Java 2 Performance And Idiom Guide

Server-Side Development Is Easier Than You Think Not so long ago, anyone who knew HTML, even casually, was considered to have a special skill. Now, if you don't at least dabble in server-side web development, you're already behind. Fortunately, Mastering JSP is a great way to build the skills you need today. Inside, you'll learn to design and develop a wide range of JSP-based web applications, beginning with a basic, dynamically generated website. From there you'll build apps that read from and write to databases, create your own custom tags, and process and present XML. Throughout, you're helped by detailed, completely illuminated examples. Coverage includes: Making a servlet configurable Using JavaBeans inside Java Server Pages Building basic JavaBeans for storing user preferences and utility functions Recovering source code from binary class files Building a basic database application Handling exceptions and using asserts Redirecting error output to a file Scanning error logs using a servlet Using a pop-up menu to avoid troublesome actions Displaying XML files using XSLT and JSP Making a servlet-based watermarking application Providing dynamic authentication with roles, permissions, and access Building a Model-View Controller application Improving prototyping with ResultSets using a custom class Abstracting data access code using Java Objects Creating a custom tag to handle client-side JavaScript Abstracting data access SQL using meta-data, JavaBeans, and EJB Note: CD-ROM/DVD and other supplementary materials are not included

as part of eBook file.

The performance of software components depends on several factors, including the execution platform on which the software components run. To simplify cross-platform performance prediction in relocation and sizing scenarios, a novel approach is introduced in this thesis which separates the application performance profile from the platform performance profile. The approach is evaluated using transparent instrumentation of Java applications and with automated benchmarks for Java Virtual Machines.

The "Writing Idiomatic Python" book is finally here! Chock full of code samples, you'll learn the "Pythonic" way to accomplish common tasks. Each idiom comes with a detailed description, example code showing the "wrong" way to do it, and code for the idiomatic, "Pythonic" alternative. *This version of the book is for Python 3.3+. There is also a Python 2.7.3+ version available.* "Writing Idiomatic Python" contains the most common and important Python idioms in a format that maximizes identification and understanding. Each idiom is presented as a recommendation to write some commonly used piece of code. It is followed by an explanation of why the idiom is important. It also contains two code samples: the "Harmful" way to write it and the "Idiomatic" way. * The "Harmful" way helps you identify the idiom in your own code. * The "Idiomatic" way shows you how to easily translate that code into idiomatic Python. This book is perfect for you: * If you're coming to Python from another programming language * If you're learning Python as a first programming language * If you're looking to increase the readability,

maintainability, and correctness of your Python code What is "Idiomatic" Python? Every programming language has its own idioms. Programming language idioms are nothing more than the generally accepted way of writing a certain piece of code. Consistently writing idiomatic code has a number of important benefits: * Others can read and understand your code easily * Others can maintain and enhance your code with minimal effort * Your code will contain fewer bugs * Your code will teach others to write correct code without any effort on your part

Java 2 Performance and Idiom GuidePrentice Hall PTR

Idioms, Pitfalls, Styles, and Programming Tips

Java Report

Inside Java 2 Platform Security

Server Component Patterns

An Introduction to Object-oriented Analysis and Design and the Unified Process

Expert One-on-One J2EE Design and Development

Software reuse promises high value to businesses that develop software, opening the door to radical improvements in productivity, cost, and time to market. This book is for those who are wondering whether they should adopt reuse and how, and also for those who have already started to adopt it but are wondering where they may be going wrong and how they could do better. It emphasizes the practical issues that influence success or failure in reuse; and offers a concise and balanced coverage of the essentials.

Der Weg von der Inbetriebnahme eines Prozessorsystems bis zur Implementierung einer Human Machine Interface (HMI) bildet den Schwerpunkt dieses Werks. Der Autor erl ä utert, wie Treiber und Betriebssystem (QNX, Linux) konfiguriert, gebaut und geladen werden. Alle notwendigen Kenntnisse werden systematisch und fundiert vermittelt, auch Fragen der Virtualisierung und der Einsatz von MultiCore-Systemen. Der Band enth ä lt praktische Beispiele sowie Anleitungen f ü r die Fehlersuche und die Performance-Optimierung, Code-Snippets werden zur Verf ü gung gestellt.

