

# Distributed Programming With Java

If you ally obsession such a referred **Distributed Programming With Java** books that will allow you worth, acquire the no question best seller from us currently from several preferred authors. If you desire to witty books, lots of novels, tale, jokes, and more fictions collections are afterward launched, from best seller to one of the most current released.

You may not be perplexed to enjoy all book collections Distributed Programming With Java that we will utterly offer. It is not going on for the costs. Its not quite what you need currently. This Distributed Programming With Java , as one of the most energetic sellers here will no question be in the midst of the best options to review.

## **Java RMI** - William Grosso 2002

If you're a distributed Java or Enterprise JavaBeans programmer, then you've undoubtedly heard of Java's Remote Method Invocation (RMI). Java programmers use RMI to write efficient, fault-tolerant distributed applications with very little time or effort. Whether you're networking across a LAN or across the Internet, RMI provides Java programmers with a lightweight solution to a heavyweight problem. Java RMI contains a wealth of experience in designing and implementing applications that use Remote Method Invocation. Novice readers will quickly be brought up to speed on why RMI is such a powerful yet easy-to-use tool for distributed programming, while experts can gain valuable experience for constructing their own enterprise and distributed systems. The book also provides strategies for working with: Serialization, Threading, The RMI registry, Sockets and socket factories, Activation, Dynamic class downloading, HTTP tunnelling, Distributed garbage collection, JNDI, CORBA. In short, a treasure trove of valuable RMI knowledge packed into one book!

## **Distributed Programming with Java** - Qusay H. Mahmoud 2000

For programmers already familiar with Java, this book offers new techniques on how to develop distributed applications. Although it discusses four paradigms--low-level Sockets, Remote Method Invocation, CORBA, and Mobile Agents--this book does not favor any one of these technologies. It also allows the reader to judge the easiest approach for a particular domain of applications.

## *Creating Components* - Charles W. Kann 2017-09-11

Concurrency is a powerful technique for developing efficient and lightning- fast software. For instance, concurrency can be used in common applications such as online order processing to speed processing and ensure transaction reliability. However, mastering concurrency is one of the greatest challenges for both new and veteran programmers. Softwar

## **Distributed Computing in Java 9** - Raja Malleswara Rao Pattamsetti 2017-06-30

Explore the power of distributed computing to write concurrent, scalable applications in Java About This Book Make the best of Java 9 features to write succinct code Handle large amounts of data using HPC Make use of AWS and Google App Engine along with Java to establish a powerful remote computation system Who This Book Is For This book is for basic to intermediate level Java developers who is aware of object-oriented programming and Java basic concepts. What You Will Learn Understand the basic concepts of parallel and distributed computing/programming Achieve performance improvement using parallel processing, multithreading, concurrency, memory sharing, and hpc cluster computing Get an in-depth understanding of Enterprise Messaging concepts with Java Messaging Service and Web Services in the context of Enterprise Integration Patterns Work with Distributed Database technologies Understand how to develop and deploy a distributed application on different cloud platforms including Amazon Web Service and Docker CaaS Concepts Explore big data technologies Effectively test and debug distributed systems Gain thorough knowledge of security standards for distributed applications including two-way Secure Socket Layer In Detail Distributed 9 computing is the concept with which a bigger computation process is accomplished by splitting it into multiple smaller logical activities and performed by diverse systems, resulting in maximized performance in lower infrastructure investment. This

book will teach you how to improve the performance of traditional applications through the usage of parallelism and optimized resource utilization in Java 9. After a brief introduction to the fundamentals of distributed and parallel computing, the book moves on to explain different ways of communicating with remote systems/objects in a distributed architecture. You will learn about asynchronous messaging with enterprise integration and related patterns, and how to handle large amount of data using HPC and implement distributed computing for databases. Moving on, it explains how to deploy distributed applications on different cloud platforms and self-contained application development. You will also learn about big data technologies and understand how they contribute to distributed computing. The book concludes with the detailed coverage of testing, debugging, troubleshooting, and security aspects of distributed applications so the programs you build are robust, efficient, and secure. Style and approach This is a step-by-step practical guide with real-world examples.

