

## Programming Pearls 2nd Edition

As recognized, adventure as well as experience approximately lesson, amusement, as capably as concord can be gotten by just checking out a book **Programming Pearls 2nd Edition** afterward it is not directly done, you could assume even more on the subject of this life, a propos the world.

We offer you this proper as with ease as easy pretension to acquire those all. We meet the expense of Programming Pearls 2nd Edition and numerous ebook collections from fictions to scientific research in any way. accompanied by them is this Programming Pearls 2nd Edition that can be your partner.

**The Art of UNIX Programming** Eric S. Raymond 2003-09-23 The Art of UNIX Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX philosophy and practice through careful case studies of the very best UNIX/Linux programs.

**C++ Coding Standards** Herb Sutter 2004-10-25 Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards. The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized—techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like What's worth standardizing—and what isn't? What are the best ways to code for scalability? What are the elements of a rational error handling policy? How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies? When (and how) should you use static and dynamic polymorphism together? How do you practice "safe" overriding? When should you provide a no-fail swap? Why and how should you prevent exceptions from propagating across module boundaries? Why shouldn't you write namespace declarations or directives in a header file? Why should you use STL vector and string

instead of arrays? How do you choose the right STL search or sort algorithm? What rules should you follow to ensure type-safe code? Whether you're working alone or with others, C++ Coding Standards will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

**The Art of Readable Code** Dustin Boswell 2011-11-03 Chapter 5. Knowing What to Comment; What NOT to Comment; Don't Comment Just for the Sake of Commenting; Don't Comment Bad Names--Fix the Names Instead; Recording Your Thoughts; Include "Director Commentary"; Comment the Flaws in Your Code; Comment on Your Constants; Put Yourself in the Reader's Shoes; Anticipating Likely Questions; Advertising Likely Pitfalls; "Big Picture" Comments; Summary Comments; Final Thoughts--Getting Over Writer's Block; Summary; Chapter 6. Making Comments Precise and Compact; Keep Comments Compact; Avoid Ambiguous Pronouns; Polish Sloppy Sentences.

**Implementation Patterns - Studentenausgabe** Kent Beck 2010

*Code Complete - Deutsche Ausgabe* Steve McConnell 2005-01 Dieses Buch ist die deutsche Übersetzung eines Klassikers der Programmierliteratur von Steve McConnell. Seine mit vielen Preisen ausgezeichneten Bücher helfen Programmierern seit Jahren, besseren und effizienteren Code zu schreiben. Das Geheimnis dieses Buches liegt in der Art, wie der Autor das vorhandene Wissen über Programmierertechniken aus wissenschaftlichen Quellen mit den Erfahrungen aus der taglichen praktischen Arbeit am Code zusammenführt und daraus die wesentlichen Grundvoraussetzungen der Softwareentwicklung und die effektivsten Arbeitstechniken ableitet. Verständliche Beispiele und klare Anleitungen vermitteln dem Leser dieses Wissen auf unkomplizierte Weise. Dieses Buch informiert und stimuliert, ganz gleich, wie viel sie bereits über Programmierung wissen, welche Entwicklungsumgebung und Sprache sie bevorzugen und welche Arten von Anwendungen sie normalerweise programmieren.

**Algorithmic Puzzles** Anany Levitin 2011-10-12 While many think of algorithms as specific to computer science, at its core algorithmic thinking is defined by the use of analytical logic to solve problems. This logic extends far beyond the realm of computer science and into the wide and entertaining world of puzzles. In *Algorithmic Puzzles*, Anany and Maria Levitin use many classic brainteasers as well as newer examples from job interviews with major corporations to show readers how to apply analytical thinking to solve puzzles requiring well-defined procedures. The book's unique collection of puzzles is supplemented with carefully developed tutorials on algorithm design strategies and analysis techniques intended to walk the reader step-by-step through the various approaches to algorithmic problem solving. Mastery of these strategies--exhaustive search, backtracking, and divide-and-conquer, among others--will aid the reader in solving not only the puzzles contained in this book, but also others encountered in interviews, puzzle collections, and throughout everyday life. Each of the 150 puzzles contains hints and solutions, along with commentary on the puzzle's origins and solution methods. The only book of its kind, *Algorithmic Puzzles* houses puzzles for all skill levels. Readers with only middle school mathematics will develop their algorithmic problem-solving skills through puzzles at the elementary level, while seasoned puzzle solvers will enjoy the challenge of thinking through more difficult puzzles.

