

<<Java程序设计>>

图书基本信息

书名：<<Java程序设计>>

13位ISBN编号：9787302198000

10位ISBN编号：7302198004

出版时间：2009-5

出版时间：清华大学出版社

作者：Robert Sedgewick

页数：693

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

THE BASIS FOR EDUCATION IN THE last millennium was "reading, writing, and arithmetic;" now it is reading, writing, and computing. Learning to program is an essential part of the education of every student in the sciences and engineering. Beyond direct applications, it is the first step in understanding the nature of computer sciences undeniable impact on the modern world. This book aims to teach programming to those who need or want to learn it, in a scientific context. Our primary goal is to empower students by supplying the experience and basic tools necessary to use computation effectively. Our approach is to teach students that writing a program is a natural, satisfying, and creative experience (not an onerous task reserved for experts). We progressively introduce essential concepts, embrace classic applications from applied mathematics and the sciences to illustrate the concepts, and provide opportunities for students to write programs to solve engaging problems. We use the Java programming language for all of the programs in this book—we refer to Java after programming in the title to emphasize the idea that the book is about fundamental concepts in programming, not Java per se. This book teaches basic skills for computational problem-solving that are applicable in many modern computing environments, and is a self-contained treatment intended for people with no previous experience in programming. This book is an interdisciplinary approach to the traditional CSI curriculum, where we highlight the role of computing in other disciplines, from materials science to genomics to astrophysics to network systems. This approach emphasizes for students the essential idea that mathematics, science, engineering, and computing are intertwined in the modern world. While it is a CSI textbook designed for any first-year college student interested in mathematics, science, or engineering (including computer science), the book also can be used for self-study or as a supplement in a course that integrates programming with another field. III

Coverage The book is organized around four stages of learning to program : ba. 1 basic elements, functions, object-oriented programming, and algorithms (with data structures). We provide the basic information readers need to build confidence in writing programs at each level before moving to the next level. An essential feature of our approach is to use example programs that solve intriguing problems, supported with exercises ranging from self-study drills to challenging problems that call for creative solutions. Basic elements include variables, assignment statements, built-in types of data, flow of control (conditionals and loops), arrays, and input/output, including graphics and sound. Functions and modules are the students first exposure to modular programming. We build familiarity with mathematical functions to introduce Java static methods, and then consider the implications of programming with functions, including libraries of functions and recursion. We stress the fundamental idea of dividing a program into components that can be independently debugged, a program into components that can be independently debugged, maintained, and reused.

Object-oriented Programming is our introduction to data abstraction. We emphasize the concepts of a data type (a set of values and a set of operations on them) and an object (an entity that holds a data-type value) and their implementation (an entity that holds a data-type value) and their implementation using Java's class mechanism. We teach students how to use, create, and design data types. Modularity, encapsulation, and other modern programming paradigms are the central concepts of this stage. Algorithms and data structures combine these modern programming paradigms with classic methods of organizing and processing data that remain effective for modern applications. We provide an introduction to classical algorithms for sorting and searching as

<<Java程序设计>>

well as fundamental data structures (including stacks , o b as well as fundamental data structures (including stacks , queues , and symbol tables) and their application , emphasizing the use of the scientific method to understand performance characteristics of implementations. Applications in science and engineering are a key feature of the text. We motivate each programming concept that we address by examining its impact on specific applications. We draw examples from applied mathematics , the physical and biological sciences , and computer science itself , and include simulation of physical .ical sciences , and computer science itself , and include simulation of physical . , 1 , , ,1 1 1 1. 1.. 1 1.. systems , numerical methods , data visualization , sound synthesis , image process. lug , financial simulation , and information technology. Specific examples include a . , treatment in the first chapter of Markov chains for web page ranks and case studs page ranks and case stud. , 1 1. , 1 ies that address the percolation problem , N-body simulation , and the small-world . IV phenomenon. These applications are an integral part of the text. They engage students in the material , illustrate the importance of the programming concepts , and provide persuasive evidence of the critical role played by computation in modern .1.. science and enZineering bineering. Our primary goal is to teach the specific mechanisms and skills that are needed to develop effective solutions to any programming problem. We work with comp effective solutions to any programming problem. We work with complete Java programs and encourage readers to use them. We focus on programming by individuals , not library programming or programming in the large (which we treat brieny in an appendix).] in an appendix). Use in the Curriculum This book is intended for a first-year college course l b .1 , ... , aimed at teaching novices to Drogram in the context of scientific applications. b novices to program in the context of scientific applications. Taught from this book , prospective maiors in any area of science and engineer.ilt from this book , prospective majors in any area of science and engineer.. 11 1. lug will learn to program in a familiar context. Any student completing a course based on this book will be well--prepared to apply their skills in later courses in science and engineering and to recognize when further education in comDuter science .ineering and to recognize when further education in computer science might be beneficial.

