Eclipse Modeling Framework Second Edition

bull; Shows how EMF unifies three important technologies: Java, XML, and UML bull; Provides a comprehensive overview of the EMF classes including a complete quick reference for all the classes and methods in the EMF 1.1 API bull; Includes examples of many common framework customizations and programming techniques

Agile Java™ Development With Spring, Hibernate and Eclipse is a book about robust technologies and effective methods which help bring simplicity back into the world of enterprise Java development. The three key technologies covered in this book, the Spring Framework, Hibernate and Eclipse, help reduce the complexity of enterprise Java development significantly. Furthermore, these technologies enable plain old Java objects (POJOs) to be deployed in light-weight containers versus heavy-handed remote objects that require heavy EJB containers. This book also extensively covers technologies such as Ant, JUnit, JSP tag libraries and touches upon other areas such as such logging, GUI based debugging, monitoring using JMX, job scheduling, emailing, and more. Also, Extreme Programming (XP), Agile Model Driven Development (AMDD) and refactoring are methods that can expedite the software development projects by reducing the amount of up front requirements and design; hence these methods are embedded throughout the book but with just enough details and examples to not sidetrack the focus of this book. In addition, this book contains well separated, subjective material (opinion sidebars), comic illustrations, tips and tricks, all of which provide real-world and practical perspectives on relevant topics. Last but not least, this book demonstrates the complete lifecycle by building and following a sample application, chapter-by-chapter, starting from conceptualization to production using the technology and processes covered in this book. In summary, by using the technologies and methods covered in this book, the reader will be able to effectively develop enterprise-class Java applications, in an agile manner!

Describes the philosophy of the Daily 5 teaching structure and includes a collection of literacy tasks for students to complete daily.

The Definitive Guide to Eclipse Rich Client Development In Eclipse Rich Client Platform, Second Edition, three Eclipse Rich Client Platform (RCP) project leaders show how to use Eclipse 3.5 (“Galileo”) to rapidly deliver cross-platform applications with rich, native-feel GUIs. The authors fully reveal the power of Eclipse as a desktop application development platform; introduce important new improvements in Eclipse 3.5; and walk through developing a full-featured, branded RCP application for Windows, Linux, Mac, and other platforms—including handheld devices and kiosks. Drawing on their extensive experience, the authors cover building, refining, and refactoring prototypes; customizing user interfaces; adding help and software management features; and building, branding, testing, and shipping finished software. They demonstrate current best practices for developing modular and dynamically extensible systems, using third-party code libraries, packaging applications for diverse environments, and much more. For Java programmers at all levels of experience, this book Introduces important new RCP features such as p2, Commands, and Databinding Thoroughly covers key RCP-related technologies such as Equinox, SWT, JFace, and OSGi Shows how to effectively brand and customize RCP application look-and-feel Walks through user interface testing for RCP applications with SWTBot Illuminates key similarities and differences between RCP and conventional plug-in development Hands-on, pragmatic, and comprehensive, this book offers all the real-world, nontrivial code examples working developers need—as well as “deep dives” into key technical areas that are essential to your success.

Written by two world class programmers and software designers, this guide explains how to extend Eclipse for software projects and how to use Eclipse to create software tools that improve development time.

The updated second edition of the popular tutorial guide to automated testing. Selenium is one of the most popular open-source automated testing tools available today. Understanding Selenium-RC and writing automated tests in a programming language are sought after skills on the job market and a great way of maximising the benefit from automated testing. Contrary to the beliefs of many testers, learning to code does not have to be complicated or hard. "Selenium Simplified" takes you through the process of installing and learning to use all the basic tools needed to write automated tests using Java as the programming language. Written in a tutorial style, this book helps you learn to code even if you haven't programmed before. No time is wasted on the theory of automation or padding about the tools. This book focuses on the practical knowledge needed to automate tests for production systems.

This is a tutorial on Spring MVC, a module in the Spring Framework for rapidly developing web applications. The MVC in Spring MVC stands for Model-View-Controller, a design pattern widely used in Graphical User Interface (GUI) development. This pattern is not only common in web development, but is also used in desktop technology like Java Swing. Sometimes called Spring Web MVC, Spring MVC is one of the most popular web frameworks today and a most sought-after skill. This book is for anyone wishing to learn to develop Java-based web applications with Spring MVC. Sample applications come as Spring Tool Suite and Eclipse projects.