## **Network Programming with Go** - Jan Newmarch 2017-05-15

Dive into key topics in network architecture and Go, such as data serialization, application level protocols, character sets and encodings. This book covers network architecture and gives an overview of the Go language as a primer, covering the latest Go release. Beyond the fundamentals, Network Programming with Go covers key networking and security issues such as HTTP and HTTPS, templates, remote procedure call (RPC), web sockets including HTML5 web sockets, and more. Additionally, author Jan Newmarch guides you in building and connecting to a complete web server based on Go. This book can serve as both as an essential learning guide and reference on Go networking. What You Will Learn Master network programming with Go Carry out data serialization Use application-level protocols Manage character sets and encodings Deal with HTTP(S) Build a complete Go-based web server Work with RPC, web sockets, and more Who This Book Is For Experienced Go programmers and other programmers with some experience with the Go language.

## ECOOP 2008 - Object-Oriented Programming - Jan Vitek 2008-07-10

It is a pleasure to present the proceedings of the 22nd European Conference on Object-Oriented Programming (ECOOP 2008) held in Paphos, Cyprus. The conference continues to serve a broad object-oriented community with a tech- cal program spanning theory and practice and a healthy mix of industrial and academic participants. This year a strong workshop and tutorial program c- plementedthemaintechnicaltrack.We had 13 workshops and 8 tutorials, as well as the co-located Dynamic Language Symposium (DLS). Finally, the program was rounded out with a keynote by Rachid Guerraoui and a banquet speech by James Noble. As in previous years, two Dahl-Nygaard awards were selected by AITO, and for the ?rst time, the ECOOP Program Committee gave a best paper award. The proceedings include 27 papers selected from 138 submissions. The papers werereviewed in a single-blind process with three to ?ve reviews per paper. P- liminary versions of the reviews were made available to the authors a week before the PC meeting to allow for short (500 words or less) author responses. The - sponses were discussed at the PC meeting and were instrumental in reaching decisions. The PC discussions followed Oscar Nierstrasz' Champion pattern. PC papers had ?ve reviews and were held at a higher standard.

Java in Distributed Systems - Marko Boger 2001-05-25

Large and complex software systems, such as Internet applications, depend on distributed applications. Although Java has helped reduce the complexity of distributed systems, developers still have to contend with diverse hardware platforms, remote communication over networks, and system failures. Java in Distributed Systems provides a comprehensive guide for anyone wishing to deepen their knowledge of Java in distributed applications. Beginning with a tutorial guide to distributed programming in the Java environment, it shows you how building blocks from threads to Jini can help you to fulfil Sun's vision, that 'the Network is the Computer'. It then goes on to focus on aspects that are still challenging researchers such as concurrency, distribution, and persistence. Key Features: - One of the few books to focus specifically on Java for building distributed applications - Coverage includes threads & sockets, RMI, CORBA, Voyager, Mobile agents, JDBC, object-oriented databases, Java spaces and Jini - Includes advanced chapters on the cutting edge of Java language development, including the author's own proposed Dejay (Distributed Java), an open-source project that offers a unified approach to concurrency, distribution and persistence

*Concurrent Programming* - Stephen J. Hartley 1998

An algorithm animation package, written in Java, is used in several of the example programs. The book concludes with a brief introduction to parallel processing with Java."--BOOK JACKET.

"Concurrent Programming: The Java Programming Language is ideal for a concurrent programming course or as a supplement in an operating systems class. Professional programmers will also find the presentation accessible."--BOOK JACKET.

Distributed Programming with Java - Gabriel Oteniya 2005

**Data Structures and Program Design Using Java** - D. Malhotra 2020-02-28

Data structures provide a means to managing large amounts of information such as large databases, using SEO effectively, and creating Internet/Web indexing services. This book is designed to present fundamentals of data structures for beginners using the Java programming language in a friendly, self-teaching format. Practical analogies using real world applications are integrated throughout the text to explain technical concepts. The book includes a variety of end-of-chapter practice exercises, e.g., programming, theoretical, and multiple-choice. Features: Covers data structure fundamentals using Java Numerous tips, analogies, and practical applications enhance understanding of subjects under discussion "Frequently Asked Questions" integrated throughout the text clarify and explain concepts Includes a variety of end-of-chapter exercises, e.g., programming, theoretical, and multiple choice