**Software Development, Design and Coding** John F. Dooley 2017-11-25 Learn the principles of good software design, and how to turn those principles into

great code. This book introduces you to software engineering – from the application of engineering principles to the development of software. You'll see how to run a software development project, examine the different phases of a project, and learn how to design and implement programs that solve specific problems. It's also about code construction – how to write great programs and make them work. Whether you're new to programming or have written hundreds of applications, in this book you'll re-examine what you already do, and you'll investigate ways to improve. Using the Java language, you'll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programs. With *Software Development, Design and Coding*, author and professor John Dooley distills his years of teaching and development experience to demonstrate practical techniques for great coding. What You'll Learn Review modern agile methodologies including Scrum and Lean programming Leverage the capabilities of modern computer systems with parallel programming Work with design patterns to exploit application development best practices Use modern tools for development, collaboration, and source code controls Who This Book Is For Early career software developers, or upper-level students in software engineering courses

**Linux in a nutshell** Ellen Siever 2005

Software Development and Professional Practice John Dooley 2011-10-13  
*Software Development and Professional Practice* reveals how to design and code great software. What factors do you take into account? What makes a good design? What methods and processes are out there for designing software? Is designing small programs different than designing large ones? How can you tell a good design from a bad one? You'll learn the principles of good software design, and how to turn those principles back into great code. *Software Development and Professional Practice* is also about code construction—how to write great programs and make them work. What, you say? You've already written eight gazillion programs! Of course I know how to write code! Well, in this book you'll re-examine what you already do, and you'll investigate ways to improve. Using the Java language, you'll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programs. You'll also talk about reading code. How do you read code? What makes a program readable? Can good, readable code replace documentation? How much documentation do you really need? This book introduces you to software engineering—the application of engineering principles to the development of software. What are these engineering principles? First, all engineering efforts follow a defined process. So, you'll be spending a bit of time talking about how you run a software development project and the different phases of a project. Secondly, all engineering work has a basis in the application of science and mathematics to real-world problems. And so does software development! You'll therefore take the time to examine how to design and implement programs that solve specific problems. Finally, this book is also about human-computer interaction and user interface design issues. A poor user interface can ruin any desire to actually use a program; in this book, you'll figure out why and how to avoid those errors. *Software Development and Professional Practice* covers many of the topics described for the ACM Computing Curricula 2001 course C292c *Software Development and Professional Practice*. It is

designed to be both a textbook and a manual for the working professional.

*Perl-Kochbuch* Tom Christiansen 2004

*Programming Pearls* Jon Louis Bentley 2000 A guide to practical programming techniques and design principles, with information on such topics as testing, debugging and timing, set representations, and string problems.

*Coders at Work* Peter Seibel 2011

**EBOOK: Object-Oriented Software Engineering: Practical Software Development Using UML and Java** LETHBRIDGE, TIM 2004-12-16 EBOOK: Object-Oriented Software Engineering: Practical Software Development Using UML and Java

**Effektives Arbeiten mit Legacy Code** Michael C. Feathers 2020-11-04 Können Sie Ihren Code leicht ändern? Können Sie fast unmittelbar Feedback bekommen, wenn Sie ihn ändern? Verstehen Sie ihn? Wenn Sie eine dieser Fragen mit nein beantworten, arbeiten Sie mit Legacy Code, der Geld und wertvolle Entwicklungszeit kostet. Michael Feathers erläutert in diesem Buch Strategien für den gesamten Entwicklungsprozess, um effizient mit großen, ungetesteten Code-Basen zu arbeiten. Dabei greift er auf erprobtes Material zurück, das er für seine angesehenen Object-Mentor-Seminare entwickelt hat. Damit hat er bereits zahlreichen Entwicklern, technischen Managern und Testern geholfen, ihre Legacy-Systeme unter Kontrolle zu bringen. Darüber hinaus finden Sie auch einen Katalog mit 24 Techniken zur Aufhebung von Dependencies, die Ihnen zeigen, wie Sie isoliert mit Programmelementen arbeiten und Code sicherer ändern können.