Customizing Alfresco with actions, web scripts, web forms, workflows, and more

As one of today's cloud computing services, Google App Engine does more than provide access to a large system of servers. It also offers you a simple model for building applications that scale automatically to accommodate millions of users. With Programming Google App Engine, you'll get expert practical guidance that will help you make the best use of this powerful platform. Google engineer Dan Sanderson shows you how to design your applications for scalability, including ways to perform common development tasks using App Engine's APIs and scalable services. You'll learn about App Engine's application server architecture, runtime environments, and scalable datastore for distributing data, as well as techniques for optimizing your application. App Engine offers nearly unlimited computing power, and this book provides clear and concise instructions for getting the most from it right from the source. Discover the differences between traditional web development and development with App Engine Learn the details of App Engine's Python and

Java runtime environments Understand how App Engine handles web requests and executes application code Learn how to use App Engine's scalable datastore, including queries and indexes, transactions, and data modeling Use task queues to parallelize and distribute work across the infrastructure Deploy and manage applications with ease
Databases and information systems are the backbone of modern information technology and are crucial to the IT systems which support all aspects of our everyday life; from government, education and healthcare, to business processes and the storage of our personal photos and archives. This book presents 22 of the best revised papers accepted following stringent peer review for the 11th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2014), held in Tallinn, Estonia, in June 2014. The conference provided a forum for the exchange of scientific achievements between the research communities of the Baltic countries and the rest of the world in the area of databases and information systems, bringing together researchers, practitioners and Ph.D. students from many countries. The subject areas covered at the conference focused on big data processing, data warehouses, data integration and services, data and knowledge management, e-government, as well as e-services and e-learning.
This book gives a detailed introduction into the Eclipse platform and covers all relevant aspects of Eclipse RCP development. Every topic in this book has a content section in which the topic is explained and afterwards you have several exercises to practice your learning. You will be guided through all relevant aspects of Eclipse 4 development using an comprehensive example which you continue to extend in the exercises. You will learn about the new programming concepts of Eclipse 4, e.g. the application model, dependency injection, CSS styling, the renderer framework, the event system and much more. Proven Eclipse technologies like SWT, JFace viewers, OSGi modularity and services, data binding, etc. are also covered in detail. This book requires a working knowledge of Java and assumes that you are familiar in using the Eclipse IDE for standard Java development. It assumes no previous experience of Eclipse plug-in and Eclipse RCP development.
A hands-on tutorial for new Eclipse GEF developers, plus a complete API reference and troubleshooting guide for all GEF developers * *A step-by-step walkthrough of all major GEF sub-products, driven by a realistic, running example. *How to design, develop and maintain commercial-quality GEF projects, avoid common pitfalls, and take full advantage of GEF's features. *Includes coverage of implementing GEF Usability and GEF Accessibility. *Detailed, example-rich coverage of testing GEF applications. As the popularity of Eclipse and SWT-based applications continues to grow, product specifications are requiring richer graphical interfaces. When standard widgets such as text editors, combo boxes and trees aren't enough, graphics may be the best choice to display information. In this practical, hands-on guide, three leading Eclipse graphics experts covers everything developers need to build the rich, visual interfaces they want. The authors introduce all three graphics frameworks available to Eclipse developers, Draw2D, Zest, and GEF, discussing the pros and cons of each. They carefully introduce each framework's API, walk through building a robust application with Draw2D, and then refactor their example application twice: first with Zest, and then with GEF. Coverage includes: Draw2D figures, layouts, connections, routing algorithms, and text support; Zest graphing and layout algorithms; GEF controllers, commands, requests, palette and tools, accessibility, usability; and much more. The book also includes a full chapter of advanced techniques, as well as practical troubleshooting guidance.
Java is now well-established as one of the world’s major programming languages, used in everything from desktop applications to web-hosted applications, enterprise systems and mobile devices. Java applications cover cloud-based services, the Internet of Things, self-driving cars, animation, game development, big data analysis and many more domains. The second edition of Foundational Java: Key Elements and Practical Programming presents a detailed guide to the core features of Java – and some more recent innovations – enabling the reader to build their skills and confidence though tried-and-trusted stages, supported by exercises that reinforce the key learning points. All the most useful and commonly applied Java syntax and libraries are introduced, along with many example programs that can provide the basis for more substantial applications. Use of the Eclipse Integrated Development Environment (IDE) and the JUnit testing framework is integral to the book, ensuring maximum productivity and code quality when learning Java, although to ensure that skills are not confined to one environment the fundamentals of the Java compiler and run time are also explained. Additionally, coverage of the Ant tool will equip the reader with the skills to automatically build, test and deploy applications independent of an IDE. Topics and features: • Presents the most up-to-date information on Java, including Java 14 • Examines the key theme of unit testing, introducing the JUnit 5 testing framework to emphasize the importance of unit testing in modern software development • Describes the Eclipse IDE, the most popular open source Java IDE and explains how Java can be run from the command line • Includes coverage of the Ant build tool • Contains numerous code examples and exercises throughout • Provides downloadable source code, self-test questions, PowerPoint slides and other supplementary material at the website http://www.foundjava.com This hands-on, classroom-tested textbook/reference is ideal for undergraduate students on introductory and intermediate courses on programming with Java. Professional software developers will also find this an excellent self-study guide/refresher on the topic. Dr. David Parsons is National Postgraduate Director at The Mind Lab, Auckland, New Zealand. He has been teaching programming in both academia and industry since the 1980s and writing about it since the 1990s.
Presents instructions for creating Android applications for mobile devices using Java.
Traditionally, research on model-driven engineering (MDE) has mainly focused on the use of models at the design, implementation, and verification stages of development. This work has produced relatively mature techniques and tools that are currently being used in industry and academia. However, software models also have the potential to be used at runtime, to monitor and verify particular aspects of runtime behavior, and to implement self-* capabilities (e.g., adaptation technologies used in self-healing, self-managing, self-optimizing systems). A key benefit of using models at runtime is that they can provide a richer semantic base for runtime decision-making related to runtime system concerns associated

