

Einführung in Java mit Greenfoot

Michael Kölling

Einführung in Java mit Greenfoot

**Spielerische Programmierung
mit Java**



PEARSON

Higher Education

München • Harlow • Amsterdam • Madrid • Boston
San Francisco • Don Mills • Mexico City • Sydney

a part of Pearson plc worldwide

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Autor dankbar.

Authorized translation from the English language edition, entitled INTRODUCTION TO PROGRAMMING WITH GREENFOOT: OBJECT-ORIENTED PROGRAMMING IN JAVA WITH GAMES AND SIMULATIONS, 2nd Edition by MICHAEL KOLLING, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2016 Pearson Education Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

GERMAN language edition published by PEARSON EDUCATION DEUTSCHLAND GMBH, Copyright © 2016.

Fast alle Produktbezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt. Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das ®-Symbol in diesem Buch nicht verwendet.

10 9 8 7 6 5 4 3 2 1

20 19 18 17 16

ISBN 978-3-86894-908-7 (Buch)
ISBN 978-3-86326-958-6 (E-Book)

© 2016 by Pearson Deutschland GmbH
Lilienthalstraße 2, D-85399 Hallbergmoos
Alle Rechte vorbehalten
www.pearson.de
A part of Pearson plc worldwide

Programmleitung: Birger Peil, bpeil@pearson.de
Fachlektorat: Prof. Dr. Carsten Schulte, Freie Universität Berlin
Übersetzung: Katharina Pieper, Berlin, pieper.katharina@googlemail.com
Coverabbildung: © Ivan Kmit / Alamy Stock Foto
Herstellung: Philipp Burkart, pburkart@pearson.de
Satz: inpunkt[w]o, Haiger (www.inpunktwo.de)
Druck- und Verarbeitung: Drukkerij Wilco BV, Amersfoort
Printed in the Netherlands

Inhaltsverzeichnis



Vorwort	11
Vorwort zur deutschen Ausgabe	13
Danksagungen	15
Vorwort zur 2. Auflage	17
Einleitung	19
Kapitel 1 Greenfoot kennenlernen	21
1.1 Die ersten Schritte	22
1.2 Objekte und Klassen	23
1.3 Mit Objekten interagieren	24
1.4 Rückgabetypen	25
1.5 Parameter	27
1.6 Die Ausführung in Greenfoot	28
1.7 Ein zweites Beispiel	29
1.8 Das Klassendiagramm verstehen	30
1.9 Mit Asteroiden spielen	31
1.10 Quelltext	32
Kapitel 2 Das erste Programm: Little Crab	37
2.1 Das little-crab-Szenario	37
2.2 Die Krabbe in Bewegung setzen	39
2.3 Drehen	41
2.4 Bildschirmränder	43
Vertiefende Aufgaben	50
Kapitel 3 Das Spiel „Little Crab“ ausbauen – fortgeschrittenere Programmiertechniken	53
3.1 Zufälliges Verhalten einbringen	53
3.2 Würmer hinzufügen	57
3.3 Würmer fressen	59
3.4 Neue Methoden erzeugen	60
3.5 Einen Hummer hinzufügen	62
3.6 Tastatursteuerung	63

3.7	Das Spiel beenden	66
3.8	Sound hinzufügen	67
3.9	Eigenen Sound herstellen	69
3.10	Automatische Vervollständigung des Codes	72
	Vertiefende Aufgaben	74
Kapitel 4 Das Spiel „Little Crab“ fertigstellen		77
4.1	Objekte automatisch erzeugen	77
4.2	Neue Objekte erzeugen	79
4.3	Variablen	80
4.4	Zuweisungen	80
4.5	Objektvariablen	81
4.6	Variablen verwenden	83
4.7	Objekte zur Welt hinzufügen	84
4.8	Die Welt speichern	85
4.9	Bilder animieren	87
4.10	Greenfoot-Bilder	87
4.11	Instanzvariablen (Zustandsfelder)	89
4.12	Die Konstruktoren der Akteur-Klassen	91
4.13	Die Bilder wechseln	93
4.14	Die if/else-Anweisung	94
4.15	Würmer zählen	95
4.16	Weitere Ideen	97
	Vertiefende Aufgaben	99
Exkurs 1 Szenarien teilen		101
E1.1	Dein Szenario teilen	101
E1.2	Auf der Greenfoot-Webseite veröffentlichen	101
E1.3	In eine Webseite exportieren	103
E1.4	In ein Programm exportieren	104
E1.5	In ein Greenfoot-Archiv exportieren	104
Kapitel 5 Punktezahl		107
5.1	WBC: der Ausgangspunkt	108
5.2	WhiteCell: eingeschränkte Bewegung	109
5.3	Bakterien: wie man sich selbst verschwinden lässt	111
5.4	Blutbahn: neue Objekte erzeugen	112
5.5	Seitliche Bewegung	113
5.6	Viren hinzufügen	114
5.7	Kollision: Bakterien entfernen	115
5.8	Variable Geschwindigkeit	115
5.9	Rote Blutzellen	116