*Java Message Service* - David A Chappell 2000-12-04

This book is a thorough introduction to Java Message Service (JMS), the standard Java application program interface (API) from Sun Microsystems that supports the formal communication known as "messaging" between computers in a network. JMS provides a common interface to standard messaging protocols and to special messaging services in support of Java programs. The messages exchange crucial data between computers, rather than between users--information such as event notification and service requests. Messaging is often used to coordinate programs in dissimilar systems or written in different programming languages. Using the JMS interface, a programmer can invoke the messaging services of IBM's MQSeries, Progress Software's SonicMQ, and other popular messaging product vendors. In addition, JMS supports messages that contain serialized Java objects and messages that contain Extensible Markup Language (XML) pages. Messaging is a powerful new paradigm that makes it easier to uncouple different parts of an enterprise application. Messaging clients work by sending messages to a message server, which is responsible for delivering the messages to their destination. Message delivery is asynchronous, meaning that the client can continue working without waiting for the message to be delivered. The contents of the message can be anything from a simple text string to a serialized Java object or an XML document. Java Message Service shows how to build applications

using the point-to-point and publish-and-subscribe models; how to use features like transactions and durable subscriptions to make an application reliable; and how to use messaging within Enterprise JavaBeans. It also introduces a new EJB type, the MessageDrivenBean, that is part of EJB 2.0, and discusses integration of messaging into J2EE.

**Distributed Java 2 Platform** - Stewart Birnam 2001

Build, deploy, and manage robust Web-enabled database applications with Java! Integrate J2EE, Oracle, Linux, and legacy databases. Covers JDBC, RMI, Swing, Apache JServ, and much more. Practical techniques for enhancing reusability and day-to-day manageability. Need to build, enhance, integrate, or manage an enterprise-class Java-based distributed database system? Tired of "toy examples"? Want to see how everything fits together in a live, breathing system? Here's your chance. Stewart Birnam shows experienced professionals all they need to create and administer robust database applications leveraging Java technologies, and running on today's key platforms-Oracle 8.x and Linux. You won't just learn a laundry list of APIs and technologies: you'll master real-world strategies for integrating them in reliable, deployable systems-and discover proven solutions for the problems that stand in your way. Understand every issue associated with building scalable, efficient, multi-tier Java code, including server configuration, brokers, client interaction, and maximizing system price/performance. Make the most of Java's database APIs; and master powerful database access techniques, from queuing to complex joins, multimedia content delivery to Oracle-specific features. Learn how to build and administer RMI servers; streamline deployment and distribution of enterprise applications; maximize code reusability; and manage your distributed application as efficiently as possible. Stewart Birnam is a Senior Database/Systems Developer at a leading visual effects and computer animation company in northern California.

Fundamentals of Distributed Object Systems - Zahir Tari 2004-04-07

Distributed Object Computing teaches readers the fundamentals of CORBA, the leading architecture for design of software used in parallel and distributed computing applications. Since CORBA is based on open standards, it is the only effective way to learn object-oriented programming for distributed systems. This language independent book allows material to be taught using Java, C++ or other Object Oriented Programming Languages.

**Distributed, Embedded and Real-time Java Systems** - M. Teresa Higuera-Toledano 2012-02-07

Research on real-time Java technology has been prolific over the past decade, leading to a large number of corresponding hardware and software solutions, and frameworks for distributed and embedded real-time Java systems. This book is aimed primarily at researchers in real-time embedded systems, particularly those who wish to understand the current state of the art in using Java in this domain. Much of the work in real-time distributed, embedded and real-time Java has focused on the Real-time Specification for Java (RTSJ) as the underlying base technology, and consequently many of the Chapters in this book address issues with, or solve problems using, this framework. Describes innovative techniques in: scheduling, memory management, quality of service and communication systems supporting real-time Java applications; Includes coverage of multiprocessor embedded systems and parallel programming; Discusses state-of-the-art resource management for embedded systems, including Java's real-time garbage collection and parallel collectors; Considers hardware support for the execution of Java programs including how programs can interact with functional accelerators; Includes coverage of Safety Critical Java for development of safety critical embedded systems.