with autonomic and adaptive systems. This book is one of the outcomes of the Dagstuhl Seminar 11481 on models@run.time held in November/December 2011, discussing foundations, techniques, mechanisms, state of the art, research challenges, and applications for the use of runtime models. The book comprises four research roadmaps, written by the original participants of the Dagstuhl Seminar over the course of two years following the seminar, and seven research papers from experts in the area. The roadmap papers provide insights to key features of the use of runtime models and identify the following research challenges: the need for a reference architecture, uncertainty tackled by runtime models, mechanisms for leveraging runtime models for self-adaptive software, and the use of models at runtime to address assurance for self-adaptive systems.

EMF: Eclipse Modeling Framework Dave Steinberg Frank Budinsky Marcelo Paternostro Ed Merks Series Editors: Erich Gamma • Lee Nackman • John Wiegand The Authoritative Guide to EMF Modeling and Code Generation The Eclipse Modeling Framework enables developers to rapidly construct robust applications based on surprisingly simple models. Now, in this thoroughly revised Second Edition, the project's developers offer expert guidance, insight, and examples for solving real-world problems with EMF, accelerating development processes, and improving software quality. This edition contains more than 40% new material, plus updates throughout to make it even more useful and practical. The authors illuminate the key concepts and techniques of EMF modeling, analyze EMF's most important framework classes and generator patterns, guide you through choosing optimal designs, and introduce powerful framework customizations and programming techniques. Coverage includes • Defining models with Java, UML, XML Schema, and Ecore • NEW: Using extended Ecore modeling to fully unify XML with UML and Java • Generating high-quality code to implement models and editors • Understanding and customizing generated code • Complete documentation of @model Javadoc tags, generator model properties, and resource save and load options • NEW: Leveraging the latest EMF features, including extended metadata, feature maps, EStore, cross-reference adapters, copiers, and content types • NEW: Chapters on change recording, validation, and utilizing EMF in stand-alone and Eclipse RCP applications • NEW: Modeling generics with Ecore and generating Java 5 code About the Authors Dave Steinberg is a software developer in IBM Software Group. He has worked with Eclipse and modeling technologies since joining the company, and has been a committer on the EMF project since its debut in 2002. Frank Budinsky, a senior architect in IBM Software Group, is an original coinventor of EMF and a founding member of the EMF project at Eclipse. He is currently cochair of the Service Data Objects (SDO) specification technical committee at OASIS and lead SDO architect for IBM. Marcelo Paternostro is a software architect and engineer in IBM Software Group. He is an EMF committer and has been an active contributor to several other Eclipse projects. Before joining IBM, Marcelo managed, designed, and implemented numerous projects using Rational's tools and processes. Ed Merks is the project lead of EMF and a colead of the top-level Modeling project at Eclipse. He holds a Ph.D. in Computing Science and has many years of in-depth experience in the design and implementation of languages, frameworks, and application development environments. Ed works as a software consultant in partnership with itemis AG.

