

Programmieren für Jung und Alt

Select from

Kai König

Ob Java, Swift oder Ruby: Das Erlernen einer Programmiersprache kann kinderleicht sein. Wie einige Apps beweisen.



Geht es in den Massenmedien um die schulische und berufliche Ausbildung junger Menschen, wird oft der Eindruck erweckt, dass heute bereits Vorschulkinder das Programmieren erlernen müssen, um überhaupt eine Chance im Berufsleben zu bekommen. Das ist sicherlich eine überspitzte Darstellung der gesellschaftlichen Entwicklung; allerdings ist eine zunehmende Beschleunigung der Technisierung in einer Vielzahl von Berufen zu beobachten.

Grundkenntnisse der Programmierung früh im Leben zu erlernen, kann daher gegebenenfalls sinnvoll sein. Diverse Apps können den Einstieg in die Programmierung erleichtern. Das betrifft nicht nur Apps für Kinder, sondern auch solche, die Quereinsteiger und bisherige Passivnutzer der Informationstechnik an die interessante Tätigkeit der Softwareentwicklung heranführen.

Die App-Infos der iX 12/15 haben bereits diverse Editoren und andere Werkzeuge zur Softwareentwicklung vorgestellt. Zwar sind sie nicht unbedingt zum Einstieg in die Programmierung gedacht, können sich aber als nützliche Werkzeuge für die nächsten Schritte erweisen. Da nahezu alle dort vorgestellten Apps nicht auf Deutsch verfügbar sind, sollte man zumindest Grundkenntnisse der englischen Sprache haben, wenn man sie nutzen will.

Hüpfen und Dinosaurier bewegen

Bei der kostenlosen iOS-App „Hopscotch“ („Himmel und Hölle“) handelt es sich in erster Linie um eine Programmierumgebung für Kinder, mit der sie einfache Spiele, Animationen oder Puzzles entwickeln können. Allerdings ist sie auch für erwachsene Einsteiger in die Programmie-

rung interessant, und selbst erfahrene Entwickler können Spaß mit der App haben. Hopscotch basiert auf den Ideen der schon etwas älteren iOS-App „Daisy the Dinosaur“ (ebenfalls kostenlos), die man noch immer problemlos auf aktuellen Geräten und iOS-Versionen ausführen kann. Samantha John hat die App im Rahmen ihrer Arbeit am MIT entwickelt, um Frauen einen einfachen und spielerischen Weg in die Programmierung anzubieten.

Bei Daisy the Dinosaur geht es darum, den Dinosaurier Daisy mit einfachen Kommandos auf einem Weg zu bewegen und zu drehen oder wachsen zu lassen beziehungsweise wieder zu verkleinern. Neben diesen einfachen Kommandos bietet die App Befehle zur Wiederholung von Kommandos (zum Beispiel *Repeat 5*) oder zum bedingten Ausführen von Befehlen mithilfe eines *When*-Konstrukts, etwa wenn man auf Daisy tippt oder das Mobilgerät schüttelt.

Das Programmieren von Daisy findet vollständig visuell statt, man zieht die Kommandos in den Programmbereich und kann sie dort mit weiteren Touchscreen-Gesten umstellen und ordnen. Daisy the Dinosaur hat zwei Spielmodi. Zum Einstieg bietet es sich an, die Tutorials im „Challenge Mode“ zu spielen und erst danach den sogenannten „Free-Play Mode“ auszuprobieren. In letzterem kann der Anwender mit allen verfügbaren Kommandos frei Programme entwerfen und Daisy beliebige Aktionen ausführen lassen.

Hopscotch übernimmt die Grundidee der visuellen Programmiersprache von Daisy the Dinosaur, geht aber weit über deren Möglichkeiten hinaus. Die App benötigt einen Hopscotch-Account, der sich beim ersten Start der App kostenlos erstellen lässt. Für Einsteiger bietet sich das Nachbauen einer der Vorlagen an – im Fall eines Spiels handelt es sich dabei um

„Geometry Dash“, ein Jump’n’Run-Spiel mit geometrischen Figuren. Während des Tutorials läuft ein Video, das Schritt für Schritt durch das visuelle Programmieren des Spiels führt.

Die visuelle Programmiersprache in Hopscotch ist ausdruckskräftiger als Daisys und erlaubt beispielsweise das Reagieren auf Geräusche, Schütteln oder auch nur leichtes Schräghalten des Gerätes. Es gibt Kollisions-Kommandos für visuelle Elemente auf dem Bildschirm, sodass man auf das Berühren oder Ineinanderstoßen von Objekten reagieren kann. Auch die Befehle, die Bewegungen und Änderungen von Objekteigenschaften auslösen, sind vielfältig und erlauben sogar das dynamische Zeichnen von Spielelementen. Fertige Spiele und Animationen kann der Anwender über die App veröffentlichen und sie so anderen zum Ausprobieren und Spielen zur Verfügung stellen.