Prospective computer science majors , in particular , can benetit from learning pective computer science majors , in particular , can benefit from learning . to program in the context of scientific applications. A computer scientist needs the same basic background in the scientific method and the same exDosure to the role .found in the scientific method and the same exposure to the role o f computation in science as does a biologist , an engineer , or a physicist. Indeed , our interdisciplinary approach enables colleges and uniVersities to teach prospective computer science majors and prospective majors in other fields o f science and engineering in the same course. We cover the material Drescribed by .ineering in the same course. We cover the material prescribed by CSI , but our focus on applications brings life to the concepts and motivates students to learn them. Our interdisciplinary approach exposes students to problems . 1. or 1.. , . 1 1. 1. 1 1. In many different disciDlines , helping them to more wisely choose a major. J pnnes , helping them to more wisely choose a major. Whatever the specific mechanism , the use of this book is best positioned early .. , in the curriculum. First , this positioning allows us to leverage familiar material in high school mathematics and science. Second , students who learn to program .if school mathematics and science. Second , students who learn to program early in their coheRe curriculum will then be able to use comDuters more effectiVely j D puters more effectiVely when movinZ on to courses in their SDecialtv Like reading and writing Drogramb pecialty. Like reading and writing , program... 1. 1. 1 1 mInR is certain to be an essential skill for any scientist or enZineer. Students who o is certain to be an essential skill for any scientist or engineer. Students who have grasDed the conceDts in this book will continually develoD that skill through a hi asped the concepts in this book will continually develop that skill through a lifetime , reaping the benefits of exploiting computation to solVe or to better understand the problems and projects that arise in their chosen field. v Prerequisites This book is meant to be suitable for typical science and

<<Java程序设计>>

This book is meant to be suitable for typical science and engineering students in their first year of college. That is, we do not expect preparation beyond what is typically required for other entry-level science and mathematics courses. Mathematical maturity is important. While we do not dwell on mathematical details, we do refer to the mathematics curriculum that students have taken in high school, including algebra, geometry, and trigonometry. Most students in our target audience (those intending to major in the sciences and engineering) automatically meet these requirements. Indeed, we take advantage of their familiarity with the basic curriculum to introduce basic programming concepts. Scientific curiosity is also an essential ingredient. Science and engineering students bring with them a sense of fascination in the ability of scientific inquiry to help explain what goes on in nature. We leverage this predilection with examples of simple programs that speak volumes about the natural world. We do not assume any specific knowledge beyond that provided by typical high school courses in mathematics, physics, biology, or chemistry. Programming experience is not necessary, but also is not harmful. Teaching programming is our primary goal, so we assume no prior programming experience. But writing a program to solve a new problem is a challenging intellectual task, so students who have written numerous programs in high school can benefit from taking an introductory programming course based on this book (just as students who have written numerous essays in high school can benefit from an introductory writing course in college). The book can support teaching students with varying backgrounds because the applications appeal to both novices and experts alike. Experience using a computer is also not necessary, but also is not at all a problem. Every college student nowadays uses a computer regularly, to communicate with friends and relatives, listen to music, process photos, and many other activities. The realization that they can harness the power of their own computer in interesting and important ways is an exciting and lasting lesson for most students. In summary, virtually all students in science and engineering are prepared to take a course based on this book as a part of their first-semester curriculum. Goals What can instructors of upper-level courses in science and engineering expect of students who have completed a course based on this book? We cover the CS1 curriculum, but anyone who has taught an introductory programming course knows that expectations of instructors in later courses are typically high: each instructor expects all students to be familiar with the computing environment and approach that he or she wants to use. A physics professor might expect some students to design a program over the weekend to run a simulation; an engineering professor might expect other students to be using a particular package to numerically solve differential equations; or a computer science professor might expect knowledge of the details of a particular programming environment. Is it realistic to meet such diverse expectations? Should there be a different introductory course for each set of students? Colleges and universities have been wrestling with such questions since computers came into widespread use in the latter part of the 20th century. Our answer to them is found in this common introductory treatment of programming, which is analogous to commonly accepted introductory courses in mathematics, physics, biology, and

