

Javacc Documentation

This book presents the latest research findings, methods and development techniques related to Ubiquitous and Pervasive Computing (UPC) as well as challenges and solutions from both theoretical and practical perspectives with an emphasis on innovative, mobile and internet services. With the proliferation of wireless technologies and electronic devices, there is a rapidly growing interest in Ubiquitous and Pervasive Computing (UPC). UPC makes it possible to create a human-oriented computing environment where computer chips are embedded in everyday objects and interact with physical world. It also allows users to be online even while moving around, providing them with almost permanent access to their preferred services. Along with a great potential to revolutionize our lives, UPC also poses new research challenges.

What others in the trenches say about *The Pragmatic Programmer*... “The cool thing about this book is that it’s great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there.” —Kent Beck, author of *Extreme Programming Explained: Embrace Change* “I found this book to be a great mix of solid advice and wonderful analogies!” —Martin Fowler, author of *Refactoring* and *UML Distilled* “I would buy a copy, read it twice, then tell

all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost.” —Kevin Ruland, Management Science, MSG-Logistics “The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike.” —John Lakos, author of *Large-Scale C++ Software Design* “This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients.” —Eric Vought, Software Engineer “Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book.” —Pete McBreen, Independent Consultant “Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop

reference for everyone who works with code for a living.” —Jared Richardson, Senior Software Developer, iRenaissance, Inc. “I would like to see this issued to every new employee at my company....” —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. “If I’m putting together a project, it’s the authors of this book that I want. . . . And failing that I’d settle for people who’ve read their book.” —Ward Cunningham

Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting

analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

The mission of the Association of Geographic Information Laboratories for Europe (AGILE) is to promote academic teaching and research at the European level, and to facilitate networking activities between geographic information laboratories, including focused meetings based on state-of-the-art presentations on key research issues and European geographic information research conferences. The AGILE Conferences on Geographic Information Science (GIS) have become an essential meeting place for European researchers and practitioners, where they meet and exchange ideas and experiences at the European level. These proceedings regroup the papers given in the Lyon conference held in April 2003 and presenting the more advanced results in GIS.

This book constitutes the refereed proceedings of the Second International Conference on Research in

Smart Cards, E-smart 2001, held in Cannes, France, in September 2001. The 20 revised full papers presented were carefully reviewed and selected from 38 submissions. Among the topics addressed are biometrics, cryptography and electronic signatures on smart card security, formal methods for smart card evaluation and certification, architectures for multi-applications and secure open platforms, and middleware for smart cards and novel applications of smart cards.

This volume is the eighth in the Intelligent Agents series associated with the ATAL workshops. These workshops on “Agent Theories, Architectures, and Languages” have established themselves as a tradition, and play the role of small but internationally well-known conferences on the subject, where besides theory per se also integration of theory and practice is in focus. Specifically, ATAL addresses issues of theories of agency, software architectures for intelligent agents, methodologies and programming languages for realizing agents, and software tools for applying and evaluating agent-based systems. ATAL 2001 featured two special tracks in which both the more theoretical / formal and the more practical aspects were present, viz. “Formal Theories of Negotiation”, organized by Frank Dignum, and “Agents for Hand-Held, Mobile, or Embedded Devices”, organized by Tim Finin. There was also an extra session on RoboCup

Rescue, organized and presented by Satoshi Tadokoro and Ranjit Nair. ATAL 2001 attracted 68 papers from over 20 countries all over the world, of which 30 were selected for presentation at the workshop and publication in this volume. We invited two outstanding speakers: Fausto Giunchiglia (Trento, Italy) and Tom Dean (Brown, USA). Formal methods are coming of age. Mathematical techniques and tools are now regarded as an important part of the development process in a wide range of industrial and governmental organisations. A transfer of technology into the mainstream of systems development is slowly, but surely, taking place. FM'99, the First World Congress on Formal Methods in the Development of Computing Systems, is a result, and a measure, of this new-found maturity. It brings an impressive array of industrial and applications-oriented papers that show how formal methods have been used to tackle real problems. These proceedings are a record of the technical symposium of FM'99: alongside the papers describing applications of formal methods, you will find technical reports, papers, and abstracts detailing new advances in formal techniques, from mathematical foundations to practical tools. The World Congress is the successor to the four Formal Methods Europe Symposia, which in turn succeeded the four VDM Europe Symposia. This session reflects an increasing openness within the