5.10	Begrenzungen hinzufügen	117
5.11	Zu guter Letzt: Punktezahlung einbauen	119
5.12	Punktezahlung in der Welt	122
5.13	Abstraktion: Punktezahlung generalisieren	124
5.14	Spielzeit hinzufügen	127
	Vertiefende Aufgaben	129

Kapitel 6 Musizieren: Ein Bildschirm-Klavier **131**

6.1	Die Tasten animieren	132
6.2	Den Sound erzeugen	135
6.3	Abstraktion: mehrere Tasten erzeugen	136
6.4	Das Klavier erstellen	138
6.5	Schleifen: die while-Schleife	139
6.6	Felder	142
	Vertiefende Aufgaben	149

Kapitel 7 Objektinteraktion: eine Einführung **151**

7.1	Interagierende Objekte	152
7.2	Objektreferenzen	152
7.3	Interaktion mit der Welt	153
7.4	Mit Akteuren interagieren	153
7.5	Der Wert null	154
7.6	Interaktionen mit Gruppen von Akteuren	155
7.7	Die Verwendung der Java-Bibliotheksklassen	157
7.8	Der Typ List	159
7.9	Eine Blätterliste	160
7.10	Die for-each-Schleife	160
	Vertiefende Aufgaben	164

Kapitel 8 Interagierende Objekte: Newtons Labor **165**

8.1	Der Ausgangspunkt: Newtons Labor	165
8.2	Hilfsklassen: SmoothMover und Vector	167
8.3	Die bestehende Klasse Body	170
8.4	Erste Erweiterung: Bewegung erzeugen	172
8.5	Die Klasse Color	173
8.6	Gravitationskraft hinzufügen	174
8.7	Gravitationskraft anwenden	177
8.8	Ausprobieren	179
8.9	Gravitationskraft und Musik	181
	Vertiefende Aufgaben	184

Kapitel 9 Kollisionserkennung: Asteroiden	185
9.1 Analyse: Was ist vorhanden?	186
9.2 Sterne zeichnen	187
9.3 Drehen	191
9.4 Vorwärtsfliegen	192
9.5 Mit Asteroiden kollidieren	195
9.6 Spielende	198
9.7 Feuerkraft hinzufügen: die Protonenwelle	202
9.8 Die Ausdehnung der Welle	203
9.9 Mit Objekten interagieren, die im Wirkungsbereich liegen	206
9.10 Verbesserungsmöglichkeiten	208
Vertiefende Aufgaben	211
Exkurs 2 Der Greeps-Wettbewerb	213
E2.1 Die ersten Schritte	214
E2.2 Die Greeps programmieren	215
E2.3 Den Wettbewerb ausführen	216
E2.4 Technische Einzelheiten	217
Kapitel 10 Bilder und Töne erzeugen	219
10.1 Vorbereitende Maßnahmen	219
10.2 Mit Sound arbeiten	221
10.3 Sound in Greenfoot aufnehmen und bearbeiten	222
10.4 Externe Soundaufnahme und -bearbeitung	223
10.5 Sounddateiformate und Dateigrößen	225
10.6 Erweiterte Steuerung: die Klasse GreenfootSound	227
10.7 Mit Bildern arbeiten	228
10.8 Bilddateien und Bildformate	228
10.9 Bilder zeichnen	230
10.10 Bilddateien und dynamisches Zeichnen kombinieren	232
Vertiefende Aufgaben	236
Kapitel 11 Simulationen	239
11.1 Füchse und Hasen	241
11.2 Ameisen	243
11.3 Futter sammeln	245
11.4 Die Welt einrichten	249
11.5 Pheromone hinzufügen	249
11.6 Pfad ausbilden	251
Kapitel 12 Greenfoot und die Kinect	255
12.1 Was kann die Kinect?	256
12.2 Die Software installieren	258