**Designing Distributed Systems** - Brendan Burns 2018-02-20

Without established design patterns to guide them, developers have had to build distributed systems from scratch, and most of these systems are very unique indeed. Today, the increasing use of containers has paved the way for core distributed system patterns and reusable containerized components. This practical guide presents a collection of repeatable, generic patterns to help make the development of reliable distributed systems far more approachable and

efficient. Author Brendan Burns—Director of Engineering at Microsoft Azure—demonstrates how you can adapt existing software design patterns for designing and building reliable distributed applications. Systems engineers and application developers will learn how these long-established patterns provide a common language and framework for dramatically increasing the quality of your system. Understand how patterns and reusable components enable the rapid development of reliable distributed systems Use the side-car, adapter, and ambassador patterns to split your application into a group of containers on a single machine Explore loosely coupled multi-node distributed patterns for replication, scaling, and communication between the components Learn distributed system patterns for large-scale batch data processing covering work-queues, event-based processing, and coordinated workflows

[Principles of Concurrent and Distributed Programming](#) - M. Ben-Ari 2006

Principles of Concurrent and Distributed Programming provides an introduction to concurrent programming focusing on general principles and not on specific systems. Software today is inherently concurrent or distributed - from event-based GUI designs to operating and real-time systems to Internet applications. This edition is an introduction to concurrency and examines the growing importance of concurrency constructs embedded in programming languages and of formal methods such as model checking.

**Distributed Programming with Ruby** - Mark Bates 2009-11-05

Complete, Hands-On Guide to Building Advanced Distributed Applications with Ruby Distributed programming techniques make applications easier to scale, develop, and deploy—especially in emerging cloud computing environments. Now, one of the Ruby community's leading experts has written the first definitive guide to distributed programming with Ruby. Mark Bates begins with a simple distributed application, and then walks through an increasingly complex series of examples, demonstrating solutions to the most common distributed programming problems. Bates presents the industry's most useful coverage of Ruby's standard distributed programming libraries, DRb and Rinda. Next, he introduces powerful third-party tools, frameworks, and libraries designed to simplify Ruby distributed programming, including his own Distribunaut. If you're an experienced Ruby programmer or architect, this hands-on tutorial and practical reference will help you meet any distributed programming challenge, no matter how complex. Coverage includes Writing robust, secure, and interactive applications using DRb—and managing its drawbacks Using Rinda to build applications with improved flexibility, fault tolerance, and service discovery Simplifying DRb service management with RingyDingy Utilizing Starfish to facilitate communication between distributed programs and to write MapReduce functions for processing large data sets Using Politics to customize the processes running on individual server instances in a cloud computing environment Providing reliable distributed queuing with the low-overhead Starling messaging server Implementing comprehensive enterprise messaging with RabbitMQ and Advanced Message Queuing Protocol (AMQP) Offloading heavyweight tasks with BackgroundDRb and DelayedJob

**Introduction to Reliable and Secure Distributed Programming** - Christian Cachin 2011-02-11

In modern computing a program is usually distributed among several processes. The fundamental challenge when developing reliable and secure distributed programs is to support the cooperation of processes required to execute a common task, even when some of these processes fail. Failures may range from crashes to adversarial attacks by malicious processes. Cachin, Guerraoui, and Rodrigues present an introductory description of fundamental distributed programming abstractions together with algorithms to implement them in distributed systems, where processes are subject to crashes and malicious attacks. The authors follow an incremental approach by first introducing basic abstractions in simple distributed environments, before moving to more sophisticated abstractions and more challenging environments. Each core chapter is devoted to one topic, covering reliable broadcast, shared memory, consensus, and extensions of consensus. For every topic, many exercises and their solutions enhance the understanding This book

represents the second edition of "Introduction to Reliable Distributed Programming". Its scope has been extended to include security against malicious actions by non-cooperating processes. This important domain has become widely known under the name "Byzantine fault-tolerance".

[Programming Distributed Computing Systems](#) - Carlos A. Varela 2013-05-31

An introduction to fundamental theories of concurrent computation and associated programming languages for developing distributed and mobile computing systems. Starting from the premise that understanding the foundations of concurrent programming is key to developing distributed computing systems, this book first presents the fundamental theories of concurrent computing and then introduces the programming languages that help develop distributed computing systems at a high level of abstraction. The major theories of concurrent computation—including the  $\pi$ -calculus, the actor model, the join calculus, and mobile ambients—are explained with a focus on how they help design and reason about distributed and mobile computing systems. The book then presents programming languages that follow the theoretical models already described, including Pict, SALSA, and JoCaml. The parallel structure of the chapters in both part one (theory) and part two (practice) enable the reader not only to compare the different theories but also to see clearly how a programming language supports a theoretical model. The book is unique in bridging the gap between the theory and the practice of programming distributed computing systems. It can be used as a textbook for graduate and advanced undergraduate students in computer science or as a reference for researchers in the area of programming technology for distributed computing. By presenting theory first, the book allows readers to focus on the essential components of concurrency, distribution, and mobility without getting bogged down in syntactic details of specific programming languages. Once the theory is understood, the practical part of implementing a system in an actual programming language becomes much easier.

