bereiken. Hier zijn enkele goede redenen:

- ✓ **Omdat het leuk is.** Mensen storten zich op van alles en nog wat, zoals stijldansen, skiën, tuinieren en schilderen, simpelweg omdat het ze leuk lijkt. De meeste van deze mensen zullen nooit een professionaal worden in hun favoriete hobby, maar dat vinden ze niet erg; voor hen is het gewoon een leuke tijdsbesteding. Hetzelfde gaat op voor programmeren; misschien vind je het gewoon leuk om een simpel programmaatje te schrijven dat de lelijke kop van je baas op het scherm tovert. En als je wat handiger bent en veel geluk hebt, schrijf je misschien een complexer programma dat miljoenen in het laatje brengt, zodat je nooit meer voor een baas met een lelijke kop hoeft te werken. In figuur 1.1 zie je een programma met de naam *Comic*

Figuur 1.1: In Comic Life maak je een stripverhaal van een collectie foto's

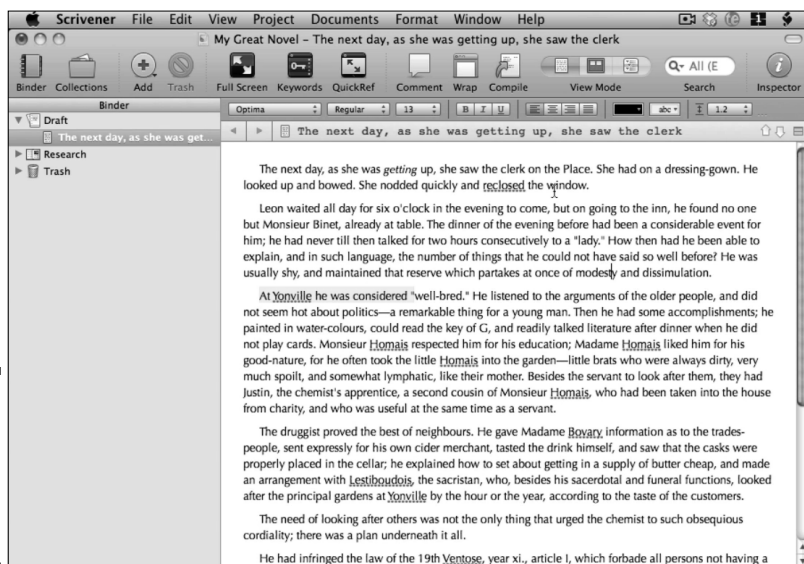


Life, waarmee je foto's kunt omtoveren tot een stripverhaal. Dit programma is geschreven op een Macintosh-computer, in een programmeertaal met de naam Objective-C.

- ✓ **Om een probleem op te lossen.** Veel mensen leren programmeren zonder de bedoeling er ooit hun brood mee te verdienen. Ze willen graag een specifiek probleem oplossen, maar omdat ze geen programma kunnen vinden dat aan hun wensen voldoet, besluiten ze het dan maar zelf te gaan schrijven. Zo leek het iemand ooit wel handig om een programma te hebben dat zijn belastingaanslag vlot kon doorrekenen. Er bleek geen geschikte software op de markt te zijn, dus besloot de man zichzelf programmeren te leren en zelf een belastingprogramma te schrijven. Het resultaat was TurboTax, inmiddels al jaren een van de populairste programma's in de Verenigde Staten voor het invullen van de belastingaangifte. Op dezelfde manier vond een freelanceschrijver dat een gewone tekstverwerker niet het beste gereedschap was voor het schrijven van zijn roman. Hij ontwikkelde daarom een speciale tekstverwerker met de naam Scrivener (zie figuur 1.2), voorzien van extra functies voor het structureren van een verhaal. Scrivener is geschreven in de ontwikkelomgeving Xcode, die gebaseerd is op C. Dit is ook het leuke van programmeren: als je een programma hebt geschreven om een eigen probleem op te lossen, blijken andere mensen er vaak ook mee geholpen.

- ✓ **Als start van een nieuwe carrière.** Zolang het aantal computers wereldwijd alleen maar blijft stijgen, zul je nooit lang zonder werk zitten als je behoorlijk kunt programmeren. Softwaredeskundigen zijn overal zeer in trek, niet alleen om nieuwe programma's te ont-

Figuur 1.2:
Scrivener
helpt schrij-
vers bij het
ordenen van
een verhaal



wikkelen, maar ook om bestaande programma's te onderhouden, te repareren of aan te passen. Computers en dus ook programma's zijn tegenwoordig in bijna iedere bedrijfstak doorgedrongen. Van boerderijen tot banken, van scholen tot hotels, van autoverhuurbedrijven tot luchtvaartmaatschappijen – overal zijn mensen nodig die met software overweg kunnen. Het aardige is ook dat leren programmeren bij uitstek iets is wat je naast je gewone werk kunt doen, zonder dat je er bijzondere dingen voor nodig hebt. Een groep alternatieve genezers heeft bijvoorbeeld een programma met de naam IBIS geschreven, bedoeld als informatiebron over hoe allerlei aandoeningen genezen kunnen worden met acupunctuur, massage, homeopathie of een bepaald dieet (zie figuur 1.3). IBIS is geschreven in de programmeertaal Revolution, de voorloper van LiveCode, die in dit boek uitgebreid aan bod komt.

- **Omwille van de uitdaging.** Veel mensen zijn gefascineerd door computers vanwege de complexe functies die deze machines kunnen uitvoeren. Het is dan ook niet verwonderlijk dat juist veel mensen met bovengemiddelde intelligentie computers zien als gereedschap om hun eigen gedachten te kunnen ordenen en uitdrukken. Een programmeur heeft om deze reden het programma Axon Idea Processor (zie figuur 1.4) gemaakt met Prolog, een populaire programmeertaal in de wereld van kunstmatige intelligentie. Het doel was een programma te maken dat mensen zou helpen ideeën, concepten en feiten te ordenen en te verwerken, zodat ze niet alleen oplossingen kunnen vinden voor bepaalde problemen, maar ook hun eigen redeneringen kunnen analyseren en verbeteren. Werken met bestaande software is vaak een dagelijkse sleur, maar zelf een programma schrijven is dat beslist niet. Je merkt ineens dat je echt je hoofd moet gebruiken!