12.3	Die ersten Schritte mit Greenfoot und Kinect	258
12.4	Die einfache Kamera	260
12.5	Der nächste Schritt: greenscreen	261
12.6	Strichmännchen: Benutzer nachzeichnen	262
12.7	Mit den Händen malen	265
12.8	Ein einfaches Kinect-Spiel: Pong	270
12.9	Zusammenfassung	275
	Vertiefende Aufgaben	276
Kapitel 13 Weitere Szenarien		277
13.1	Murmeln	277
13.2	Fahrstühle	279
13.3	Boids	280
13.4	Explosionen	281
13.5	Breakout	282
13.6	Plattform-Springer	283
13.7	Wave	284
13.8	Karten	284
Anhang A Installation von Greenfoot		287
A.1	Greenfoot installieren	287
A.2	Die Buchszenarien installieren	287
Anhang B Greenfoot API		289
Anhang C Kollisionserkennung		295
C.1	Übersicht über die Methoden	295
C.2	Hilfsmethoden	296
C.3	Niedrige kontra hohe Auflösung	296
C.4	Sich überschneidende Objekte	297
C.5	Objekte in der Umgebung	298
C.6	Nachbarn	299
C.7	Objekte im Umkreis	300
Anhang D Anmerkungen zur Java-Syntax		301
D.1	Java-Datentypen	301
D.2	Java-Operatoren	303
D.3	Java-Kontrollstrukturen	305
Anhang E Die RGB-Farbtabelle		313
Register		315



Vorwort

Greenfoot ist eine Programmierumgebung, die zum Selbststudium oder in Anfängerkursen auf Schul-/Hochschulniveau eingesetzt werden kann, um die Prinzipien der Programmierung zu lehren und zu lernen. Dank ihrer Flexibilität und Skalierbarkeit eignet sie sich für Schülerinnen und Schüler sowie für Studierende gleichermaßen.

Greenfoot führt mithilfe der Programmiersprache Java in die objektorientierte Programmierung ein und wurde speziell mit dem Ziel entwickelt, objektorientierte Konzepte und Prinzipien auf leicht verständliche Weise zu vermitteln.

Die Greenfoot-Umgebung macht Interaktionen und die Erstellung von Grafiken zu einem Kinderspiel. Schüler und Studenten können sich darauf konzentrieren, die Anwendungslogik zu ändern und mit Objekten herumzuxperimentieren. Die Entwicklung von Simulationen und interaktiven Spielen ist denkbar einfach und das Feedback erfolgt sofort. Zur Programmierumgebung von Greenfoot gehört jetzt ein rahmenbasierter Editor: Der Code wird hier in farblich voneinander abgesetzten Blöcken dargestellt und erleichtert damit die Zuordnung der einzelnen Textteile.

Ziel der Umgebung ist es, Lernende ohne Vorkenntnisse und Erfahrungen durch schnelle Erfolgserlebnisse für das Programmieren zu begeistern. Schon Anfänger erzielen rasch einfache Animationen und können anspruchsvolle, professionell aussehende Szenarien realisieren.

Dieses Buch basiert auf der Greenfoot-Version 2.4.0 und berücksichtigt die Neuerungen, die in der Zwischenzeit erschienenen Version 3.0, die sich jedoch nur in wenigen Details von der Vorgängerversion unterscheidet. Eine Sache springt jedoch sofort ins Auge (die Screenshots sind mit der neuesten Version hergestellt worden): die neue Version enthält keinen separaten Kompilier-Button mehr – dies wird jetzt automatisch vom System durchgeführt.