BrunoBuchberger This book is a synopsis of basic and applied research done at the various re search institutions of the Softwarepark Hagenberg in Austria. Starting with 15 coworkers in my Research Institute for Symbolic Computation (RISC), I initiated the Softwarepark Hagenberg in 1987 on request of the Upper Aus trian Government with the objective of creating a scienti?c, technological, and economic impulse for the region and the international community. In the meantime, in a joint e?ort, the Softwarepark Hagenberg has grown to the current (2009) size of over 1000 R&D employees and 1300 students in six research institutions, 40 companies and 20 academic study programs on the bachelor, master's and PhD level. The goal of the Softwarepark Hagenberg is innovation of economy in one of the most important current technologies: software. It is the message of this book that this can only be achieved and guaranteed long term by "watering the root", namely emphasis on research, both basic and applied. In this book, we summarize what has been achieved in terms of research in the various research institutions in the Softwarepark Hagenberg and what research vision we have for the imminent future. When I founded the Softwarepark Hagenberg, in addition to the "watering the root" principle, I had the vision that such a technology park can only prosper if we realize the "magic triangle", i.e. the close interaction of research, academic education, and business applications at one site, see Figure 1.

This book discusses how model-based approaches can improve the daily practice of software professionals. This is known as Model-Driven Software Engineering (MDSE) or, simply, Model-Driven Engineering (MDE). MDSE practices have proved to increase efficiency and effectiveness in software development, as demonstrated by various quantitative and qualitative studies. MDSE adoption in the software industry is foreseen to grow exponentially in the near future, e.g., due to the convergence of software development and business analysis. The aim of this book is to provide you with an agile and flexible tool to introduce you to the MDSE world, thus allowing you to quickly understand its basic principles and techniques and to choose the right set of MDSE instruments for your needs so that you can start to benefit from MDSE right away. The book is organized into two main parts. The first part discusses the foundations of MDSE in terms of basic concepts (i.e., models and transformations), driving principles, application scenarios, and current standards, like the well-known MDA initiative proposed by OMG (Object Management Group) as well as the practices on how to integrate MDSE in existing development processes. The second part deals with the technical aspects of MDSE, spanning from the basics on when and how to build a domain-specific modeling language, to the description of Model-to-Text and Model-to-Model transformations, and the tools that support the management of MDSE projects. The second edition of the book features: a set of completely new topics, including: full example of the creation of a new modeling language (IFML), discussion of modeling issues and approaches in specific domains, like business process modeling, user interaction modeling, and enterprise architecture complete revision of examples, figures, and text, for improving readability, understandability, and coherence better formulation of definitions, dependencies between concepts and ideas addition of a complete index of book content In addition to the contents of the book, more resources are provided on the book's website http://www.mdse-book.com, including the examples presented in the book.

If you are a developer who is familiar with Ext JS and want to augment your skills to create even better web applications, this is the book for you. Basic knowledge of JavaScript/HTML/CSS and any server-side language (PHP, Java, C#, Ruby, or Python) is required.

This book is aimed at indie and existing game developers as well as those who want to get started with game development using LibGDX. Basic knowledge of Java programming and game development is required.