international community of researchers and practitioners: papers were submitted covering a wide variety of formal methods and application areas. The programme committee reflects the Congress's international nature, with a membership of 84 leading researchers from 38 different countries. The committee was divided into 19 tracks, each with its own chair to oversee the reviewing process. Our collective task was a difficult one: there were 259 high-quality submissions from 35 different countries. CSMR 2004 covers a wide range of ongoing research activities in the area of software maintenance, reengineering, and software evolution. The papers examine theory and practice in the areas of software maintenance, architecture design recovery, software evolution, reverse engineering, web applications, dynamic analysis, historical analysis, product families and experience reports. Immersing students in Java and the Java Virtual Machine (JVM), Introduction to Compiler Construction in a Java World enables a deep understanding of the Java programming language and its implementation. The text focuses on design, organization, and testing, helping students learn good software engineering skills and become better programmers. The book covers all of the standard compiler topics, including lexical analysis, parsing, abstract syntax trees, semantic analysis, code generation, and register allocation. The authors also demonstrate how JVM code can be translated to a register machine, specifically the MIPS architecture. In addition, they discuss recent strategies, such as just-in-time compiling and hotspot compiling, and present an overview of

leading commercial compilers. Each chapter includes a mix of written exercises and programming projects. By working with and extending a real, functional compiler, students develop a hands-on appreciation of how compilers work, how to write compilers, and how the Java language behaves. They also get invaluable practice working with a non-trivial Java program of more than 30,000 lines of code. Fully documented Java code for the compiler is accessible at <http://www.cs.umb.edu/j--/>

This book contributes the thoroughly refereed post-conference proceedings of the 6th International Conference on Web-Based Learning, ICWL 2007, held in Edinburgh, UK, in August 2007. The 55 revised full papers presented together with 1 keynote talk were carefully reviewed and selected from about 180 submissions. The papers are organized in topical sections on personalized e-learning, learning resource organization and management, framework and standards for e-learning, test authoring, question generation and assessment, language learning, science education, visualization technologies for content delivery and learning behavior, practice and experience sharing, security, privacy and mobile e-learning, as well as blended learning. The LNCS journal Transactions on Computational Science reflects recent developments in the field of Computational Science, conceiving the field not as a mere ancillary science but rather as an innovative approach supporting many other scientific disciplines. The journal focuses on original high-quality research in the realm of computational science in parallel and distributed environments, encompassing the facilitating theoretical foundations and the applications of large-scale computations and massive data processing. It addresses researchers and practitioners in areas ranging from aerospace to biochemistry, from electronics to geosciences, from mathematics to software architecture,

presenting verifiable computational methods, findings and solutions and enabling industrial users to apply techniques of leading-edge, large-scale, high performance computational methods. The fifth volume of the Transactions on Computational Science journal, edited by Yingxu Wang and Keith C.C. Chan, is devoted to the subject of cognitive knowledge representation. This field of study focuses on the internal knowledge representation mechanisms of the brain and how these can be applied to computer science and engineering. The issue includes the latest research results in internal knowledge representation at the logical, functional, physiological, and biological levels and describes their impacts on computing, artificial intelligence, and computational intelligence.

The fourth edition of the European Conference on Model-Driven Architecture – Foundations and Applications (ECMDA-FA 2008) was dedicated to furthering the state of knowledge and fostering the industrialization of the model-driven architecture (MDA) methodology. MDA is an initiative proposed by the Object Management Group (OMG) for platform-generic software development. It promotes the use of models in the specification, design, analysis, synthesis, deployment, and evolution of complex software systems. ECMDA-FA 2008 focused on engaging key European and international researchers and practitioners in a dialogue which will result in a stronger, more efficient industry, producing more reliable software on the basis of state-of-the-art research results. ECMDA-FA is a forum for exchanging information, discussing the latest results and arguing about future developments of MDA. It is a pleasure to be able to introduce the proceedings of ECMDA-FA 2008. ECMDA-FA addresses various MDA areas including model management, executable models, concrete syntaxes, aspects and concerns, validation and testing, model-based systems engineering, model-driven

development and service-oriented architectures, and the application of model-driven development.