Zugriff auf Ergänzungsmaterial

Die Lernhilfen und das Zusatzmaterial, das im Buch erwähnt wird, sind über die Greenfoot-Website <http://www.greenfoot.org/book/> bzw. über die Website des Verlags erhältlich unter: <http://www.pearson-schule.de>. Weitere Informationen auf Deutsch (z.B. Klassendokumentationen) gibt es auf der Seite www.greenfoot-center.de.





Zusatzmaterial für Lehrkräfte

Die folgenden Zusätze sind für Lehrkräfte erhältlich:

- Ein Diskussionsforum für Lehrer
- Zusätzliche Übungen zu diesem Buch
- Zugang zum „Greenroom“, der Arbeitsblätter und anderes Lehrmaterial anbietet



Zusatzmaterial für Lernende

Die folgenden Zusätze sind für Schülerinnen und Schüler bzw. für Studierende erhältlich:

- Die Greenfoot-Software
- Die in diesem Buch besprochenen Szenarien
- Die Greenfoot Gallery – ein Schaufenster für die Szenarien
- Video-Tutorials
- Ein Diskussionsforum
- Technische Unterstützung

Weitere Details zu Greenfoot und diesem Buch kannst du der Einleitung entnehmen.



Vorwort zur deutschen Ausgabe

Greenfoot und dieses Buch laden zum Spielen ein. Es gibt Vogelschwärme zu beobachten, Wombats oder Raumschiffe zu steuern, Musik zu machen und vieles mehr. Diese vielen unterschiedlichen Szenarien und der kreative Umgang damit machen den Reiz des Buchs aus.

Würde ich gefragt, welches mein Lieblingsszenario ist, fiel mir die Entscheidung schwer. Die Krabben gefallen mir, insbesondere wie sie auf die Steuerungsbefehle reagieren. Auch die Ameisen und Vögel sind sehr interessant, wenn man sich ansieht wie sich aus einfachen programmierten Regeln derart echt wirkendes Verhalten erzeugen lässt. Das gilt auch für die Planetensimulation. Am interessantesten finde ich die Mischung aus Klavier- und Planetensimulation. Hier findet bestimmt jeder sein Lieblingsprojekt zum Spielen. Das Schöne an den Greenfoot-Szenarien ist, dass man sie zunächst einfach benutzen, dann aber auch selbst gestalten, ändern und ausbauen kann. Dieses „Herumspielen“ macht Spaß und regt zu eigenen Experimenten und Weiterentwicklungen an.

Für Schülerinnen und Schüler

Du kannst mit dem Buch selbst einen Einstieg in Greenfoot und ins Programmieren finden. Ich würde dir dann raten, die Kapitel in der angegebenen Reihenfolge durchzuarbeiten. Mit vielleicht einer Ausnahme: Willst du am Wettbewerb teilnehmen, dann sieh dir am besten vorher auch noch die weiteren Kapitel an, dort findest du ein paar wirklich hilfreiche Tipps dafür. (Ich verrate jetzt aber nicht genau, wo du sie findest.)

Du musst jedoch nicht unbedingt jede Aufgabe bearbeiten und lösen. Vielmehr kannst du selbst entscheiden, was dir Spaß macht, und gerne auch deine eigenen Szenarien gestalten. In diesem Fall kannst du das Buch als Hilfe zum Nachschlagen nutzen, denke auch an die in Greenfoot eingebauten Hilfen.

Für Lehrkräfte

Das Buch führt in die objektorientierte Programmierung ein, es richtet sich eindeutig (auch) an Anfänger, vermittelt aber auch viele Tipps für Fortgeschrittene.

Für den Einsatz im Unterricht bietet sich vermutlich an, sofort im zweiten Kapitel, d.h. dem Krabben-Szenario, zu beginnen und darauf die eigene Einführung aufzubauen oder das Buch sogar als eine Art Wochenplan zu nutzen und die Lernenden die im Text eingestreuten Aufgaben bearbeiten zu lassen. Die dabei vermittelten Techniken, Konzepte und Begriffe lassen sich dann in Auswertungsphasen sichern und vertiefen.