**Distributed Programming** - A. Udaya Shankar 2012-09-15

Distributed Programming: Theory and Practice presents a practical and rigorous method to develop distributed programs that correctly implement their specifications. The method also covers how to write specifications and how to use them. Numerous examples such as bounded buffers, distributed locks, message-passing services, and distributed termination detection illustrate the method. Larger examples include data transfer protocols, distributed shared memory, and TCP network sockets. Distributed Programming: Theory and Practice bridges the gap between books that focus on specific concurrent programming languages and books that focus on distributed algorithms. Programs are written in a "real-life" programming notation, along the lines of Java and Python with explicit instantiation of threads and programs. Students and programmers will see these as programs and not "merely" algorithms in pseudo-code. The programs implement interesting algorithms and solve problems that are large enough to serve as projects in programming classes and software engineering classes. Exercises and examples are included at the end of each chapter with on-line access to the solutions. Distributed Programming: Theory and Practice is designed as an advanced-level text book for students in computer science and electrical engineering. Programmers, software engineers and researchers working in this field will also find this book useful.

**Nonsequential and Distributed Programming with Go** - Christian Maurer 2021-01-19

Der Band bietet eine kompakte Einführung in die Nichtsequentielle Programmierung als gemeinsamen Kern von Vorlesungen über Betriebssysteme, Verteilte Systeme, Parallele Algorithmen, Echtzeitprogrammierung und Datenbanktransaktionen. Basiskonzepte zur Synchronisation und Kommunikation nebenläufiger Prozesse werden systematisch dargestellt: Schlösser, Semaphore, Monitore, lokaler und netzweiter Botschaftenaustausch. Die Algorithmen sind in der Programmiersprache Google Go formuliert, mit der viele Synchronisationskonzepte ausgedrückt werden können.

[Java Programming with CORBA](#) - Gerald Brose 2001-07-05

The leading guide for Java developers who build business applications with CORBA Acknowledged experts present advanced techniques and real-world examples for building both simple and

complex programs using Java with CORBA. The authors begin with a quick overview of CORBA, Java, object request brokers (ORBs), and EJB components, then quickly move on to show how to use them to build complete Java applications. This new volume features in-depth code examples, as well as expanded coverage of cutting-edge topics, including Portable Object Adaptor (POA), Remote Method Invocation (RMI) over IIOP, and EJB.

*Java Distributed Computing* - Jim Farley 1998-01-01

Distributed computing and Java go together naturally. As the first language designed from the bottom up with networking in mind, Java makes it very easy for computers to cooperate. Even the simplest applet running in a browser is a distributed application, if you think about it. The client running the browser downloads and executes code that is delivered by some other system. But even this simple applet wouldn't be possible without Java's guarantees of portability and security: the applet can run on any platform, and can't sabotage its host. Of course, when we think of distributed computing, we usually think of applications more complex than a client and server communicating with the same protocol. We usually think in terms of programs that make remote procedure calls, access remote databases, and collaborate with others to produce a single result. Java Distributed Computing discusses how to design and write such applications. It covers Java's RMI (Remote Method Invocation) facility and CORBA, but it doesn't stop there; it tells you how to design your own protocols to build message passing systems and discusses how to use Java's security facilities, how to write multithreaded servers, and more. It pays special attention to distributed data systems, collaboration, and applications that have high bandwidth requirements. In the future, distributed computing can only become more important. Java Distributed Computing provides a broad introduction to the problems you'll face and the solutions you'll find as you write distributed computing applications. Topics covered in Java Distributed Computing: Introduction to Distributed Computing Networking Basics Distributed Objects (Overview of CORBA and RMI) Threads Security Message Passing Systems Distributed Data Systems (Databases) Bandwidth Limited Applications Collaborative Systems

**Parallel and Distributed Programming Using C++** - Cameron Hughes 2004

This text takes complicated and almost unapproachable parallel programming techniques and presents them in a simple, understandable manner. It covers the fundamentals of programming for distributed environments like Internets and Intranets as well as the topic of Web Based Agents. Distributed programming with Java and CORBA - Mehmet Bozkurt 2001