There are so many people

who deserve warm thanks and gratitude. The fruitful collaboration of the Organization, Steering and Program Committee member and the vibrant community led to a successful conference: ECMDA-FA2008 obtained excellent results in terms of submissions, program size, and attendance. The Program Committee accepted, with the help of additional reviewers, research papers and industry papers for ECMDA-FA 2008: We received 87 submissions. Of these, a total of 31 were accepted including 21 research papers and 10 industry papers. We thank them for the thorough and high-quality selection process.

This book provides a practical introduction to computationally solving discrete optimization problems using dynamic programming. From the examples presented, readers should more easily be able to formulate dynamic programming solutions to their own problems of interest. We also provide and describe the design, implementation, and use of a software tool that has been used to numerically solve all of the problems presented earlier in the book.

Dieses Buch liefert sowohl die theoretischen Grundlagen als auch das praktische Handwerkszeug zur Anwendung der Model-Driven-Architecture der OMG. Nach einer Definition und der Einordnung des Ansatzes in die heutige Welt des Software-Engineering sowie der Vorstellung der notwendigen Grundlagen entwickeln die Autoren anhand eines Fallbeispiels ein prototypisches MDA Framework und setzen so die zuvor erarbeiteten Erkenntnisse in die Praxis um. Besonderes Augenmerk wird dabei auf die Verwendung von Open-Source Technologien, insbesondere der Eclipse-Plattform und ihrer untergliederten Projekte gelegt. Eine Übersicht über verwandte Ansätze und sinnvolle Ergänzungen zum vorgestellten Vorgehen sowie eine

abschließende Bewertung und ein vorsichtiger Ausblick in die Zukunft runden die Darstellung ab. Das Buch richtet sich damit sowohl an Entscheider, die ein fundiertes Basiswissen zur MDA erlangen wollen, als auch an Praktiker, die Antworten auf konkrete Fragestellungen zur Umsetzung benötigen.

A comprehensive collection of problems, solutions, and practical examples for anyone programming in Java, "The Java Cookbook" presents hundreds of tried-and-true Java "recipes" covering all of the major APIs as well as some APIs that aren't as well documented in other Java books. The book provides quick solutions to particular problems that can be incorporated into other programs, but that aren't usually programs in and of themselves.

The four-volume set LNCS 2657, LNCS 2658, LNCS 2659, and LNCS 2660 constitutes the refereed proceedings of the Third International Conference on Computational Science, ICCS 2003, held concurrently in Melbourne, Australia and in St. Petersburg, Russia in June 2003. The four volumes present more than 460 reviewed contributed and invited papers and span the whole range of computational science, from foundational issues in computer science and algorithmic mathematics to advanced applications in virtually all application fields making use of computational techniques. These proceedings give a unique account of recent results in the field.

This textbook describes all phases of a compiler: lexical analysis, parsing, abstract syntax, semantic actions, intermediate representations, instruction

selection via tree matching, dataflow analysis, graph-coloring register allocation, and runtime systems. It includes good coverage of current techniques in code generation and register allocation, as well as the compilation of functional and object-oriented languages, that is missing from most books. The most accepted and successful techniques are described concisely, rather than as an exhaustive catalog of every possible variant, and illustrated with actual Java classes. This second edition has been extensively rewritten to include more discussion of Java and object-oriented programming concepts, such as visitor patterns. A unique feature is the newly redesigned compiler project in Java, for a subset of Java itself. The project includes both front-end and back-end phases, so that students can build a complete working compiler in one semester. Ant is the premiere build management tool for use in Java environments. Unlike traditional build management tools such as GNU Make, Ant is itself written in Java, is platform independent, and interfaces well with the utilities in Sun's Java software development kit (SDK). In addition to being platform independent, Ant is also independent of the integrated development environment (IDE) being used. IDE independence is important for open source projects (or other projects) in which the various developers might use different IDEs. Using Ant, Java developers can: define build chunks, the

results that they must produce, and the dependencies between them; automatically retrieve source code from source control systems such as PVCs; build applications by having Ant compile the necessary source files in the proper order. Ant build files are written using XML---a well-established standard---so programmers using Ant are not required to learn yet another scripting language. They will likely already know XML, and will be able to leverage that knowledge. Ant is an open source project, and part of the Jakarta project. Jakarta is Sun's open source reference implementation for the JSP and Servlets specifications, and is part of the Apache group's work.