Die verwendete Begrifflichkeit, insbesondere auch die deutsche Übersetzung, richtet sich jedoch nicht unbedingt nach den lokalen Gepflogenheiten. Die „klassischen“ Begriffe wie Schleife oder Variable sind unstrittig. Die „neueren“ objektorientierten Konzepte werden jedoch recht unterschiedlich benannt: Instanzvariable oder Datenfeld, Zustandsfeld, Instanzvariable, Attribut bzw. Objektattribut? Wir hoffen, eher sprechende und gleichzeitig neutrale Begriffe gefunden zu haben, die im Unterricht natürlich nach Belieben um die in der jeweiligen Schule oder im jeweiligen Bundesland verwendete Terminologie ergänzt werden können.

Aus didaktischer Perspektive bietet das Buch zwei Besonderheiten: zum einen die Idee des spielerischen Herangehens, das vom Benutzen zum Gestalten leitet; zum anderen die Einführung in einen Programmierprozess.

Das Buch führt in einen Programmierprozess ein, in dem Programmieren als ein schrittweiser Problemlöseprozess aufgefasst wird. Die Vermittlung erfolgt zumeist beiläufig und spielerisch. Der Prozess wird also nicht am Stück erklärt, sondern erst nach und nach sichtbar. Dazu werden Programmieraufgaben in einzeln zu lösende Aufgaben zerlegt: In den Szenarien durch eine (vorgegebene) Klassenstruktur und indem vorhandenes Verhalten der Objekte schrittweise ergänzt wird. Das zentrale Element des Prozesses besteht im Benutzen und Erzeugen von Hilfsmethoden – wenn man möchte, die generelle Modellierungstechnik zur präzisen Beschreibung des Verhaltens. Für einen Überblick über weitere Elemente seien exemplarisch genannt: Kapitel 3.4: Methoden als Strukturierungsmittel, um den Quelltext lesbar zu halten; Kapitel 6.3: Lösungen verallgemeinern (Abstraktion); sowie Kapitel 9.5: schrittweises Implementieren einer umgangssprachlich beschriebenen Lösungsidee.

Die zweite Besonderheit ist das Benutzen, Erkunden und Umgestalten von Szenarien, welches den hauptsächlichen Zugang zu den Lerninhalten bildet. Szenarien „verstecken“ für den aktuellen Lernstand und Lernziele Unwichtiges oder zu Schwieriges bzw. geben es in Hilfsklassen vor. Die auf diese Art und Weise erarbeiteten Konzepte und Beispiele füllen schrittweise den Werkzeugkasten, der zum Bearbeiten weiterer Aufgaben und Szenarien eingesetzt wird. Dieser Ansatz versucht den kreativen Zugang zum Programmieren zu fördern, indem er dazu anregt, Lösungsmuster, die an einzelnen Szenarien erlernt wurden, in neuen Szenarien wiederzuverwenden, anzupassen und zu vertiefen. Die Szenarien bilden so eine Fundgrube für konkrete Lösungsmuster (Interaktion, Kollision, Bewegung, Animation, Sound ...). Ein Beispiel dafür ist, wie im achten Kapitel die Planetensimulation um eine musikalische Untermalung ergänzt wird. Dazu wird das zuvor eingeführte Klavier-Szenario umgestaltet und in das aktuelle Szenario eingefügt.

Da Konzepte und Techniken durchgängig am Beispiel, im Zusammenhang mit der Entwicklung oder Erweiterung eines Szenarios, erklärt werden, erscheinen sie als Techniken und Werkzeuge zur Lösung der jeweils anliegenden konkreten Aufgaben – bzw. für die spielerische Veränderung und Ausgestaltung von Szenarien. Die Aufgabe der Lehrkraft liegt daher vor allem im Verdeutlichen und Bewusstmachen dessen, was alles gelernt wurde. Nämlich nicht nur die Gestaltung eines neuen oder veränderten Szenarios, sondern auch die dazu notwendigen Programmierkonzepte, Problemlösestrategien (zerlegen, abstrahieren) sowie allgemeine und spezifische Lösungsmuster. Mit anderen Worten: Der spielerische Ansatz des Buches dient nicht nur der Motivation der Lernenden, sondern ist gleichzeitig auch methodisches Prinzip. In diesem Sinne: viel Spaß beim Spielen und Spielenlassen.

Carsten Schulte, Berlin



Danksagungen

Dieses Buch ist quasi die Spitze eines Eisbergs. Es ist eine Einführung in die Programmierung mit Java, aber diese Art des Zugangs wäre ohne das Greenfoot-Ökosystem nicht möglich. Diesem Buch liegt die jahrelange Arbeit mehrerer Leute zugrunde, die mitgeholfen haben, Greenfoot aufzubauen.