*Distributed Programming* - A. Udaya Shankar 2012-09-15

Distributed Programming: Theory and Practice presents a practical and rigorous method to develop distributed programs that correctly implement their specifications. The method also covers how to write specifications and how to use them. Numerous examples such as bounded buffers, distributed locks, message-passing services, and distributed termination detection illustrate the method. Larger examples include data transfer protocols, distributed shared memory, and TCP network sockets. Distributed Programming: Theory and Practice bridges the gap between books that focus on specific concurrent programming languages and books that focus on distributed algorithms. Programs are written in a "real-life" programming notation, along the lines of Java and Python with explicit instantiation of threads and programs. Students and programmers will see these as programs and not "merely" algorithms in pseudo-code. The programs implement interesting algorithms and solve problems that are large enough to serve as projects in programming classes and software engineering classes. Exercises and examples are included at the end of each chapter with on-line access to the solutions. Distributed Programming: Theory and Practice is designed as an advanced-level text book for students in computer science and electrical engineering. Programmers, software engineers and researchers working in this field will also find this book useful.

*Programming Essentials Using Java* - William McAllister 2017-02-02

This is a one-semester, introductory programming textbook in Java that uses game applications as

a central pedagogical tool to improve student engagement, learning outcomes, and retention. Game programming is incorporated into the text in a way that does not compromise the amount of material traditionally covered in a basic programming course and permits instructors who are not familiar with game programming and computer graphics concepts to realize the verified pedagogical advantages of game applications. The companion disc includes a game environment that is easily integrated into projects created with the popular Java Development Environments, including Eclipse, NetBeans, and JCreator in a student-friendly way and also includes a set of executable student games to pique their interest by giving them a glimpse into their future capabilities. The material presented in the book is in full compliance with the 2013 ACM/IEEE computer science curriculum guidelines. It has been used to teach programming to students whose majors are within and outside of the computing fields. Ancillaries include a comprehensive instructor's resource disc with programming solutions, slides, quizzes, projects, and more. FEATURES: \* Uses an objects-early approach to learning Java \* Follows the 2013 ACM/IEEE computer science curriculum guidelines \* Integrates game applications as a central pedagogical tool to improve student engagement, learning outcomes, and retention \* Includes a companion disc with projects created with the popular Java Development Environments; also includes a set of executable student games, source code, and figures \* Uses working programs to illustrate concepts under discussion \* Complete instructor's resource package available upon adoption

*Head First Java* - Kathy Sierra 2005-02-09

Learning a complex new language is no easy task especially when it's an object-oriented computer programming language like Java. You might think the problem is your brain. It seems to have a mind of its own, a mind that doesn't always want to take in the dry, technical stuff you're forced to study. The fact is your brain craves novelty. It's constantly searching, scanning, waiting for something unusual to happen. After all, that's the way it was built to help you stay alive. It takes all the routine, ordinary, dull stuff and filters it to the background so it won't interfere with your brain's real work--recording things that matter. How does your brain know what matters? It's like the creators of the Head First approach say, suppose you're out for a hike and a tiger jumps in front of you, what happens in your brain? Neurons fire. Emotions crank up. Chemicals surge. That's how your brain knows. And that's how your brain will learn Java. Head First Java combines puzzles, strong visuals, mysteries, and soul-searching interviews with famous Java objects to engage you in many different ways. It's fast, it's fun, and it's effective. And, despite its playful appearance, Head First Java is serious stuff: a complete introduction to object-oriented programming and Java. You'll learn everything from the fundamentals to advanced topics, including threads, network sockets, and distributed programming with RMI. And the new, second edition focuses on Java 5.0, the latest version of the Java language and development platform. Because Java 5.0 is a major update to the platform, with deep, code-level changes, even more careful study and implementation is required. So learning the Head First way is more important than ever. If you've read a Head First book, you know what to expect--a visually rich format designed for the way your brain works. If you haven't, you're in for a treat. You'll see why people say it's unlike any other Java book you've ever read. By exploiting how your brain works, Head First Java compresses the time it takes to learn and retain--complex information. Its unique approach not only shows you what you need to know about Java syntax, it teaches you to think like a Java programmer. If you want to be bored, buy some other book. But if you want to understand Java, this book's for you.

*Java Distributed Objects* - Bill McCarty 1999

This book is a comprehensive guide to Java distributed computing. The book covers networking, distributed computing architectures, advanced Java facilities, security, data managing, and specific distributed computing techniques including sockets, Remote Method Invocation, Java servlets, Microsoft's Distributed Component Model, and the Common Object Request Broker Architecture.