Puzzles for Hackers Ivan Sklyarov 2005 These puzzles and mind-benders serve as a way to train logic and help developers, hackers, and system administrators discover unconventional solutions to common IT problems. Users will learn to find bugs in source code, write exploits, and solve nonstandard coding tasks and hacker puzzles. Cryptographic puzzles, puzzles for Linux and Windows hackers, coding puzzles, and puzzles for web designers are included.

*Praktische C++-Programmierung* Steve Oualline 2004

*Struktur und Interpretation von Computerprogrammen* Harold Abelson 1991 Diese moderne Einführung in die Informatik ist am renommierten Massachusetts Institute of Technology entstanden und repräsentiert den dortigen Ausbildungsstandard für Studenten der Informatik und der Elektrotechnik. Das ganzheitliche Verständnis der Informatik unter Einbeziehung der Künstlichen Intelligenz, das in diesem Buch vermittelt wird, hat es weltweit, und insbesondere auch im deutschsprachigen Bereich, bereits in der englischen Originalfassung zu einer beliebten Grundlage für die Einführungsvorlesung gemacht. Zur Notation der Programme wird Scheme verwendet, ein Dialekt der Programmiersprache Lisp, der die Leistungsfähigkeit und die Eleganz von Lisp und Algol verbindet. Die Besonderheit dieser einführenden Vorlesung beruht auf zwei Grundüberzeugungen: 1. Eine Computersprache ist nicht einfach ein Weg, einen Computer zur Ausführung von Operationen zu bewegen, sondern vielmehr ein neuartiges Medium, um Vorstellungen über Verfahrensweisen auszudrücken. So müssen Programme geschrieben werden, damit Menschen sie lesen und modifizieren, und nur nebenbei, damit Maschinen sie ausführen können. 2. Das Wesentliche bei einer Vorlesung auf diesem Niveau ist weder die Syntax von speziellen Sprachkonstruktionen, noch sind es raffinierte Algorithmen zur effizienten Berechnung bestimmter Funktionen oder gar die mathematische Analyse von Algorithmen oder die Grundlagen der Informatik,

sondern vielmehr die Techniken, mit denen die geistige Komplexität großer Softwaresysteme unter Kontrolle gehalten werden kann.

*10 Spiele, die du in deinem Kopf spielen kannst* Sam Gorski 2019-10-14  
Fantasie ist wie Zocken - nur krasser. Ob High-End-Computer, Virtual-Reality-Brillen oder Exit-Räume: Spiele werden immer ausgefallener und aufwendiger. Doch die besten Spiele kann man ohne Controller, ohne Stifte und sogar ohne Mitspieler spielen. Man braucht nur sich selbst - und dieses Buch. Es enthält zehn großartige Gedankenspiele, in denen du die Charaktere, die Spielwelt und alle kleinen Details selbst zu steuern lernst. Was genau in den Spielen passieren wird? Das musst du schon selbst herausfinden. ++  
Ausgangssperre, Homeoffice, keine sozialen Kontakte? So furchtbar die aktuelle Corona-Krise ist, sie kann auch wahnsinnig langweilig sein. Dieses Buch ist eine prima Beschäftigung und lenkt dich in Zeiten der Isolation ein wenig ab. Bleib gesund! ++

Expert-C-Programmierung Peter Van der Linden 1995

**Einführung in Perl** Randal L. Schwartz 2012 Perl ist eine Skriptsprache zur einfachen Bearbeitung von Texten, Dateien und Prozessen. Ursprünglich ein beliebtes Werkzeug von Unix-Systemadministratoren für die zahllosen alltäglichen Aufgaben hat sich Perl zu einer ausgewachsenen Programmiersprache für nahezu jede Rechnerplattform entwickelt und wird für Web- und Datenbank-Programmierung, XML-Verarbeitung, Systemadministration und vieles mehr eingesetzt. Das Schweizer Messer der Programmiersprachen  
Gleichzeitig ist Perl immer noch das Schweizer Messer für die kleinen alltäglichen Aufgaben. Perl ist schnell, macht Spaß und erweist sich als außerordentlich nützlich. Viele haben Perl gelernt, weil sie mussten, und benutzen es weiter, weil sie es lieben. Für Einsteiger Einführung in Perl ist ein sorgfältig abgestimmter Kurs für Einsteiger von drei der erfahrensten Perl-Dozenten. Mit vielen Programmierbeispielen sowie Übungen und ausgearbeiteten Lösungen zu jedem Thema zeigen die Autoren Schritt für Schritt, wie man mit Perl, Version 5.14, programmiert. Ideal für Systemadministratoren und Programmierer Einführung in Perl ist das ideale Buch für Systemadministratoren und Programmierer, die schon nach kurzer Zeit einsetzbare Perl-Skripte schreiben wollen.