<<Java程序设计>>

chemistry. An introduction to Programming .., 1..1, , 1.., strives to provide the basic preparation needed by all students in science and en.. 1.1 1. 1 1 1 1. 1 1 1 engineering, while sending the clear message that there is much more to understand about computer science than programming. Instructors teaching students who have studied from this book can expect that they have the knowledge and expert, , , .. 1 1 -1 1. 1. 1 ence necessary to enable them to adaDt to new comDutational environments and to J to enable them to adapt to new computational environments and to effectiVely exDloit comDuters in diVerse aDplications.] ploit computers in diVerse applications. What can students who have completed a course based on this book expect to accomplish in later courses Our messaZe is that DroZramminR is not difficult to learn and that harnesso programming is not difficult to learn and that harness. lug the power of the computer is rewarding. Students who master the material in this book are prepared to address computational challenges wherever they might appear later in their careers. They learn that modern programming environments, such as the one provided by Java, help open the door to any computational problem they might encounter later, and they Rain the condence to learn, evaluate, j but encounter later, and they gain the confidence to learn, evaluate, and use other computational tools. Students interested in computer science will be well-prepared to pursue that interest; students in science and engineering will be . ready to integrate comDutation into their studies. j crate computation into their studies. - yi l Booksite An extensiVe amount of information that supplements this text may be found on the web at http : //www. cs. p ri nceton. edu/Int rop rog rammi ng For economy, we refer to this site as the booksite throughout. It contains material], we refer to this site as the booksite throughout. It contains material for instructors, students, and casual readers of the book. We briefly describe this j ., 1 1 1 1 1.. 1 1. material here, though, as all web users know, it is best surveyed by browsing. With a few exceptions to support testing, the material is all publicly available. One of the most important implications of the booksite is that it empow. , . 1 1, ers instructors and students to use their own computers to teach and learn the . . 1 ^ . 1 1 1 1. 1. material. Anyone with a comDuter and a browser can begin learninZ to DroZram j puter and a browser can begin learning to program by following a few instructions on the booksite. The process is no more difficult than downloading a media Dlaver or a souR As with any website, our booksite is b a media player or a song. As with any website, our booksite is . 1 1. , continually evolVing it is an essential resource for everyone who owns this book. In j b. It is an essential resource for everyone who owns this book. In . 1... . 1 1 1 1. 1.. 1 1 r 1. particular, the supplemental materials are critical to our goal of making computer .. 1 science an integral comDonent of the education of all scientists and engineers. .raf component of the education of all scientists and engineers. For instructors, the booksite contains information about teaching. This information is primarily organized around a teaching style that we have developed 1 1 1 1 1 re 1 1 1 1. over the past decade, where we offer two lectures per week to a large audience, past decade, where we offer two lectures per week to a large audience, 1 1 1 1. 1 1 1. 1 1 1 supplemented by two class sessions per week where students meet in small groups m with instructors or teaching assistants. The booksite has Dresentation slides for the b assistants. The booksite has presentation slides for the lectures, which set the tone. For teaching assistants, the booksite contains detailed problem sets and prod [he booksite contains detailed problem sets and pro.. , . 1 1 1 1 1. r 1 1 1 1. 1 programming projects, which are based on exercises from the book but contain much 1. 1 v 1.... 1 1 1 1 more detail. Each programming assignment is intended to teach a relevant concept programming assignment is intended to teach a relevant concept . , in the context of an interesting application while presenting an inviting and engage pplication while presenting an inviting and engag. 1 1 1 1 1 al. r. 1 1. lug challenge to each student. The progression of assignments embodies our ape 1.. al 1 1. proach to teaching programming. The booksite fully specifies all the assignments 1. 1 1 . 1 1 1. and provides detailed, structured information to help students complete them in the allotted time, including descriptions of suggested approaches and outlines for what should be taught in class sessions. .lit in class sessions. For students, the booksite contains

