Concurrent Programming The Java Programming Language

Master the essentials of concurrent programming,including testingand debugging This textbook examines languages and libraries for multithreadedprogramming. Readers learn how to create threads in Java and C++,and develop essential concurrent programming and problem-solvingskills. Moreover, the textbook sets itself apart from othercomparable works by helping readers to become proficient in keytesting and debugging techniques. Among the topics covered, readersare introduced to the relevant aspects of Java, the POSIX Pthreadslibrary, and the Windows Win32 Applications ProgrammingInterface. The authors have developed and fine-tuned this book through theconcurrent programming courses they have taught for the past twentyyears. The material, which emphasizes practical tools andtechniques to solve concurrent programming problems, includesoriginal results from the authors' research. Chaptersinclude: * Introduction to concurrent programming * The critical section problem * Semaphores and locks * Monitors * Message-passing * Message-passing in distributed programs * Testing and debugging concurrent programs As an aid to both students and instructors, class libraries havebeen implemented to provide working examples of all the materialthat is covered. These libraries and the testing techniques theysupport can be used to assess student-written programs. Each chapter includes exercises that build skills in programwriting and help ensure that readers have mastered the chapter'skey concepts. The source code for all the listings in the text andfor the synchronization libraries is also provided, as well asstartup files and test cases for the exercises. This textbook is designed for upper-level undergraduates andgraduate students in computer science. With its abundance ofpractical material and inclusion of working code, coupled with anemphasis on testing and debugging, it is also a highly usefulreference for practicing programmers.
A multi-user game, web site, cloud application, or networked database can have thousands of users all interacting at the same time. You need a powerful, industrial-strength tool to handle the really hard problems inherent in parallel, concurrent environments. You need Erlang. In this second edition of the bestselling Programming Erlang, you'll learn how to write parallel programs that scale effortlessly on multicore systems. Using Erlang, you'll be surprised at how easy it becomes to deal with parallel problems, and how much faster and more efficiently your programs run. That's because Erlang uses sets of parallel processes-not a single sequential process, as found in most programming languages. Joe Armstrong, creator of Erlang, introduces this powerful language in small steps, giving you a complete overview of Erlang and how to use it in common scenarios. You'll start with sequential programming, move to parallel programming and handling errors in parallel programs, and learn to work confidently with distributed programming and the standard Erlang/Open Telecom Platform (OTP) frameworks. You need no previous knowledge of functional or parallel

programming. The chapters are packed with hands-on, real-world tutorial examples and insider tips and advice, and finish with exercises for both beginning and advanced users. The second edition has been extensively rewritten. New to this edition are seven chapters covering the latest Erlang features: maps, the type system and the Dialyzer, WebSockets, programming idioms, and a new stand-alone execution environment. You'll write programs that dynamically detect and correct errors, and that can be upgraded without stopping the system. There's also coverage of rebar (the de facto Erlang build system), and information on how to share and use Erlang projects on github, illustrated with examples from cowboy and bitcask. Erlang will change your view of the world, and of how you program. What You Need The Erlang/OTP system. Download it from erlang.org.

This Multi Pack consists of: *Concurrent Systems (ISBN 0201177676) *Concurrent Programming in Java: Design Principles and Pattern (ISBN 0201310090)

Take a deeper dive into the key mechanisms for writing concurrent and parallel programs. Discover how to parallelize a sequential program.