**Perlen der Programmierkunst.** Jon Louis Bentley 2000-01

*Algorithm Engineering* Gerd Stoelting Brodal 2003-06-30 This book constitutes the refereed proceedings of the 5th Workshop on Algorithm Engineering, WAE 2001, held in Aarhus, Denmark, in August 2001. The 15 revised full papers presented were carefully reviewed and selected from 25 submissions. Among the topics addressed are implementation, experimental testing, and fine-tuning of discrete algorithms; novel use of discrete algorithms in other disciplines; empirical research on algorithms and data structures; and methodological issues regarding the process of converting user requirements into efficient algorithmic solutions and implementations.

**Cocoa** Aaron Hillegass 2012 Diese deutsche Übersetzung des Bestsellers von Aaron Hillegass ist das Standardwerk zur Mac-Programmierung. Hillegass behandelt alle Grundlagen, die Sie zur Programmierung für den Mac mit Cocoa brauchen, um featurereiche Anwendungen für OS X zu entwickeln. Das Buch ist eine wertvolle Ressource für jeden Mac-Programmierer!

*The Art of Unit Testing* Roy Osherove 2015-02-15 Lesbare, wartbare und zuverlässige Tests entwickeln Stubs, Mock-Objekte und automatisierte

FrameworksEinsatz von .NET-Tools inkl. NUnit, Rhino Mocks und Typemock Isolator Unit Testing, richtig durchgeführt, kann den Unterschied ausmachen zwischen einem fehlgeschlagenen Projekt und einem erfolgreichen, zwischen einer wartbaren Code-Basis und einer, die niemand freiwillig anpackt, zwischen dem Nach-Hause-Kommen um 2 Uhr nachts oder zum Abendessen, selbst noch kurz vor dem Release-Termin. Roy Osherove führt Sie Schritt für Schritt von einfachen Tests zu Tests, die wartbar, lesbar und zuverlässig sind. Er geht danach auf die Grundlagen des Interaction Testings ein und stellt schließlich bewährte Vorgehensweisen für das Schreiben, das Verwalten und das Warten der Unit Tests in echten Projekten vor. Darüber hinaus werden auch fortgeschrittene Themen behandelt wie Mocks, Stubs und Frameworks wie etwa Typemock Isolator und Rhino Mocks. Sie werden eine Menge zu fortgeschrittenen Testmustern und zur Testorganisation, zum Arbeiten mit Legacy Code und auch zu untestbarem Code erfahren. Und Sie lernen Werkzeuge kennen, die Sie beim Testen von Datenbanken und anderen Technologien brauchen werden. Alle Beispiele sind mit Visual Studio in C# geschrieben, so dass die Beispiele insbesondere für .NET-Entwickler nützlich sind. Aber auch für Programmierer anderer Sprachen wird das Buch von großem Nutzen sein, da die Prinzipien des Unit Testings für andere Sprachen dieselben sind. Roys Blog finden Sie auf ISerializable.com. Aus dem Inhalt: Verwenden eines Test-Frameworks (NUnit)Grundlegende TestattributeStubs zum Auflösen von AbhängigkeitenInteraction Testing mit Mock-ObjektenTesthierarchie und OrganisationDie Säulen guter TestsIntegration von Unit Testing in das UnternehmenUmgang mit Legacy Code

Algorithms on Trees and Graphs Gabriel Valiente 2013-04-17 Graph algorithms is a well-established subject in mathematics and computer science. Beyond classical application fields, such as approximation, combinatorial optimization, graphics, and operations research, graph algorithms have recently attracted increased attention from computational molecular biology and computational chemistry. Centered around the fundamental issue of graph isomorphism, this text goes beyond classical graph problems of shortest paths, spanning trees, flows in networks, and matchings in bipartite graphs. Advanced algorithmic results and techniques of practical relevance are presented in a coherent and consolidated way. This book introduces graph algorithms on an intuitive basis followed by a detailed exposition in a literate programming style, with correctness proofs as well as worst-case analyses. Furthermore, full C++ implementations of all algorithms presented are given using the LEDA library of efficient data structures and algorithms.