Concurrent and Distributed Computing in Java - Vijay K. Garg 2005-01-28

Concurrent and Distributed Computing in Java addresses fundamental concepts in concurrent computing with Java examples. The book consists of two parts. The first part deals with techniques for programming in shared-memory based systems. The book covers concepts in Java such as threads, synchronized methods, waits, and notify to expose students to basic concepts for multi-threaded programming. It also includes algorithms for mutual exclusion, consensus, atomic objects, and wait-free data structures. The second part of the book deals with programming in a message-passing system. This part covers resource allocation problems, logical clocks, global property detection, leader election, message ordering, agreement algorithms, checkpointing, and message logging. Primarily a textbook for upper-level undergraduates and graduate students, this thorough treatment will also be of interest to professional programmers.

**Java Network Programming and Distributed Computing** - David Reilly 2002

Java's rich, comprehensive networking interfaces make it an ideal platform for building today's networked, Internet-centered applications, components, and Web services. Now, two Java networking experts demystify Java's complex networking API, giving developers practical insight into the key techniques of network development, and providing extensive code examples that show exactly how it's done. David and Michael Reilly begin by reviewing fundamental Internet architecture and TCP/IP protocol concepts all network programmers need to understand, as well as general Java features and techniques that are especially important in network programming, such as exception handling and input/output. Using practical examples, they show how to write clients and servers using UDP and TCP; how to build multithreaded network applications; and how to utilize HTTP and access the Web using Java. The book includes detailed coverage of server-side application development; distributed computing development with RMI and CORBA; and email-enabling applications with the powerful JavaMail API. For all beginning to intermediate Java programmers, network programmers who need to learn to work with Java.

**Foundations of Multithreaded, Parallel, and Distributed Programming** - Gregory R. Andrews 2000

Foundations of Multithreaded, Parallel, and Distributed Programming covers, and then applies, the core concepts and techniques needed for an introductory course in this subject. Its emphasis is on the practice and application of parallel systems, using real-world examples throughout. Greg Andrews teaches the fundamental concepts of multithreaded, parallel and distributed computing and relates them to the implementation and performance processes. He presents the appropriate breadth of topics and supports these discussions with an emphasis on performance. Features Emphasizes how to solve problems, with correctness the primary concern and performance an important, but secondary, concern Includes a number of case studies which cover such topics as pthreads, MPI, and OpenMP libraries, as well as programming languages like Java, Ada, high performance Fortran, Linda, Occam, and SR Provides examples using Java syntax and discusses how Java deals with monitors, sockets, and remote method invocation Covers current programming techniques such as semaphores, locks, barriers, monitors, message passing, and remote invocation Concrete examples are executed with complete programs, both shared and distributed Sample applications include scientific computing and distributed systems  
0201357526B04062001

**Introduction to Reliable Distributed Programming** - Rachid Guerraoui 2006-05-01

In modern computing a program is usually distributed among several processes. The fundamental challenge when developing reliable distributed programs is to support the cooperation of processes required to execute a common task, even when some of these processes fail. Guerraoui and Rodrigues present an introductory description of fundamental reliable distributed programming abstractions as well as algorithms to implement these abstractions. The authors follow an incremental approach by first introducing basic abstractions in simple distributed environments, before moving to more sophisticated abstractions and more challenging environments. Each core chapter is devoted to one specific class of abstractions, covering reliable delivery, shared memory, consensus and various forms of agreement. This textbook comes with a

companion set of running examples implemented in Java. These can be used by students to get a better understanding of how reliable distributed programming abstractions can be implemented and used in practice. Combined, the chapters deliver a full course on reliable distributed programming. The book can also be used as a complete reference on the basic elements required to build reliable distributed applications.

**Concurrent, Real-Time and Distributed Programming in Java** - Badr Benmammar 2017-12-27

This book provides an introduction to concurrent, real-time, distributed programming with Java object-oriented language support as an algorithm description tool. It describes in particular the mechanisms of synchronization (cooperative and competitive) and sharing of data (internal class, static variables) between threads in Java. He then discusses the use of Java for real-time applications. Consequently, a presentation of the RTSJ (Real Time Specification for Java) specification dedicated to the development of real-time applications in Java is also introduced in this book. Finally, a presentation of programming distributed in Java is presented in this book. We are particularly interested in communication using the TCP Sockets and high-level communication using Java Remote Method Invocation (RMI). The book also contains an annex which contains a practical set of application exercises in relation to the theme of the book. Knowledge of the Java language is a prerequisite for understanding the book.