"When you begin using multi-threading throughout an application, the importance of clean architecture and design is critical. . . . This places an emphasis on understanding not only the platform's capabilities but also emerging best practices. Joe does a great job interspersing best practices alongside theory throughout his book." – From the Foreword by Craig Mundie, Chief Research and Strategy Officer, Microsoft Corporation Author Joe Duffy has risen to the challenge of explaining how to write software that takes full advantage of concurrency and hardware parallelism. In Concurrent Programming on Windows, he explains how to design, implement, and maintain large-scale concurrent programs, primarily using C# and C++ for Windows. Duffy aims to give application, system, and library developers the tools and techniques needed to write efficient, safe code for multicore processors. This is important not only for the kinds of problems where concurrency is inherent and easily exploitable—such as server applications, compute-intensive image manipulation, financial analysis, simulations, and AI algorithms—but also for problems that can be speeded up using parallelism but require more effort—such as math libraries, sort routines, report generation, XML manipulation, and stream processing algorithms. Concurrent Programming on Windows has four major sections: The first introduces concurrency at a high level, followed by a section that focuses on the fundamental platform features, inner workings, and API details. Next, there is a section that describes common patterns, best practices, algorithms, and data structures that emerge while writing concurrent software. The final section covers many of the common system-wide architectural and process concerns of concurrent programming. This is the only book you'll need in order to learn the best practices and common patterns for programming with concurrency on Windows and .NET.

This book is an in-depth introduction to Erlang, a programming language ideal for any situation where concurrency, fault tolerance, and fast response is essential. Erlang is gaining widespread adoption with the advent of multi-core processors and their new scalable approach to concurrency. With this guide you'll learn how to write complex concurrent programs in Erlang, regardless of your programming background or experience. Written by leaders of the international Erlang community -- and based on their training material -- Erlang Programming focuses on the language's syntax and semantics, and explains pattern matching, proper lists, recursion, debugging, networking, and concurrency. This book helps you: Understand the strengths of Erlang and why its designers included specific features Learn the concepts behind concurrency and Erlang's way of handling it Write efficient Erlang programs while keeping code neat and readable Discover how Erlang fills the requirements for distributed systems Add simple graphical user interfaces with little effort Learn Erlang's tracing mechanisms for debugging concurrent and distributed systems Use the built-in Mnesia database and other table storage features Erlang Programming provides exercises at the end of each chapter and simple examples throughout the book.
If you're looking to take full advantage of multi-core processors with concurrent programming, this practical book provides the knowledge and hands-on experience you need. The Art of Concurrency is one of the few resources to focus on implementing algorithms in the shared-memory model of multi-core processors, rather than just theoretical models or distributed-memory architectures. The book provides detailed explanations and usable samples to help you transform algorithms from serial to parallel code, along with advice and analysis for avoiding mistakes that programmers typically make when first attempting these computations. Written by an Intel engineer with over two decades of parallel and concurrent programming experience, this book will help you: Understand parallelism and concurrency Explore differences between programming for shared-memory and distributed-memory Learn guidelines for designing multithreaded applications, including testing and tuning Discover how to make best use of different threading libraries, including Windows threads, POSIX threads, OpenMP, and Intel Threading Building Blocks Explore how to implement concurrent algorithms that involve sorting, searching, graphs, and other practical computations The Art of Concurrency shows you how to keep algorithms scalable to take advantage of new processors with even more cores. For developing parallel code algorithms for concurrent programming, this book is a must.
The Art of Multiprocessor Programming, Second Edition, provides users with an authoritative guide to multicore programming. This updated edition introduces higher level software development skills relative to those needed for efficient single-core programming, and includes comprehensive coverage of the new principles, algorithms, and tools necessary for effective multiprocessor programming. The book is an ideal resource for students and professionals alike

who will benefit from its thorough coverage of key multiprocessor programming issues. Features new exercises developed for instructors using the text, with more algorithms, new examples, and other updates throughout the book Presents the fundamentals of programming multiple threads for accessing shared memory Explores mainstream concurrent data structures and the key elements of their design, as well as synchronization techniques, from simple locks to transactional memory systems