Zuallererst beruht dieses Buch auf der Software selbst – Greenfoot –, doch das ist noch nicht alles. Viel Zeit und Mühe sind in den Entwurf der Websites (die Websites der Greenfoot-Gemeinde, den Greenroom), die Entwicklung der Materialien, in den Aufbau und die Unterstützung einer Anwendergemeinde, in Workshops und Öffentlichkeitsarbeit geflossen und eine Reihe von Leuten waren maßgeblich daran beteiligt.

Poul Henriksen war die erste Person, die mich bei diesem Projekt begleitet hat. Er begann die Greenfoot-Implementierung als Teil seiner Masterarbeit und hat in den ersten Jahren den größten Beitrag zur Software geleistet. Davin McCall, Bruce Quig und Neil Brown gehören zur nächsten Entwicklergeneration, die viele Jahre an Greenfoot mitgearbeitet und große Teile des Entwurfs und der Implementierung des Systems, so wie es heute ist, geformt haben. Es ist nicht einfach, ein Softwaresystem dieser Größe mit so wenig Leuten und Ressourcen zu warten, doch alle sind herausragende Programmierer und haben es geschafft, ein System zu entwickeln, das seit fast zehn Jahren überlebt hat und mit wenigen Problemen auf Millionen von Rechnern in der ganzen Welt läuft. Dies ist eine außergewöhnliche Leistung und ich bin sehr glücklich, diese Leute in meinem Team zu haben.

Andere wichtige Beiträge zum Ökosystem kamen von Ian Utting sowie unseren neueren Teammitgliedern Amjad Altadmri und Fabio Hedayioglu. Eine breite Palette an Aktivitäten hat geholfen, die Greenfoot-Gemeinde zu dem zu machen, was sie heute ist.

Die Entwicklung von Greenfoot wurde von Oracle Inc. durch gemeinnützige Spenden über viele Jahre hinweg unterstützt. Diese anhaltende Förderung hat es uns ermöglicht, unsere Gruppe zu erhalten – ohne diese gäbe es Greenfoot nicht. Wir sind sehr dankbar für das Engagement, das einen wesentlichen Beitrag zur Bildungsgemeinschaft leistet.

Die Leute bei Pearson Education haben sich tapfer geschlagen – angesichts der vielen Verzögerungen, die ich verursacht habe, indem ich jeden möglichen Abgabetermin zum Einsenden des Manuskripts verpasst habe. Tracy Johnson hat mit mir an diesem Buch von Anfang an gearbeitet und mich auch nun bei dieser zweiten

Auflage begleitet. Sie war ständig positiv, enthusiastisch und aufmunternd und ihre Unterstützung hat viel ausgemacht. Camille Trentacoste und Carole Snyder haben wichtige Kleinarbeit geleistet, um dieses Buch herzustellen, und ich bin dankbar für ihren Beitrag und ihre Hilfe.

Die erste Auflage wurde von einer Reihe Personen fachlich durchgesehen, deren viele ausführliche, wohlüberlegte und nützliche Kommentare mir eine große Hilfe waren. Zu ihnen gehören Carolyn Oates, Damianne President, Detlef Rick, Gunnar Johannesmeyer, Josh Fishburn, Mark Hayes, Marla Parker, Matt Jadud, Todd O'Bryan, Lael Grant, Jason Green, Mark Lewis, Rodney Hoffman und Michael Kadri. Sie alle haben viele Verbesserungsvorschläge gemacht und dazu beigetragen, auch die verborgensten Fehler zu finden. Josh Buhl und Adrienne Decker haben viele hilfreiche Vorschläge nach der Veröffentlichung der ersten Auflage eingebracht, die geholfen haben, die Beispiele für die zweite Auflage zu verbessern.