Initially, computer systems performance analyses were carried out primarily because of limited resources. Due to ever increasing functional complexity of computational systems and user requirements, performance engineering continues to play a major role in software development. This book assesses the state of the art in performance engineering. Besides revised chapters drawn from two workshops on performance engineering held in 2000, additional chapters were solicited in order to provide complete coverage of all relevant aspects. The first part is devoted to the relation between software engineering and performance engineering; the second part focuses on the use of models, measures, and tools; finally, case studies with regard to concrete technologies are presented. Researchers, professional software engineers, and advanced

students interested in performance analysis will find this book an indispensable source of information and reference.

This book, written by one of the designers of generics, is a thorough explanation of how to use generics, and particularly, the effect this facility has on the way developers use collections.

Getting the Most Out of Your Code

Embedded Technologies

MSDN Magazine

A Study in Cultural History

How Tests Drive the Code

Celebrating Identity, Constructing Community

An update to the bestselling UML classic, this title has been revised to cover the unified process and Rational Software's processes. Larman also shows developers how to make practical use of the most significant recent developments in object-oriented analysis and design.

This book introduces the advanced features of Java. Among these are OO design and analysis of Java programs, implementing callbacks, enhancing the Java toolkit, meta-programming in Java, security, multiple threads, 3D imaging, and access to third party software.

Examples from different regions, of varied genres, illustrate how contemporary performance participates in and gives expression to the complex social changes taking place in Indonesia

today.

Music and queerness interact in many different ways. The Oxford Handbook of Music and Queerness brings together many topics and scholarly disciplines, reflecting the diversity of current research and methodology. Each of the book's six sections exemplifies a particular rhetoric of queer music studies. The section "Kinds of Music" explores queer interactions with specific musics such as EDM, hip hop, and country. "Versions" explores queer meanings that emerge in the creation of a version of a pre-existing text, for instance in musical settings of Biblical texts or practices of karaoke. "Voices and Sounds" turns in various ways to the materiality of music and sound. "Lives" focuses on interactions of people's lives with music and queerness. "Histories" addresses moments in the past, beginning with times when present conceptualizations of sexuality had not yet developed and moving to cases studies of more recent history, including the creation of pop songs in response to HIV/AIDS and the Eurovision song contest. The final section, "Cross-cultural Queerness," asks how to understand gender and sexuality in locations where recent Euro-American concepts may not be appropriate.

Applying UML and Patterns

Java Performance Tuning

Java to Kotlin

Just Java 2

EU Cost Action 253

Unit Testing in Java

Helps readers eliminate performance problems, covering topics including bottlenecks, profiling tools, strings, algorithms,

distributed systems, and servlets.

What is this book about? The results of using J2EE in practice are often disappointing: applications are often slow, unduly complex, and take too long to develop. Rod Johnson believes that the problem lies not in J2EE itself, but in that it is often used badly. Many J2EE publications advocate approaches that, while fine in theory, often fail in reality, or deliver no real business value. Expert One-on-One: J2EE Design and Development aims to demystify J2EE development. Using a practical focus, it shows how to use J2EE technologies to reduce, rather than increase, complexity. Rod draws on his experience of designing successful high-volume J2EE applications and salvaging failing projects, as well as intimate knowledge of the J2EE specifications, to offer a real-world, how-to guide on how you too can make J2EE work in practice. It will help you to solve common problems with J2EE and avoid the expensive mistakes often made in J2EE projects. It will guide you through the complexity of the J2EE services and APIs to enable you to build the simplest possible solution, on time and on budget. Rod takes a practical, pragmatic approach, questioning J2EE orthodoxy where it has failed to deliver results in practice and instead suggesting

effective, proven approaches. What does this book cover? In this book, you will learn When to use a distributed architecture When and how to use EJB How to develop an efficient data access strategy How to design a clean and maintainable web interface How to design J2EE applications for performance Who is this book for? This book would be of value to most enterprise developers. Although some of the discussion (for example, on performance and scalability) would be most relevant to architects and lead developers, the practical focus would make it useful to anyone with some familiarity with J2EE. Because of the complete design-deployment coverage, a less advanced developer could work through the book along with a more introductory text, and successfully build and understand the sample application. This comprehensive coverage would also be useful to developers in smaller organisations, who might be called upon to fill several normally distinct roles. What is special about this book? Wondering what differentiates this book from others like it in the market? Take a look: It does not just discuss technology, but stress its practical application. The book is driven from the need to solve common tasks, rather than by the elements of J2EE. It discuss risks in J2EE