Multicore microprocessors are now at the heart of nearly all desktop and laptop computers. While these chips offer exciting opportunities for the creation of newer and faster applications, they also challenge students and educators. How can the new generation of computer scientists growing up with multicore chips learn to program applications that exploit this latent processing power? This unique book is an attempt to introduce concurrent programming to first-year computer science students, much earlier than most competing products.This book assumes no programming background but offers a broad coverage of Java. It includes 159 numbered and numerous inline examples as well as 301 exercises categorized as conceptual, programming, and experiments. The problem-oriented approach presents a problem, explains supporting concepts, outlines necessary syntax, and finally provides its solution. All programs in the book are available for download and experimentation. A substantial index of 5,039 entries makes it easy for readers to locate relevant information.In a fast-changing field, this book is continually updated and refined. The 2013 version is the sixth draft edition of this volume, and features numerous revisions based on student feedback. This book is a must-have tutorial for software developers aiming to write concurrent programs in Scala, or broaden their existing knowledge of concurrency. This book is intended for Scala programmers that have no prior knowledge about concurrent programming, as well as those seeking to broaden their existing knowledge about concurrency. Basic knowledge of the Scala programming language will be helpful. Readers with a solid knowledge in another programming language, such as Java, should find this book easily accessible.

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.

Real-time functionality is essential for developing many consumer, industrial, and systems devices. While the C/C++ programming language is most often used in the creation of real-time software, the Java language, with its simple and familiar object-oriented programming model, offers many advantages over current real-time practices. Concurrent and Real-Time Programming in Java

covers the motivations for, and semantics of, the extensions and modifications to the Java programming environment that enable the Java platform (Virtual Machine) to meet the requirements and constraints of real-time development. Key aspects of concurrent and real-time programming and how they are implemented in Java are discussed, such as concurrency, memory management, real-time scheduling, and real-time resource sharing.

Threads are a fundamental part of the Java platform. As multicore processors become the norm, using concurrency effectively becomes essential for building high-performance applications. Java SE 5 and 6 are a huge step forward for the development of concurrent applications, with improvements to the Java Virtual Machine to support high-performance, highly scalable concurrent classes and a rich set of new concurrency building blocks. In Java Concurrency in Practice , the creators of these new facilities explain not only how they work and how to use them, but also the motivation and design patterns behind them. However, developing, testing, and debugging multithreaded programs can still be very difficult; it is all too easy to create concurrent programs that appear to work, but fail when it matters most: in production, under heavy load. Java Concurrency in Practice arms readers with both the theoretical underpinnings and concrete techniques for building reliable, scalable, maintainable concurrent applications. Rather than simply offering an inventory of concurrency APIs and mechanisms, it provides design rules, patterns, and mental models that make it easier to build concurrent programs that are both correct and performant. This book covers: Basic concepts of concurrency and thread safety Techniques for building and composing thread-safe classes Using the concurrency building blocks in java.util.concurrent Performance optimization dos and don'ts Testing concurrent programs Advanced topics such as atomic variables, nonblocking algorithms, and the Java Memory Model

Software development today is embracing functional programming (FP), whether it's for writing concurrent programs or for managing Big Data. Where does that leave Java developers? This concise book offers a pragmatic, approachable introduction to FP for Java developers or anyone who uses an object-oriented language. Dean Wampler, Java expert and author of Programming Scala (O'Reilly), shows you how to apply FP principles such as immutability, avoidance of side-effects, and higher-order functions to your Java code. Each chapter provides exercises to help you practice what you've learned. Once you grasp the benefits of functional programming, you'll discover that it improves all of the code you write. Learn basic FP principles and apply them to object-oriented programming Discover how FP is more concise and modular than OOP Get useful FP lessons for your Java type design—such as avoiding nulls Design data structures and algorithms using functional programming principles Write concurrent programs using the Actor model and software transactional memory Use functional libraries and frameworks for Java—and learn where to go next to deepen your functional programming skills