Ein großer Dank gebührt auch Kerstin Wachholz für ihr exzellentes Korrekturlesen – sie hat viele meiner Fehler gefunden und meine Sprachmängel berichtigt – und meinem guten Freund, Michael Caspersen, der mich von Anfang bestätigt hat, was sehr wichtig für mich war – zum Teil, weil es zur Verbesserung des Buches beitrug, aber mehr noch, weil es mich ermutigte, daran zu glauben, dass die Idee des Greenfoot-Systems für Lehrer interessant sein könnte und auf alle Fälle beendet werden sollte.



Vorwort zur 2. Auflage

Dies ist die zweite Auflage des Buches. Wir haben versucht, an dem festzuhalten, was beim ersten Mal gut funktioniert hat, und die Teile zu verbessern, die nicht so leichtgängig waren, wie sie hätten sein können.

Den Gesamtstil des Buches behalten wir bei: die interaktive Präsentation der Programmierprojekte, die praktischen Aufgaben eingestreut in die Diskussion und die Erläuterungen sowie den allgemeinen Ton. Das hat sehr gut funktioniert.

Es gab jedoch auch Punkte in der ersten Auflage, an denen Leser in der zweiten Hälfte des Buches den Fortschritt als recht herausfordernd empfanden. Deshalb haben wir jetzt zwei weitere Kapitel eingefügt, um einige Konzepte langsamer und schrittweise einzuführen und um mehr Praxis mit den schwierigeren Konzepten zu bekommen. Wir haben auch jedem Kapitel eine bedeutende Anzahl an Aufgaben hinzugefügt, um die behandelten Konzepte mehr einzuüben und zu festigen. Dazu gehört auch, dass wir jetzt viel mehr Praxis-Szenarien darstellen und verwenden.

Außerdem ist ein Kapitel zur Greenfoot-Programmierung mit der Microsoft Kinect hinzugekommen. Obwohl es nicht jeder Leser benutzen kann (weil man dazu die entsprechende Hardware zur Verfügung haben muss), glauben wir, dass der Enthusiasmus und die Begeisterung, die unsere Beispiele hervorgerufen haben, wenn wir sie auf Workshops vorstellten, die Aufnahme rechtfertigen. Hier steckt so viel Potenzial!

Und natürlich wurde das Buch aktualisiert, um neue Funktionen von aktuelleren Versionen der Greenfoot-Software einzubeziehen. Wir haben Greenfoot angepasst, um einige beliebte Aufgaben möglich oder einfacher zu machen und um einige Konzepte besser zu illustrieren. Im Buch findet sich dies in den neuen Szenarien wieder.

Insgesamt hoffen wir, dass das hinzugekommene Material dir hilft, deinen Pfad durch das Labyrinth – das Lernen zu programmieren – noch reibungsloser und interessanter zu gestalten.

Einleitung



Willkommen zu Greenfoot! In diesem Buch wollen wir dir zeigen, wie du mithilfe der Greenfoot-Umgebung und der Programmiersprache Java Simulationen, Spiele und andere grafische Computerprogramme entwickeln kannst.

Unser primäres Ziel ist es, dir das Programmieren beizubringen. Fast ebenso wichtig ist uns aber auch, dir den Spaß am Programmieren zu vermitteln. Während die Beispiele, die wir in diesem Buch besprechen, eindeutig auf die Greenfoot-Umgebung zugeschnitten sind, sind die zugrunde liegenden Konzepte allgemeingültiger Natur. Wenn du dieses Buch durchgearbeitet hast, bist du mit den allgemeinen Grundsätzen einer modernen, objektorientierten Programmiersprache vertraut. Gleichzeitig lernst du, wie du dein eigenes Computerspiel, ein Bildschirm-Klavier oder eine Simulation eines biologischen Systems erstellst.

Dieses Buch ist ausgesprochen praxisorientiert. Die Kapitel und Übungen stellen immer konkrete, praktische Aufgaben in den Mittelpunkt. Zuerst gibt es ein Problem, das wir lösen müssen, dann beschäftigen wir uns mit den Sprachkonstrukten und Programmierstrategien, mit denen wir das Problem lösen können. Dieser Ansatz unterscheidet sich grundlegend von dem vieler anderer Bücher zur Einführung in die Programmierung, deren Aufbau sich an den Sprachkonstrukten orientiert.