development It takes the reader through the entire design, development and build process of a non-trivial application. This wouldn't be compressed into one or two chapters, like the Java Pet Store, but would be a realistic example comparable to the complexity of applications readers would need to build. At each point in the design, alternative choices would be discussed. This would be important both where there's a real problem with the obvious alternative, and where the obvious alternatives are perhaps equally valid. It emphasizes the use of OO design and design patterns in J2EE, without becoming a theoretical book Choice Outstanding Academic Title, 2008. The uniqueness of shape as a perceptual property lies in the fact that it is both complex and structured. Shapes are perceived veridically—perceived as they really are in the physical world, regardless of the orientation from which they are viewed. The constancy of the shape percept is the sine qua nonof shape perception; you are not actually studying shape if constancy cannot be achieved with the stimulus you are using. Shape is the only perceptual attribute of an object that allows unambiguous identification. In this first book devoted exclusively to the perception of shape by humans and machines,

Zygmunt Pizlo describes how we perceive shapes and how to design machines that can see shapes as we do. He reviews the long history of the subject, allowing the reader to understand why it has taken so long to understand shape perception, and offers a new theory of shape. Until recently, shape was treated in combination with such other perceptual properties as depth, motion, speed, and color. This resulted in apparently contradictory findings, which made a coherent theoretical treatment of shape impossible. Pizlo argues that once shape is understood to be unique among visual attributes and the perceptual mechanisms underlying shape are seen to be different from other perceptual mechanisms, the research on shape becomes coherent and experimental findings no longer seem to contradict each other. A single theory of shape perception is thus possible, and Pizlo offers a theoretical treatment that explains how a three-dimensional shape percept is produced from a two-dimensional retinal image, assuming only that the image has been organized into two-dimensional shapes. Pizlo focuses on discussion of the main concepts, telling the story of shape without interruption. Appendixes provide the basic mathematical and computational information necessary for a technical understanding

of the argument. References point the way to more in-depth reading in geometry and computational vision.

This authoritative Java security book is written by the architect of the Java security model. It chronicles J2EE v1.4 security model enhancements that will allow developers to build safer, more reliable, and more impenetrable programs.

Advanced Programming for the Java 2 Platform

Java Generics and Collections

Effective Java

Component Infrastructures Illustrated with EJB

A developer's guide to MIDP 2.0

Java Concurrency in Practice

It takes a week to travel the 8,000 miles overland from Java to Kotlin. If you're an experienced Java developer who has tried the Kotlin language, you were probably productive in about the same time. You'll have found that they do things differently in Kotlin, though. Nullability is important, collections are different, and classes are final by default. Kotlin is more functional, but what does that mean, and how should it change the way that you program? And what about all that Java code that you still have to support? Your tour guides Duncan and Nat first made the trip in 2015, and

they've since helped many teams and individuals follow in their footsteps. Travel with them as they break the route down into legs like Optional to Nullable, Beans to Values, and Open to Sealed Classes. Each explains a key concept and then shows how to refactor production Java to idiomatic Kotlin, gradually and safely, while maintaining interoperability. The resulting code is simpler, more expressive, and easier to change. By the end of the journey, you'll be confident in refactoring Java to Kotlin, writing Kotlin from scratch, and managing a mixed language codebase as it evolves over time.

Are you looking for a deeper understanding of the Java™ programming language so that you can write code that is clearer, more correct, more robust, and more reusable? Look no further! Effective Java™, Second Edition, brings together seventy-eight indispensable programmer's rules of thumb: working, best-practice solutions for the programming challenges you encounter every day. This highly anticipated new edition of the classic, Jolt Award-winning work has been thoroughly updated to cover Java SE 5 and Java SE 6 features introduced since the first edition. Bloch explores new design patterns and language idioms, showing you how to make the most of features ranging from generics to enums, annotations to autoboxing. Each chapter in the book consists of several "items" presented in the form of a short, standalone essay that provides specific advice, insight into Java

platform subtleties, and outstanding code examples. The comprehensive descriptions and explanations for each item illuminate what to do, what not to do, and why. Highlights include: New coverage of generics, enums, annotations, autoboxing, the for-each loop, varargs, concurrency utilities, and much more Updated techniques and best practices on classic topics, including objects, classes, libraries, methods, and serialization How to avoid the traps and pitfalls of commonly misunderstood subtleties of the language Focus on the language and its most fundamental libraries: java.lang, java.util, and, to a lesser extent, java.util.concurrent and java.io Simply put, Effective Java™, Second Edition, presents the most practical, authoritative guidelines available for writing efficient, well-designed programs.
DepCoS – RELCOMEX is an annual series of conferences organized by Wrocław University of Technology to promote a comprehensive approach to evaluation of system performability which is now commonly called dependability. In contrast to classic analyses which were concentrated on reliability of technical resources and structures built from them, dependability is based on multi-disciplinary approach to theory, technology and maintenance of a system considered to be a multifaceted amalgamation of technical, information, organization, software and human (users, administrators, supervisors, etc.) resources. Diversity of processes being