Master the art of fast, effective Java development with the power of concurrent and parallel programming About This Book Get detailed coverage of important recipes on multi-threading and parallel programming This book takes a close look at the Java 9 APIs and their impact on concurrency See practical examples on thread safety, high-performance classes, safe sharing, and a whole lot more Who This Book Is For The book is for Java developers and programmers at an intermediate to advanced level. It

will be especially useful for developers who want to take advantage of task-based recipes using Java 9's concurrent API to program thread-safe solutions. What You Will Learn Find out to manage the basic components of the Java Concurrency API Use synchronization mechanisms to avoid data race conditions and other problems of concurrent applications Separate the thread management from the rest of the application with the Executor framework Solve problems using a parallelized version of the divide and conquer paradigm with the Fork / Join framework Process massive data sets in an optimized way using streams and reactive streams See which data structures we can use in concurrent applications and how to use them Practice efficient techniques to test concurrent applications Get to know tips and tricks to design concurrent applications In Detail Writing concurrent and parallel programming applications is an integral skill for any Java programmer. Java 9 comes with a host of fantastic features, including significant performance improvements and new APIs. This book will take you through all the new APIs, showing you how to build parallel and multi-threaded applications. The book covers all the elements of the Java Concurrency API, with essential recipes that will help you take advantage of the exciting new capabilities. You will learn how to use parallel and reactive streams to process massive data sets. Next, you will move on to create streams and use all their intermediate and terminal operations to process big collections of data in a parallel and functional way. Further, you'll discover a whole range of recipes for almost everything, such as thread management, synchronization, executors, parallel and reactive streams, and many more. At the end of the book, you will learn how to obtain information about the status of some of the most useful components of the Java Concurrency API and how to test concurrent applications using different tools. Style and approach This recipe-based book will allow you to explore the exciting capabilities of concurrency in Java. After reading this book, you will be able to comfortably build parallel applications in Java 9. More than ever, learning to program concurrency is critical to creating faster, responsive applications. Speedy and affordable multicore hardware is driving the demand for high-performing applications, and you can leverage the Java platform to bring these applications to life. Concurrency on the Java platform has evolved, from the synchronization model of JDK to software transactional memory (STM) and actor-based concurrency. This book is the first to show you all these concurrency styles so you can compare and choose what works best for your applications. You'll learn the benefits of each of these models, when and how to use them, and what their limitations are. Through hands-on exercises, you'll learn how to avoid shared mutable state and how to write good, elegant, explicit synchronization-free programs so you can create easy and safe concurrent applications. The techniques you learn in this book will take you from dreading concurrency to mastering and enjoying it. Best of all, you can work with Java or a JVM language of your choice - Clojure, JRuby, Groovy, or Scala - to reap the growing power of multicore hardware. If you are a Java programmer, you'd need JDK 1.5 or later and the Akka 1.0 library. In addition, if you program in Scala, Clojure, Groovy or JRuby you'd need the latest version of your preferred language. Groovy programmers will also need GPars. Concurrency provides a thoroughly updated approach to the basic concepts and techniques behind concurrent programming. Concurrent programming is complex and demands a much more formal approach than sequential programming. In order to develop a thorough understanding of the topic Magee and Kramer present concepts, techniques and problems through a variety of forms: informal descriptions,

illustrative examples, abstract models and concrete Java examples. These combine to provide problem patterns and associated solution techniques which enable students to recognise problems and arrive at solutions. New features include: New chapters covering program verification and logical properties. More student exercises. Supporting website contains an updated version of the LTSA tool for modelling concurrency, model animation, and model checking. Website also includes the full set of state models, java examples, and demonstration programs and a comprehensive set of overhead slides for course presentation.