As knowledge-based software engineering matures and increasingly automates the software engineering life cycle, software engineering resources are shifting towards knowledge acquisition and the automated reuse of expert knowledge for developing software artifacts. This book summarizes the work and new research results presented at the Tenth Joint Conference on Knowledge-based Software Engineering (JCKBSE 2012), held on the island of Rhodes, Greece, in August 2012. The biennial Joint Conference on Knowledge-Based Software Engineering brings together researchers and practitioners to share ideas on the foundations, techniques, tools, and applications of knowledge-based software engineering theory and practice.

Topics addressed include theoretical foundations, practical techniques, software tools, applications and/or experience reports in knowledge-based software engineering. This book is published in the subseries Knowledge-Based Intelligent Engineering Systems (KBIES).

Many of the early issues in the field of tele-learning are now not only recognised but are being addressed, through professional and staff development routes, through innovative technological solutions, and through approaches and concepts that are better suited to particular educational contexts. *TeLE-LEARNING: The Challenge for the Third Millennium* provides details of the most recent advances in this area.

Service science constitutes an interdisciplinary approach to systematic innovation in service systems, integrating managerial, social, legal, and engineering aspects to address the theoretical and practical challenges of the services industry and its economy. This book contains the refereed proceedings of the 4th International Conference on Exploring Services Science (IESS), held in Porto, Portugal, in February 2013. This year, the conference theme was *Enhancing Service System Fundamentals and Experiences*, chosen to address the current need to explore enhanced methods, approaches, and techniques for a more sustainable and comprehensive economy and society. The 19

full and 9 short papers accepted for IESS were selected from 78 submissions and presented ideas and results related to innovation, services discovery, services engineering, and services management, as well as the application of services in information technology, business, healthcare, and transportation. This book constitutes the thoroughly refereed proceedings of the 10th International Joint Conference on Software Technologies, ICSOFT 2015, held in Colmar, France, in July 2015. The 23 revised full papers presented were carefully reviewed and selected from 117 submissions. The papers are organized around the following conference tracks: enterprise software technologies; software project management; software engineering methods and techniques; distributed and mobile software systems.

This is the 12th volume in a series on information modelling and knowledge bases. The topics of the articles cover a wide variety of themes in the domain of information modelling, design and specification of information systems and knowledge bases, ranging from foundations and theories to systems construction and application studies. The contributions in this volume represent the following major themes: models in intelligent activity; concept modelling and conceptual modelling; conceptual modelling and information requirements specification; collections of concepts, knowledge base design, and database design; human-computer interaction and modelling; software

engineering and modelling; and applications.

Long-awaited revision to a unique guide that covers both compilers and interpreters Revised, updated, and now focusing on Java instead of C++, this long-awaited, latest edition of this popular book teaches programmers and software engineering students how to write compilers and interpreters using Java. You'll write compilers and interpreters as case studies, generating general assembly code for a Java Virtual Machine that takes advantage of the Java Collections Framework to shorten and simplify the code. In addition, coverage includes Java Collections Framework, UML modeling, object-oriented programming with design patterns, working with XML intermediate code, and more.

Containing 101 fun, interesting, and useful ways to get more out of Java, this title targets developers and system architects who have some basic Java knowledge but may not be familiar with the wide range of libraries available.

This tutorial volume includes revised and extended lecture notes of six long tutorials, five short tutorials, and one peer-reviewed participant contribution held at the 4th International Summer School on Generative and Transformational Techniques in Software Engineering, GTTSE 2011. The school presents the state of the art in software language engineering and generative and transformational techniques in software engineering with coverage of foundations, methods, tools, and case studies.