realized (data processing, system management, system monitoring, etc.), their concurrency and their reliance on in-system intelligence often severely impedes construction of strict mathematical models and calls for application of intelligent and soft computing methods. This book presents the proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, which took place in Brunów Palace, Poland, from 30th June to 4th July, 2014. The articles selected for this volume illustrate the variety of topics that must be included in system dependability analysis: tools, methodologies and standards for modelling, design and simulation of the systems, security and confidentiality in information processing, specific issues of heterogeneous, today often wireless, computer networks or management of transportation networks.
Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No

matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor—or evolve—code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

Java 2 Performance and Idiom Guide

Architecture, API Design, and Implementation

NetBeans: The Definitive Guide

State of the Art and Current Trends

Java Performance: The Definitive Guide

Essential JTAPI: Java Telephony API

A Local Area Network (LAN) is a network usually within a single office or building that links desktop computers with each other and with peripherals such as servers and printers. The interconnect is the electrical and functional association of two different services, often provided by different suppliers, and it is from LAN inter-connection that telecoms operators

seek to profit. The application of LAN interconnection via satellite can be used to complement and extend existing terrestrial public access networks through interconnection of clusters of broadband islands (such as LANs and MANs) in remote regions, where terrestrial lines are expensive to install and operate. Examples include: * Hospitals/clinics in remote and rural areas can be connected to the central hospitals in a tele-medicine environment * Remote offices can be connected to the central office to facilitate tele-working * University/colleges can be inter-connected to provide tele-education facilities Similarly, the possibility to provide access to such facilities in developing regions of the world is also viable and particularly attractive in the short to mid-term. Private LAN connection facilities could also be made available to the corporate user, offering the possibility to establish broadband internet access within a closed user group. Such a scenario could be of interest to the financial sector. By gathering the knowledge and experiences of well-known satellite systems experts from different parts of Europe this comprehensive volume provides detailed analysis on technical aspects for interconnecting local area network using satellite. Starting from traffic source modelling for different types of applications and services to different types of transmission techniques and networking functions for supporting such services, different case studies are presented to analyse the performance of such technologies. By providing an insight to current and future developments in satellite communications systems and by covering a broad range of materials in technical aspects in relation to satellite communication systems technologies, this volume will be of tremendous use to researchers, academia and industry. * First book to

present such a thorough description of the reliability functions of satellite systems * Discusses IP over satellite * Provides a unique analysis and description of different simulation tools that are under development for evaluating the performance of satellite systems * Includes a chapter devoted to traffic modelling for satellite systems * Reviews current research and developments in security and discusses how such security functions can be implemented over satellite networks * Addresses different types of routing strategies and includes three different case studies which have been carried out to analyse the performance of different routing strategies

Coding and testing are often considered separate areas of expertise. In this comprehensive guide, author and Java expert Scott Oaks takes the approach that anyone who works with Java should be equally adept at understanding how code behaves in the JVM, as well as the tunings likely to help its performance. You'll gain in-depth knowledge of Java application performance, using the Java Virtual Machine (JVM) and the Java platform, including the language and API. Developers and performance engineers alike will learn a variety of features, tools, and processes for improving the way Java 7 and 8 applications perform. Apply four principles for obtaining the best results from performance testing Use JDK tools to collect data on how a Java application is performing Understand the advantages and disadvantages of using a JIT compiler Tune JVM garbage collectors to affect programs as little as possible Use techniques to manage heap memory and JVM native memory Maximize Java threading and synchronization performance features Tackle performance issues in Java