Vom Visuellen hin zum Technischen

Einen weiteren Schritt in Richtung Komplexität bietet die für iOS und Android verfügbare App „Tynker“. Tynker ist ebenfalls eine App zum Entwickeln von Spielen oder Puzzles für Programmierneinsteiger, bietet allerdings vielfältigere Optionen hinsichtlich der verfügbaren grafischen Themen, Character Sprites und vor allem im Bereich der Animation. Im Vergleich zu Hopscotch ist die visuelle Programmierungsumgebung eher für ältere Kinder und Jugendliche konzipiert und beinhaltet in den Kommandoblöcken auch technisches Vokabular wie *if true then*, „infinite loops“ und Anweisungen wie *set rotation style left-right*. Diese Beispiele sollen keine Kritik an Tynker darstellen. Die Sprache funktioniert im Kontext der App sehr gut, stellt aber leicht höhere Anforderungen an die Nutzer als vergleichbare Umgebungen.

Positiv fällt die große Zahl verfügbarer Tutorials auf. Tynker ist kostenlos, bietet allerdings Erweiterungen über In-App-Käufe zu Preisen von 1,99 bis 4,99 Euro – je nach gewähltem Paket. Diese Erweiterungen schalten weitere Code-Herausforderungen und -Spiele in verschiedenen Themenbereichen frei. Tynker erfordert sowohl auf iOS- als auch auf Android-Geräten ein Tablet – auf als Mobiltelefon klassifizierten Geräten lässt sich die App nicht installieren.

Es sei noch angemerkt, dass einer der Urväter der hier vorgestellten visuellen Programmierungsumgebungen die Sprache Scratch ist. Von Scratch gibt es keine App-Umsetzung, MIT Education hat aber

vor einiger Zeit eine vereinfachte Tablet-Version ScratchJr veröffentlicht, die sich explizit an junge Kinder im Alter von fünf bis sieben Jahren (und deren Eltern) richtet.

Wer über das Stadium des visuellen Programmierens hinaus ist, sollte sich die iOS-App „Lrn – Learn to code“ anschauen. Sie ist kostenlos und beinhaltet Kurse in HTML und CSS. Zum Preis von je 2,99 Euro lassen sich weitere Kurse zu Ruby, Python oder JavaScript freischalten, für 6,99 Euro gibt es ein Komplettpaket.

Die Struktur von Lrn erinnert ein wenig an die in vielen Sprachen verfügbaren Koans-Übungen. Lrn präsentiert einen kurzen Informationsblock zum Thema und zeigt dann ein Codebeispiel, in dem der Lernende Inhalte ergänzen muss. Der fertige Code lässt sich dann innerhalb von Lrn ausführen, und das Ergebnis wird von der App kommentiert. Dazu gibt es Quizfragen zum Wiederholen und Flashcards zum Einprägen des Gelernten. Man darf natürlich nicht erwarten, es nur nach Nutzung der App zur Meisterschaft in einer der Sprachen zu bringen, aber Lrn ist eine interessante und gut geeignete App, mit der Einsteiger sich in kurzer Zeit

Grundkenntnisse einer der unterstützten Sprachen aneignen können.

Wer eine ähnliche Lernumgebung für Java oder Swift sucht, ist mit Javvy oder Swifty gut bedient. Bei beiden handelt es sich um kostenlose iOS-Apps vom selben Entwickler, die sich mit In-App-Käufen je nach App zu Preisen von 0,99 Euro bis zu 9,99 Euro über das kostenlos verfügbare Tutorialmaterial hinaus aufwerten lassen. Javvy ist auch für Android erhältlich und folgt dort dem gleichen Preismodell. Beide Apps sind gelungen und empfehlenswert für alle, die sich in die Grundlagen von Java oder Swift einarbeiten wollen.

Wöchentliches Selbststudium mit täglichen Fragen

Einen etwas anderen Ansatz zum Lernen einer Sprache oder Technologie verfolgt „Pocket Programming“ für Android. Die App ist für Rails und Android verfügbar und kostet jeweils 0,92 Euro. Über ein zwei- bis vierwöchiges Programm bekommt man an jedem Tag zehn Fragen gestellt und muss sie – unterstützt durch eigene Recherche – beantworten. Pocket

Alle App-Infos

www.heise.de/ix/online/app-infos/



Programming hat viele Elemente von Gamification und kann daher gut zur Unterstützung des Selbststudiums dienen. Größtes Manko der beiden Apps ist die eher schlechte Übersetzung ins Englische. Leider wimmelt es von Fehlern in Rechtschreibung oder Satzbau.