<<Java程序设计>>

quick access to much of the material in the book , including source code , plus extra material to encourage self--learning. Solutions are provided for many of the books exercises , including complete program code and test data. There is a wealth of information associated with programming assignments , including suggested approaches , checklists , FAQs , and test data. For casual readers (including instructors , teaching assistants , and students!) , the booksite is a resource for accessing all manner of extra information associated with the books content. All of the booksite content provides web links and other routes to pursue more information about the topic under consideration. There is far more information accessible than any individual could fully digest , but our goal is to provide enough to whet any readers appetite for more information about the books content.

Acknowledgements This project has been under development since 1992 , so far too many people have contributed to its success for us to acknowledge them all here. Special thanks are due to Anne Rogers for helping to start the ball rolling; to Dave Hanson , Andrew Appel , and Chris van Wyk , for their patience in explaining logic data abstraction; and to Lisa Worthington , for being the first to truly relish the challenge of teaching this material to first-year students. We also gratefully acknowledge the efforts of /dev/126 (the summer students who have contributed so much of the content); the faculty , graduate students , and teaching staff who have dedicated themselves to teaching this material over the past 15 years here at Princeton; and the thousands of undergraduates who have dedicated themselves to learning it.

<<Java程序设计>>

内容概要

本书由计算机著名专家Robert Sedgewick编写。

本书根据学习编程的四个阶段（基本元素、函数、面向对象程序设计、算法和数据结构）来组织。

基本元素包括变量、赋值语句、数据的内置类型、控制流（条件和循环）、数组和输入输出、包括图和声音。

函数包括库函数和递归函数，将程序划分成可以独立调试、维护和复用的程序段。

然后介绍如何使用、创建和设计数据类型。

最后介绍了经典的排序和搜索算法，同时也介绍了基本的数据结构（包括堆栈、队列和符号表）及它们的应用，强调了使用科学方法来理解实现的性能特性。

本书示例丰富，涉及了Java在各种学科之中的应用，具有很高的参考价值。

本书适用于高等院校计算机及相关专业的研究生和本科生，以及从事编程工作的专业人员。

<<Java程序设计>>

作者简介

Robert Sedgwick, 普林斯顿大学计算机科学系教授, 在美国斯坦福大学获得博士学位, 师从算法与程序设计大师Donald E. Knuth教授。

他所著作的《算法》系列书在计算机界有着举足轻重的影响力。

<<Java程序设计>>

书籍目录

Preface Elements of Programming 1.1 Your First Program 1.2 Built-in Types of Data 1.3 Conditionals and Loops 1.4 Arrays 1.5 Input and Output 1.6 Case Study: Random Web Surfer Functions and Modules 2.1 Static Methods 2.2 Libraries and Clients 2.3 Recursion 2.4 Case Study: Percolation Object-Oriented Programming 3.1 Data Types 3.2 Creating Data Types 3.3 Designing Data Types 3.4 Case study: N-body Simulation Algorithms and Data Structures 4.1 Performance 4.2 Sorting and Searching 4.3 Stacks and Queues 4.4 Symbol Tables 4.5 Case study: small World

<<Java程序设计>>

编辑推荐

按学习编程的四个阶段来组织本书。

示例丰富，涉及了Java在各种学科之中的应用，具有很高的参考价值。

内容齐备的配套网址，编码示例、附加练习等内容可以从

<<Java程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>