Master the principles and techniques of multithreaded programming with the Java 8 Concurrency API About This Book Implement concurrent applications using the Java 8 Concurrency API and its new components Improve the performance of your applications or process more data at the same time, taking advantage of all of your resources. Construct real-world examples related to machine learning, data mining, image processing, and client/server environments Who This Book Is For If you are a competent Java developer with a good understanding of concurrency but have no knowledge of how to effectively implement concurrent programs or use streams to make processes more efficient, then this book is for you. What You Will Learn Design concurrent applications by converting a sequential algorithm into a concurrent one Discover how to avoid all the possible problems you can get in concurrent algorithms Use the Executor framework to manage concurrent tasks without creating threads Extend and modify Executors to adapt their behavior to your needs Solve problems using the divide and conquer technique and the Fork/Join framework Process massive data sets with parallel streams and Map/Reduce implementation Control data-race conditions using concurrent data structures and synchronization mechanisms Test and monitor concurrent applications In Detail Concurrency programming allows several large tasks to be divided into smaller sub-tasks, which are further processed as individual tasks that run in parallel. All the sub-tasks are combined together once the required results are achieved; they are then merged to get the final output. The whole process is very complex. This process goes from the design of concurrent algorithms to the testing phase where concurrent applications need extra attention. Java includes a comprehensive API with a lot of ready-to-use components to implement powerful concurrency applications in an easy way, but with a high flexibility to adapt these components to your needs. The book starts with a full description of design principles of concurrent applications and how to parallelize a sequential algorithm. We'll show you how to use all the components of the Java Concurrency API from basics to the most advanced techniques to implement them in powerful concurrency applications in Java. You will be using real-world examples of complex algorithms related to machine learning, data mining, natural language processing, image processing in client / server environments. Next, you will learn how to use the most important components of the Java 8 Concurrency API: the Executor framework to execute multiple tasks in your applications, the phaser class to implement concurrent tasks divided into phases, and the Fork/Join framework to implement concurrent tasks that can be split into smaller problems (using the divide and conquer technique). Toward the end, we will cover the new inclusions in Java 8 API, the Map and Reduce model, and the Map and Collect model. The book will also teach you about the data structures and synchronization utilities to avoid data-race conditions and other critical problems. Finally, the book ends with a detailed description of the tools and techniques that you can use to test a Java concurrent application. Style and approach A complete guide implementing real-world examples with algorithms related to machine learning, data mining, and natural language processing in client/server environments. All the examples are explained in a step-by-step approach.

Java is one of the most popular programming languages for developing software applications, especially cloud computing, big data analytics, and various commonly used enterprise applications. Applications in these areas require high performance and scalability, driving unprecedented high demands for skills in Java concurrent programming. Whether you are a beginner or an experienced Java developer, it is

desirable to keep sharpening your Java programming skills with a systematic understanding of the power and flexibility that the modern Java concurrent programming frameworks offer. This book attempts to provide an easier entry into Java concurrent programming for those who are passionate about programming, especially for those who are motivated and determined to develop high-performance and scalable Java software. Specifically, this book covers the following subjects: * Chapter 1 Multithreaded Programming in Java * Chapter 2 Java ExecutorService Framework * Chapter 3 Java Collections Framework * Chapter 4 Atomic Operations * Chapter 5 Locks * Chapter 6 Synchronizers * Chapter 7 Synchronized Collections * Chapter 8 Parallel Programming Using the Fork-Join Framework * Appendix A Algorithm Analysis * Appendix B The Bridge Exercise To minimize the possibilities of any kind of misperceptions and misunderstandings, this text emphasizes understanding how various classes are actually implemented internally, accompanied with carefully-crafted examples. A great additional benefit out of this approach is that it gives all of us an opportunity to appreciate how those masters coded all of those classes that we use every day for our Java concurrent programming tasks. This book will not only enhance your Java concurrent programming skills specifically but also Java programming skills in general. In addition, you will learn how to judge properly on whether threads are running normally or in an undesirable livelock or starvation or deadlock situation.

If you have a working knowledge of Haskell, this hands-on book shows you how to use the language's many APIs and frameworks for writing both parallel and concurrent programs. You'll learn how parallelism exploits multicore processors to speed up computation-heavy programs, and how concurrency enables you to write programs with threads for multiple interactions. Author Simon Marlow walks you through the process with lots of code examples that you can run, experiment with, and extend. Divided into separate sections on Parallel and Concurrent Haskell, this book also includes exercises to help you become familiar with the concepts presented: Express parallelism in Haskell with the Eval monad and Evaluation Strategies Parallelize ordinary Haskell code with the Par monad Build parallel array-based computations, using the Repa library Use the Accelerate library to run computations directly on the GPU Work with basic interfaces for writing concurrent code Build trees of threads for larger and more complex programs Learn how to build high-speed concurrent network servers Write distributed programs that run on multiple machines in a network