Zum Abschluss ein Blick auf Apps die das Erlernen von SQL unterstützen sollen. Beim kostenlosen „SQL Primer“ für iOS handelt es sich eher um ein Referenzwerk, das verschiedene Selektions- und Join-Arten anhand von Beispielen auflistet. Hier findet keine darüber hinausgehende Interaktion mit dem Lernenden statt. Das ebenfalls kostenlose „Learn SQL“ für Android und iOS behandelt ähnliche Themen, arbeitet diese aber mit Videos, Tests und motivierenden Elementen wie einer Highscore-Liste deutlich interaktiver und damit interessanter auf. (ka)

Alle Links: www.ix.de/ix1608136



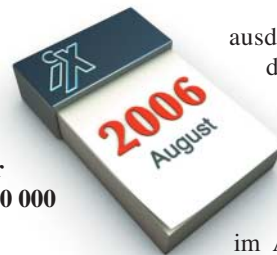
Vor 10 Jahren: Spam, energisch bekämpft

Im Hier und Heute meldete heise online im Juni, dass der US-Amerikaner Sanford Wallace anderthalb Jahre in einem Gefängnis verbringen und 310 000 US-Dollar Schadensersatz zahlen muss.

Wallace hatte eine halbe Million Facebook-Konten gekapert und 27 Millionen Spam-Mails an die Nutzer geschickt. Nicht gemeldet wurde, dass der Richter Maßnahmen verhängte, die Wallace sozialisieren sollen: Nach dem Gefängnisaufenthalt muss er sich in psychologische Betreuung begeben und darf fünf Jahre lang keinen Computer besitzen oder nutzen. Sanford Wallace ist kein Unbekannter: 2008 und 2011 wurde er für seine organisierten Spam-Aktionen verurteilt. Vor 10 Jahren erhielten er und seine Firma SmartBOT dafür die damals höchste Geldstrafe von fünf Millionen Dollar. Zuvor war 2005 das Buch des Journalisten Brian McWilliams über die „Spam Kings“ erschienen. In ihm stellte er dar, wie der Nachtclubbesitzer Wallace mit seiner Firma Cyberpromo in den 90er-Jahren die damals populären Mailboxsysteme Compuserve und America Online mit Spam überschüttete, nachdem er mit Junk-Fax-Aktionen reich geworden war. Immerhin:

anders als David Hawke, der zweite große „Spam King“ der USA, hatte Wallace kein politisches Motiv, sondern war auf das große Geld aus. So versuchte er, Myspace zu erpressen und war im Geschäft mit Online-Pornos tätig. Hawke unterstützte mit seinen Spam-Gewinnen Neonazis wie „Final Solution“. Er wurde in zahlreichen Verfahren unter anderem für den Vertrieb von „pflanzlichem Viagra“ verurteilt.

Vor 10 Jahren kommentierte iX-Redakteur Christian Kirsch in der Ausgabe 8/2006 das „Elektronischer-Geschäftsverkehr-Vereinheitlichungsgesetz“ der großen Koalition. Ein wichtiger Bestandteil dieses Gesetzespakets war das Telemediengesetz, in dem unter anderem Versender von Spam eine Ordnungsstrafe von 50 000 Euro zahlen müssen, wenn ihr Spam im Header der E-Mail nicht eindeutig als Spam identifizierbar ist. „Wie so oft hierzulande kreierte die Gesetzgebungsmaschine und gebar eine zahllose Drohung. Denn was sich die Politiker da haben



ausdenken lassen, erinnert an den Versuch, Banküberfälle durch Parkverbote vor den Filialen zu verhindern,“ schrieb Kirsch. Zum einen kümmerten sich Spammer im Ausland wenig um eine deutsche Richtlinie, zum anderen sei es unerheblich, ob im Absender „Böser Spamversender“ steht und im Betreff „pflanzliches Viagra“ aufgeführt sei: Unerwünscht und lästig sei Spam allemal. „Ginge es wirklich darum, wenigstens europäische Spammer abzuschrecken, müsste man die Tat selbst zumindest als Ordnungswidrigkeit einstufen.“

Das Anti-Spam-Gesetz der großen Koalition hatte in der Tat keine größeren Auswirkungen: Ausländische Spammer kümmerten sich nicht drum und ein bekannter deutscher Versender verschickte hinfort einen „Newsletter zu pflanzlichen Aufbaustoffen“. Damit war dem Reinhaltungsgebot im E-Mail-Header Genüge getan. Was uns zurück zu Wallace bringt. Zu seinen Facebook-Attacken befragt, erklärte er, dass mit E-Mail-Spam kein Geschäft mehr zu machen sei. Nur Social-Web-Plattformen seien noch attraktiv. „Viele nützliche Informationen für sie“ künden umlautbefreit im Spam-Ordner davon, dass er unrecht hatte. Detlef Borchers (ka)