Folglich ist dieses Buch am Anfang nicht so theorielastig wie die meisten anderen Programmierbücher, sondern wartet gleich mit vielen praktischen Beispielen auf. Vielleicht fragst du dich, warum wir Greenfoot verwenden. Die Antwort lautet, weil erst die Greenfoot-Umgebung diesen didaktischen Ansatz möglich macht. Greenfoot erlaubt uns zu spielen. Und das bedeutet nicht, Computerspiele zu spielen. Es bedeutet vielmehr, spielerisch programmieren zu lernen. Wir können – bequem und interaktiv – Objekte erzeugen, sie auf dem Bildschirm verschieben, ihre Methoden aufrufen und beobachten, was sie machen. Dadurch fällt der Einstieg in die Programmierung leichter, als dies ohne eine solche Umgebung der Fall wäre.

Ein praxisorientierter Ansatz bedeutet aber nicht, dass dieses Buch auf die Erläuterung der notwendigen Theorie und Prinzipien verzichtet. Lediglich die Reihenfolge wurde geändert. Anstatt ein Konzept zuerst theoretisch einzuführen und dann durch Übungen zu festigen, wenden wir das Konstrukt gleich an. Dabei erklären wir nur so viel, wie für die Lösung der vorliegenden Aufgabe unbedingt nötig ist; auf den theoretischen Hintergrund gehen wir erst später ein. Didaktisch verfolgen wir häufig einen Spiralansatz: Wir führen einige Aspekte eines Konzeptes ein, wenn wir uns das erste Mal damit auseinandersetzen, kommen später in einem anderen Kontext darauf zurück und vertiefen auf diese Weise nach und nach unser Verständnis.

Der Schwerpunkt dieses Buches liegt vor allem darauf, dem Leser zu zeigen, wie interessant, kurzweilig und wichtig die Arbeit eines Programmierers sein kann. Es gibt wirklich keinen Grund, warum Computerprogrammierung trocken, formal oder langweilig sein muss. Spaß beim Programmieren ist durchaus in Ordnung. Wir glauben, dass es uns gelungen ist, den Lehrstoff interessant aufzubereiten und auf pädagogisch solider Basis zu vermitteln.

Dieses Buch eignet sich zum Selbststudium, kann aber auch in Informatikkursen eingesetzt werden. Die Übungen sind über das ganze Buch verstreut. Wenn du sie alle durchgearbeitet hast, wirst du am Ende über ansehnliche Programmierkenntnisse verfügen.

Die in diesem Buch besprochenen Übungen sind einfach genug, um von Schülern bewältigt zu werden, lassen andererseits jedoch genug Spielraum für Erweiterungen, die auch für erfahrene Programmierer interessant sein dürften und eine Herausforderung darstellen. Greenfoot ist zwar eine Lernumgebung, aber Java keine Programmiersprache für Kinder. Da wir für dieses Buch Java als Programmiersprache gewählt haben, liegt es in deiner Hand, wie komplex und schwierig die hier vorgestellten Projekte – und die, die du noch in Greenfoot entwickeln möchtest – sind.

Zwar kann man mit Greenfoot schnell und bequem einfache Spiele erstellen, doch ist es genauso gut möglich, sehr anspruchsvolle Simulationen von komplexen Systemen zu entwickeln, die Algorithmen der künstlichen Intelligenz, Agententechnologie, Datenbankkonnektivität, Netzwerkkommunikation usw. einsetzen. Java ist eine sehr umfangreiche Sprache, die dir die ganze Welt der Programmierung erschließt, und Greenfoot wird dir mit Sicherheit hinsichtlich der Sprachelemente, die du verwendest, keine Beschränkungen auferlegen.

In anderen Worten, Greenfoot lässt sich gut skalieren¹. Es erleichtert jungen Anfängern den Einstieg in die Programmierung, erlaubt aber gleichzeitig erfahrenen Programmierern die Implementierung interessanter, anspruchsvoller Szenarien.

Programmieren ist eine kreative Disziplin und Greenfoot ist ein Werkzeug, das dich dabei unterstützt, deine Ideen umzusetzen.

1 Skalieren (von *scale*: Maßstab; und *to scale*: die Größe ändern) bedeutet in der Informatik, dass die Software nicht nur für die derzeitige Größe des zu bearbeitenden Problems geeignet ist, sondern auch für größere und komplexere Aufgaben.