This book constitutes the thoroughly refereed post-conference proceedings of the Third International

Symposium on Applications of Graph Transformations, AGTIVE 2007, held in Kassel, Germany, in October 2007. The 30 revised full papers presented together with 2 invited papers were carefully selected from numerous submissions during two rounds of reviewing and improvement. The papers are organized in topical sections on graph transformation applications, meta-modeling and domain-specific language, new graph transformation approaches, program transformation applications, dynamic system modeling, model driven software development applications, queries, views, and model transformations, as well as new pattern matching and rewriting concepts. The volume moreover contains 4 papers resulting from the adjacent graph transformation tool contest and concludes with 9 papers summarizing the state of the art of today's available graph transformation environments.

This book constitutes the refereed proceedings of the 10th International Conference on Algebraic Methodology and Software Technology, AMAST 2004, held in Stirling, Scotland, UK in July 2004. The 35 revised full papers presented together with abstracts of 5 invited talks and an invited paper were carefully reviewed and selected from 63 submissions. Among the topics covered are all current issues in formal methods related to algebraic approaches to software engineering including abstract data types, process algebras, algebraic specification, model checking, abstraction, refinement, model checking, state machines, rewriting, Kleene algebra, programming logic, etc.

Digital Communities in a Networked Society: e-

Commerce, e-Business and e-Government deals with the accelerating evolution in the computerization of society. This evolution, or should we call it a revolution, is dominantly driven by the Internet, and documented by the novelties introduced, year by year, by Information and Communication Technologies. The book contains recent results of research and development in the areas of: -E-government, -Business models of e-applications, -Innovative structures in the internet, -Auctions and e-payment, -Future aspects of communication, -Internet and the web, -Advanced platforms and grid computing, -Cooperation and integration, -Modeling and construction of e-services.

This book contains both relevant real-world research, as well as reviews of different areas of interest in the software engineering literature, such as clone identification. The contents of the various sections will provide a better understanding of known problems and detailed treatment of advanced topics. Consequently, the book consolidates the work and findings from leading researchers in the software research community in key areas such as maintainability, architectural recovery, code analysis, software migration, and tool support.

Software Engineering: The Implementation Phase provides the conceptual foundation required for the design and development of complex distributed and real-time applications. It summarizes a large spectrum of concepts, enabling readability and ease of comprehension by concentrating on Java and presenting mainly the programmatic aspects and basic functioning. The idea behind this book is not to overwhelm the reader, but instead provide sufficient information for the creation of prototypes, covering

most aspects of an application. This volume guides you to determine for yourself which is the best global architecture for an application: thin or thick client; direct database access or EJBs; listeners or multi-threading; etc. It also presents a number of architectures, namely sets of components, threads and links, around which the functionalities of the application may be built and checked incrementally. One of the strengths of this book is the availability of simple examples, which have all been carefully checked and are available at the book's website.

This book constitutes the refereed proceedings of the 13th International SDL Forum, SDL 2007, held in Paris, France. The 17 revised full papers presented were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on model driven engineering, testing, language extensions, implementation and modeling experience and extensions—addressing all aspects of systems design and system design languages.

1. 1 Background In this work, we develop a framework for the design of multi-agent systems inspired by (human) organizational principles. Organizations are complex entities formed to overcome various limitations of individual agencies, such as cognitive, physical, temporal and institutional limitations. There is a parallel between the complexity of organizations and multi-agent systems. Therefore, we explore the use of concepts, methods and techniques from human organizational design as architectural principles for multi-agent systems. Three research lines are presented: organizational modeling and coordination, interoperability and agent models. Organizational modeling and coordination are concerned with how resources (i. e. agents) can be identified and related to each other. In order to have agents cooperate, several issues of interoperability have to be addressed. Agent models deal with the design of individual intelligent software

agents, taking into account typical features of agent intelligence. Every (human) activity raises two challenges: division of labor and coordination [Mintzberg, 1993]. Division of labor is the decomposition of work (or goals) into various distinct tasks. Coordination refers to managing relations between these tasks to carry out the work. The patterns of division of labor, responsibilities (people who do the work), clustering of responsibilities into units and coordination between units can be defined by organizational structures [Galbraith, 1973]. The design of an organization should cover how one or more actors are engaged in one or more tasks, where knowledge, capabilities and resources are distributed.

[Copyright: b3fccf0d02960b666532ddf89f538096](#)