**Concurrent Programming in Java** - Douglas Lea 2000

Software -- Programming Languages.

**Programming with Enterprise JavaBeans, JTS, and OTS** - Andreas Vogel 1999-04-27

A complete hands-on guide to programming with the powerful new distributed object standards. If you want to take advantage of the bold new generation of object monitor technologies, you need to understand the underlying distributed object and transactional models. This book gets you quickly up and running with the essential knowledge and skills you need to program transactions. In particular, the book covers the Enterprise JavaBeans (EJB), Java Transaction Service (JTS), and Object Transaction Service (OTS) specifications. The book explains all three specifications and includes design patterns as well as substantial real-world programming examples. Although all sample code is implemented with Inprise Applications Server, it is strictly written to the applicable specifications and is fully portable across specification-compliant EJB containers and vendors. On the companion Web site you'll find. \* Sample code. \* OTS and EJB updates.

**Jini in a Nutshell** - Scott Oaks 2000

What is Jini, and how do you use it ? What can it do ? Is it hype, or an important new paradigm for building a new generation of computing technology ? Many developers have been asking these questions over the past year. And the only way to separate the hype from the reality, to figure out what Jini is really all about, is to roll up your sleeves and start working with Jini, to see what it can do. " Jini in a Nutshell " is a quick reference in the tradition of "Java in a Nutshell". It aims to get you started with Jini as quickly and effectively as possible. It includes : thorough coverage of all important concepts, including RMI, basic Jini programming, leasing, transactions, and remote events. Discussions of standard services, including JavaSpaces, many helper services, and service administration. Many examples demonstrating how to write effective Jini code. A quick reference to all the Jini classes, including standard services and support classes. Coverage of the initial Jini release (Jini 1\*0), plus the first major update (1\*1 alpha). Jini is an extension of the Java platform that enables you to write high-reliability software requiring minimal configuration by the user. Things should "just work !" Things should use the network automatically, and be able to look up the services they need to do their work without prior configuration. And they should work reliably, even if parts of the system or the network fail. Although Jini is often discussed as a technology for embedded systems, these goals are as relevant to enterprise software as to building a networked refrigerator. The community of developers interested in Jini goes way beyond embedded systems developers ; it includes people working on business systems, people working on massively distributed computing projects, and people who are just trying to change the way we think of (and

use) computing tools, whether they are software or hardware. No matter what your interests, it's safe to bet that you're interested in building things that "just work." If you are, you need to investigate Jini's claims. And the best way to do that is to ignore the hype and start writing code. This book shows you how.

**Design Patterns for Distributed Programming** - Robert Ott 1997

Reactive Systems in Java - Clement Escoffier 2021-11-10

Reactive systems and event-driven architecture are becoming indispensable to application design, and companies are taking note. Reactive systems ensure that applications are responsive, resilient, and elastic no matter what failures or errors may be occurring, while event-driven architecture offers a flexible and composable option for distributed systems. This practical book helps Java developers bring these approaches together using Quarkus 2.x, the Kubernetes-native Java framework. Clement Escoffier and Ken Finnigan show you how to take advantage of event-driven and reactive principles to build robust distributed systems, reducing latency and increasing

throughput, particularly in microservices and serverless applications. You'll also get a foundation in Quarkus to help you create true Kubernetes-native applications for the cloud. Understand the fundamentals of reactive systems and event-driven architecture Learn how to use Quarkus to build reactive applications Combine Quarkus with Apache Kafka or AMQP to build reactive systems Develop microservices that utilize messages with Quarkus for use in event-driven architectures Learn how to integrate external messaging systems, such as Apache Kafka, with Quarkus Build applications with Quarkus using reactive systems and reactive programming concepts

**JavaTech, an Introduction to Scientific and Technical Computing with Java** - Clark S. Lindsey 2005-10-13

"JavaTech demonstrates the ease with which Java can be used to create powerful network applications and distributed computing applications. It can be used as a textbook for introductory or intermediate level programming courses, and for more advanced students and researchers who need to learn Java for a particular task. JavaTech is up to date with Java 5.0."--BOOK JACKET.