Effektiv C++ programmieren Scott Meyers 2011

*Introductory Programming with Simple Games* Brian C. Ladd 2010-03-29 This is an excellent resource for programmers who need to learn Java but aren't interested in just reading about concepts. Introduction to Java Programming with Games follows a spiral approach to introduce concepts and enable them to write game programs as soon as they start. It includes code examples and problems that are easy to understand and motivates them to work through to find the solutions. This game-motivated presentation will help programmers quickly apply what they've learned in order to build their skills.

*Algorithm Engineering* Denmark) Wae 200 (2001 Aarhus 2001-08-15 Compact DFA representation for fast regular expression search / Gonzalo Navarro / - The



reviewed by at least four program committee members (assisted by subreferees). A special issue of the ACM Journal of Experimental Algorithmics will be devoted to selected papers from WAE 2000.

Das Lied des Blutes Anthony Ryan 2014-09-03 Vaelin Al Sorna, der berühmteste Gefangene des Reichs und sein größter Kämpfer, erzählt die atemberaubende Geschichte seines Lebens. Er ist an Bord eines Schiffes, das ihn zu dem Ort bringen soll, an dem es für ihn um Leben und Tod geht. Einst war er von seinem Vater als Zögling in das Kloster des Sechsten Ordens gebracht worden, wo er zum Krieger ausgebildet wurde. Nun sind die Namen, die er sich im Kampf verdient hat, über alle Grenzen hin bekannt: Schwert des Königs, Dunkelklinge, Rabenschatten und am gefürchtetsten: Hoffnungstöter.

*Programmieren in C* Robert Klima 2010-12-28 C ist eine der bedeutendsten und eine sehr häufig eingesetzte Programmiersprache. Die Autoren haben jahrelange Erfahrung mit dieser Programmiersprache und vermitteln Lesern das Wesentliche - die Programmiermethodik: Was ist Programmieren? Wie werden programmtechnische Probleme gelöst? Schrittweise wird die Programmierung anhand der Sprache C erlernt und mit Beispielen und Aufgaben vertieft. Der Sprachumfang von C wird vorgestellt und kritisch betrachtet, um typische Fehler zu vermeiden. Vorkenntnisse im Programmieren werden nicht vorausgesetzt.

*R Programming for Bioinformatics* Robert Gentleman 2008-07-14 Due to its data handling and modeling capabilities as well as its flexibility, R is becoming the most widely used software in bioinformatics. R Programming for Bioinformatics explores the programming skills needed to use this software tool for the solution of bioinformatics and computational biology problems. Drawing on the author's first-hand experiences as an expert in R, the book begins with coverage on the general properties of the R language, several unique programming aspects of R, and object-oriented programming in R. It presents methods for data input and output as well as database interactions. The author also examines different facets of string handling and manipulations, discusses the interfacing of R with other languages, and describes how to write software packages. He concludes with a discussion on the debugging and profiling of R code. With numerous examples and exercises, this practical guide focuses on developing R programming skills in order to tackle problems encountered in bioinformatics and computational biology.

**Sieben Wochen, sieben Sprachen (Prags)** Bruce A. Tate 2011-06-30 Mit diesen sieben Sprachen erkunden Sie die wichtigsten Programmiermodelle unserer Zeit. Lernen Sie die dynamische Typisierung kennen, die Ruby, Python und Perl so flexibel und verlockend macht. Lernen Sie das Prototyp-System verstehen, das das Herzstück von JavaScript bildet. Erfahren Sie, wie das Pattern Matching in Prolog die Entwicklung von Scala und Erlang beeinflusst hat. Entdecken Sie, wie sich die rein funktionale Programmierung in Haskell von der Lisp-Sprachfamilie, inklusive Clojure, unterscheidet. Erkunden Sie die parallelen Techniken, die das Rückgrat der nächsten Generation von Internet-Anwendungen bilden werden. Finden Sie heraus, wie man Erlangs "Lass es abstürzen"-Philosophie zum Aufbau fehlertoleranter Systeme nutzt. Lernen Sie das Akteur-Modell kennen, das das parallele Design bei Io und Scala bestimmt. Entdecken Sie, wie Clojure die Versionierung nutzt, um einige der schwierigsten Probleme der Nebenläufigkeit zu lösen. Hier finden Sie alles