EE and Java SE APIs Improve Java-driven database application performance
Software testing is indispensable and is one of the most discussed topics in software development today. Many companies address this issue by assigning a dedicated software testing phase towards the end of their development cycle. However, quality cannot be tested into a buggy application. Early and continuous unit testing has been shown to be crucial for high quality software and low defect rates. Yet current books on testing ignore the developer's point of view and give little guidance on how to bring the overwhelming amount of testing theory into practice. Unit Testing in Java represents a practical introduction to unit testing for software developers. It introduces the basic test-first approach and then discusses a large number of special issues and problem cases. The book instructs developers through each step and motivates them to explore further. Shows how the discovery and avoidance of software errors is a demanding and creative activity in its own right and can build confidence early in a project. Demonstrates how automated tests can detect the unwanted effects of small changes in code within the entire system. Discusses how testing works with persistency, concurrency, distribution, and web applications. Includes a discussion of testing with C++ and Smalltalk.
PLEASE PROVIDE COURSE INFORMATION PLEASE PROVIDE
Design Telecom Projects with Java
Book Review Index
Journal of Object-oriented Programming
Java in the 14th Century

A Master Cumulation
Programming Java 2 Micro Edition for Symbian OS
Vols. 8-10 of the 1965-1984 master cumulation constitute a title index.
"... engaging overview of Java 2 standard edition (J2SE 1.5) .... on back cover.
This is the authoritative reference for understanding and using the NetBeans Integrated Development Environment for creating new software with Java. Contains a detailed tutorial.
Hands-on information to help you fully exploit the capabilities of MIDP 2.0 on Symbian OS (including MMA, WMA and Bluetooth). This practical guide will walk you through developing example applications illustrating key functionality and explain how to install these applications onto real devices. Focuses on J2ME MIDP 1.0 and 2.0, as this platform has become the Java standard for phones Covers the optional J2ME APIs that Symbian OS Java is currently supporting Code samples are provided throughout Contains case studies that demonstrate how to develop games and enterprise applications
Mastering JSP

Practical Software Reuse
Quantifying and Predicting the Influence of Execution Platform on Software Component Performance
Service Efficient Network Interconnection Via Satellite
Writing Idiomatic Python 3.3
Software -- Programming Languages.

* Full analysis of performance characteristics of the .NET Framework, including actual benchmark results * Information on the internals of the .NET Framework and exposure to the various elements that make up the .NET Framework * Description of tools and techniques for identifying performance problems developers may encounter * References to sources of further information on various performance topics * Written by a Microsoft MVP in a technically unique style and of the highest quality

A detailed exploration of the basic patterns underlying today's component infrastructures. The latest addition to this best-selling series opens by providing an "Alexandrian-style" pattern language covering the patterns underlying EJB, COM+ and CCM. It addresses not only the underlying building blocks, but also how they interact and why they are used. The

second part of the book provides more detail about how these building blocks are employed in EJB. In the final section the authors fully explore the benefits of building a system based on components. * Examples demonstrate how the 3 main component infrastructures EJB, CCM and COM+ compare * Provides a mix of principles and concrete examples with detailed UML diagrams and extensive source code * Forewords supplied by industry leaders: Clemens Syzperski and Frank Buschmann

This book presents a set of 11 papers accompanying the lectures of leading researchers given at the 7th edition of the International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2007, held in Bertinoro, Italy in May/June 2007. SFM 2007 was devoted to formal techniques for performance evaluation and covered several aspects of the field.

Software Development

Advanced Java

Java/Jini Technologies

Formal Methods for Performance Evaluation

7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2007, Bertinoro, Italy, May

8-June 2, 2007, Advanced Lectures

Vom Treiber bis zur Grafik-Anbindung

The present second volume of the Nägara-Kertägama edition contains those notes on the tex.t and the translation of the major poem and the appended minor writings and charters that may be of interest to students of the J avanese language. Perusal of these notes on idiom and linguistics will be found the more fruitful if the glossary (volume V of the present edition) is consulted continuously. No lengthy discussions of linguistic subjects are to be found in the present volume. They would be out of place in this new Nägara Kertägama edition, for its tenor is primarily sociological. Exceptions have been made only for some places where short grammatical discus sions seemed in place in order to elucidate points of the translation. In the notes on the contemporaneous minor writings and charters differences between the scholarly idiom of Court literature (Nägara Kertägama and Royal charters) on the one side and the popular ver nacular idiom of daily life on the other have been pointed out repeatedly. The close relationship of the 14th century Majapahit vemacular with modern J avanese is apparent. Occasional remarks on words betonging to regional idioms, either Eastem Javanese Majapahit or Ka

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

21-22 August 2001, Denver, USA

American Book Publishing Record

Performing Contemporary Indonesia

The British National Bibliography

Maximizing .NET Performance

Code Complete