In this second edition, you will find thoroughly updated coverage of the Javao 2 platform and new or expanded coverage of: Memory model Cancellation Portable parallel programming Utility classes for concurrency control The Java platform provides a broad and powerful set of APIs, tools, and technologies. One of its most powerful capabilities is the built-in support for threads. This makes concurrent programming an attractive yet challenging option for programmers using the Java programming language. This book shows readers how to use the Java platform's threading model more precisely by helping them to understand the patterns and tradeoffs associated with concurrent programming. You will learn how to initiate, control, and coordinate concurrent activities using the class java.lang. Thread, the keywords synchronized and volatile, and the methods wait, notify, and notifyAll. In addition, you will find detailed coverage of all aspects of concurrent programming, including such topics as confinement and synchronization, deadlocks and conflicts, state-dependent action control, asynchronous message passing and control flow, coordinated interaction, and structuring web-based and computational services. The book targets intermediate to advanced programmers interested in mastering the complexities of concurrent programming. Taking a design pattern approach, the book offers standard design techniques for creating and implementing components that solve common concurrent programming challenges. The numerous code examples throughout help clarify the subtleties of the concurrent programming concepts discussed. 0201310090B04062001. Explains how to use Java's portable platforms to program and use threads effectively and efficiently while avoiding common mistakes

Software -- Programming Languages.

Looks at the concepts and techniques of concurrent programming.

A complete description of Erlang, a programming language for building robust concurrent systems. The book contains many examples of how robust real-time systems can be programmed using this language.

Java Thread Programming shows you how to take full advantage of Java's thread facilities: when to use threads to increase your program's efficiency, how to use them effectively, and how to avoid common mistakes. There is thorough coverage of the Thread API, ThreadGroup classes, the Runnable interface, and the synchronized operator. Extensive, complete, code examples show programmers the details of creating and managing threads in real-world applications.

JR is an extension of the Java programming language with additional concurrency mechanisms based on those in the SR (Synchronizing Resources) programming language. The JR implementation executes on UNIX-based systems (Linux, Mac OS X, and Solaris) and Windows-based systems. It is available free from the JR webpage. This book describes the JR programming language and illustrates how it can be used to write concurrent programs for a variety of applications. This text presents numerous small and large example programs. The source code for all programming examples and the given parts of all programming exercises are available on the JR webpage. Dr. Ronald A. Olsson and Dr. Aaron W. Keen, the authors of this text, are the designers and implementors of JR.

A definitive guide to mastering and implementing concurrency patterns in your applications Key Features Build scalable apps with patterns in multithreading, synchronization, and functional programming Explore the parallel programming and multithreading techniques to make the code run faster Efficiently use the techniques outlined to build reliable applications Book Description Selecting the correct concurrency architecture has a significant impact on the design and performance of your applications. This book explains how to leverage the different characteristics of parallel architecture to make your code faster and more efficient. To start with, you'll understand the basic concurrency concepts and explore patterns around explicit locking, lock free programming, futures & actors. Then, you'll get insights into different concurrency models and parallel algorithms and put them to practice in different scenarios to realize your application's true potential. We'll take you through multithreading design patterns, such as master, slave, leader, follower, map-reduce, and monitor, also helping you to learn hands-on coding using these patterns. Once you've grasped all of this, you'll move on to solving problems using synchronizer patterns. You'll discover the rationale for these patterns in distributed & parallel applications, followed by studying how future composition, immutability and the monadic flow help create more robust code. Toward the end of the book, you'll learn about the actor paradigm and actor patterns - the message passing concurrency paradigm. What you will learn Explore parallel architecture Get acquainted with concurrency models Internalize design themes by implementing multithreading patterns Get insights into concurrent design patterns Discover design principles behind many java threading abstractions Work with functional concurrency patterns Who this book is for This is a must-have guide for developers who want to learn patterns to build scalable and high-performing apps. It's assumed that you already have a decent level of programming knowledge.