in einem Buch. Nutzen Sie die Konzepte einer Sprache, um kreative Lösungen in einer anderen Programmiersprache zu finden – oder entdecken Sie einfach eine Sprache, die Sie bisher nicht kannten. Man kann nie wissen – vielleicht wird sie sogar eines ihrer neuen Lieblingswerkzeuge.

**Reguläre Ausdrücke Kochbuch** Jan Goyvaerts 2010 Für Entwickler, die regelmäßig mit Texten arbeiten, sind reguläre Ausdrücke so lebensnotwendig wie die Luft zum Atmen. Doch wer sich nur oberflächlich mit diesem Hilfsmittel auskennt, gerät leicht in unangenehme Situationen. Selbst erfahrene Programmierer haben immer wieder mit schlechter Performance, falsch positiven oder falsch negativen Ergebnissen und unerklärlichen Fehlern zu kämpfen. Dieses Kochbuch schafft Abhilfe: Anhand von über 100 Rezepten für C#, Java, JavaScript, Perl, PHP, Python, Ruby und VB.NET lernen Sie, wie Sie reguläre Ausdrücke gekonnt einsetzen, typische Fallen umgehen und so viel wertvolle Zeit sparen. Mit Tutorial für Anfänger: Falls Sie noch nicht – oder nur wenig – mit regulären Ausdrücken gearbeitet haben, dienen Ihnen die ersten Kapitel dieses Buchs als Tutorial, das Sie mit den Grundlagen der Regexes und empfehlenswerten Tools vertraut macht. So sind Sie für die komplexeren Beispiele in den darauf folgenden Kapiteln bestens gerüstet. Tricks und Ideen für Profis: Auch erfahrene Regex-Anwender kommen ganz auf ihre Kosten: Jan Goyvaerts und Steven Levithan, zwei anerkannte Größen im Bereich reguläre Ausdrücke, gewahren tiefe Einblicke in ihren Erfahrungsschatz und überraschen mit eleganten Lösungen für fast jede denkbare Herausforderung. Deckt die unterschiedlichen Programmiersprachen ab: In allen Rezepten werden Regex-Optionen sowie Varianten für die verschiedenen Programmier- und Skriptsprachen aufgezeigt. Damit lassen sich sprachenspezifische Bugs sicher vermeiden.

*Effective Awk Programming* Arnold Robbins 2015-03-03 This practical guide serves as both a reference and tutorial for POSIX-standard awk and for the GNU implementation, called gawk . This book is useful for novices and awk experts alike. In this thoroughly revised edition, author and gawk lead developer Arnold Robbins describes the awk language and gawk program in detail, shows you how to use awk and gawk for problem solving, and then dives into specific features of gawk .

**Understanding the Discrete Element Method** Hans-Georg Matuttis 2014-05-12 Gives readers a more thorough understanding of DEM and equips researchers for independent work and an ability to judge methods related to simulation of polygonal particles Introduces DEM from the fundamental concepts (theoretical mechanics and solidstate physics), with 2D and 3D simulation methods for polygonal particles Provides the fundamentals of coding discrete element method (DEM) requiring little advance knowledge of granular matter or numerical simulation Highlights the numerical tricks and pitfalls that are usually only realized after years of experience, with relevant simple experiments as applications Presents a logical approach starting with the mechanical and physical bases, followed by a description of the techniques and finally their applications Written by a key author presenting ideas on how to model the dynamics of angular particles using polygons and polyhedral Accompanying website includes MATLAB-Programs providing the simulation code for two-dimensional polygons Recommended for researchers and graduate students who deal with particle models in areas such as fluid dynamics, multi-body engineering, finite-element methods, the geosciences, and multi-

scale physics.

*programming-pearls-2nd-edition*

*Downloaded from [sendy.burda.ro](https://sendy.burda.ro) on October 2,  
2022 by guest*