A thorough and self-contained account. Use either as textbook or as a primary reference work.

Learn the basics of parallel programming in Java to write more efficient, performant code.

Master the principles to make applications robust, scalable and responsive About This Book Implement concurrent applications using the Java 9 Concurrency API and its new components Improve the performance of your applications and process more data at the same time,

taking advantage of all of your resources Construct real-world examples related to machine learning, data mining, natural language processing, and more Who This Book Is For This book is for competent Java developers who have basic understanding of concurrency, but knowledge of effective implementation of concurrent programs or usage of streams for making processes more efficient is not required What You Will Learn Master the principles that every concurrent application must follow See how to parallelize a sequential algorithm to obtain better performance without data inconsistencies and deadlocks Get the most from the Java Concurrency API components Separate the thread management from the rest of the application with the Executor component Execute phased-based tasks in an efficient way with the Phaser components Solve problems using a parallelized version of the divide and conquer paradigm with the Fork / Join framework Find out how to use parallel Streams and Reactive Streams Implement the "map and reduce" and "map and collect" programming models Control the concurrent data structures and synchronization mechanisms provided by the Java Concurrency API Implement efficient solutions for some actual problems such as data mining, machine learning, and more In Detail Concurrency programming allows several large tasks to be divided into smaller sub-tasks, which are further processed as individual tasks that run in parallel. Java 9 includes a comprehensive API with lots of ready-to-use components for easily implementing powerful concurrency applications, but with high flexibility so you can adapt these components to your needs. The book starts with a full description of the design principles of concurrent applications and explains how to parallelize a sequential algorithm. You will then be introduced to Threads and Runnables, which are an integral part of Java 9's concurrency API. You will see how to use all the components of the Java concurrency API, from the basics to the most advanced techniques, and will implement them in powerful real-world concurrency applications. The book ends with a detailed description of the tools and techniques you can use to test a concurrent Java application, along with a brief insight into other concurrency mechanisms in JVM. Style and approach This is a complete guide that implements real-world examples of algorithms related to machine learning, data mining, and natural language processing in client/server environments. All the examples are explained using a step-by-step approach.

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.

"Clojure programming ... This functional programming language not only lets you take advantage of Java libraries, services, and other JVM resources, it rivals other dynamic languages such as Ruby and Python. With this comprehensive guide, you'll learn Clojure fundamentals with examples that relate it to languages you already know"--P. [4] of cover. Concurrent Programming in JavaDesign Principles and PatternsAddison-Wesley Professional "Multithreaded Programming with Java Technology is the first complete guide to multithreaded development with the Java 2 platform. Multithreading experts Bil Lewis and Daniel J. Berg cover the underlying structures upon which threads are built; thread construction; and thread lifecycles, including birth, life, death, and cancellation. Next, using extensive code examples, they cover everything developers need to know to make the most of multithreading."--BOOK JACKET.Title Summary field provided by Blackwell North America, Inc. All Rights Reserved

When it comes to choosing, using, and maintaining a database, understanding its internals is essential. But with so many distributed databases and tools available today, it's often difficult to understand what each one offers and how they differ. With this practical guide, Alex Petrov guides developers through the concepts behind modern database and storage engine internals. Throughout the book, you'll explore relevant material gleaned from numerous books, papers, blog posts, and the source code of several open source databases. These resources are listed at the end of parts one and two. You'll discover that the most significant distinctions among many modern databases reside in subsystems that determine how storage is organized and how data is distributed. This book examines: Storage engines: Explore storage classification and taxonomy, and dive into B-Tree-based and immutable Log Structured storage engines, with differences and use-cases for each Storage building blocks: Learn how database files are organized to build efficient storage, using auxiliary data structures such as Page Cache, Buffer Pool and Write-Ahead Log Distributed systems: Learn step-by-step how nodes and processes connect and build complex communication patterns Database clusters: Which consistency models are commonly used by modern databases and how distributed storage systems achieve consistency

Copyright: 1b1654bcb5471623b341c60f0666434d