Discover WTP, the New End-to-End Toolset for Java-Based Web Development The Eclipse Web Tools Platform (WTP) seamlessly integrates all the tools today's Java Web developer needs. WTP is both an unprecedented Open Source resource for working developers

and a powerful foundation for state-of-the-art commercial products. Eclipse Web Tools Platform offers in-depth descriptions of every tool included in WTP, introducing powerful capabilities never before available in Eclipse. The authors cover the entire Web development process–from defining Web application architectures and development processes through testing and beyond. And if you're seeking to extend WTP, this book provides an introduction to the platform's rich APIs. The book also Presents step-by-step coverage of developing persistence, business logic, and presentation tiers with WTP and Java Introduces best practices for multiple styles of Web and Java EE development Demonstrates JDBC database access and configuration Shows how to configure application servers for use with WTP Walks through creating Web service application interfaces Covers automated testing with JUnit and Cactus, and automated builds utilizing Ant, Maven, and CruiseControl Introduces testing and profiling Web applications with the Eclipse Test and Performance Tools Platform (TPTP) project Describes how to extend WTP with new servers, file types, and WSDL extensions Foreword Preface Acknowledgments About the Authors Part I: Getting Started Chapter 1: Introduction Chapter 2: About the Eclipse Web Tools Platform Project Chapter 3: Quick Tour Chapter 4: Setting Up Your Workspace Part II: Java Web Application Development Chapter 5: Web Application Architecture and Design Chapter 6: Organizing Your Development Project Chapter 7: The Presentation Tier Chapter 8: The Business Logic Tier Chapter 9: The Persistence Tier Chapter 10: Web Services Chapter 11: Testing Part III: Extending WTP Chapter 12: Adding New Servers Chapter 13: Supporting New File Types Chapter 14: Creating WSDL Extensions Chapter 15: Customizing Resource Resolution Part IV: Products and Plans Chapter 16: Other Web Tools Based on Eclipse Chapter 17: The Road Ahead Glossary References Index This book is an invaluable resource for every Eclipse and enterprise Java Web developer: both those who use Eclipse to build other Web applications, and those who build Eclipse technologies into their own products. Complete source code examples are available at www.eclipsewtp.org.

Automatic layout is an important tool for the efficient use of graphical models in a model-driven engineering (MDE) context. Since the 1980s, research on graph layout methods has led to a multitude of different approaches, and several free software libraries for graph layout are available. However, today's practically relevant MDE tools hardly reflect this diversity. This thesis aims to support the use of automatic graph layout in such tools. A special focus is on the requirements of data flow models, where constraints on the positioning of ports and the routing of hyperedges pose additional challenges. These constraints are approached with extensions of the layer-based graph layout method. Furthermore, we discuss an infrastructure for managing collections of layout algorithms, allowing to flexibly specify layout configurations. These concepts are implemented in an open-source project based on Eclipse, an extensible platform that is well-known as a Java IDE and also hosts a large number of MDE tools. The presented contributions allow to integrate high-quality automatic layout into these tools with low effort.

Get up to speed on Scala, the JVM language that offers all the benefits of a modern object model, functional programming, and an advanced type system. Packed with code examples, this comprehensive book shows you how to be productive with the language and ecosystem right away, and explains why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. This second edition covers recent language features, with new chapters on pattern matching, comprehensions, and advanced functional programming. You'll also learn about Scala's command-line tools, third-party tools, libraries, and language-aware plugins for editors and IDEs. This book is ideal for beginning and advanced Scala developers alike. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big-data apps, using Scala's functional combinators Use traits for mixin composition and pattern matching for data extraction Learn the sophisticated type system that combines FP and object-oriented programming concepts Explore Scala-specific concurrency tools, including Akka Understand how to develop rich domain-specific languages Learn good design techniques for building scalable and robust Scala applications

Computer-Aided Design of User Interfaces VI gathers the latest experience of experts, research teams and leading organisations involved in computer-aided design of user interactive applications. This area investigates how it is desirable and possible to support, to facilitate and to speed up the development life cycle of any interactive system: requirements engineering, early-stage design, detailed design, deelopment, deployment, evaluation, and maintenance. In particular, it stresses how the design activity could be better understood for different types of advanced interactive ubiquitous computing, and multi-device environments.

Summary GWT in Action, Second Edition is a completely revised edition of the best-selling GWT book. It covers the new features introduced in GWT 2.4 and 2.5, as well as the best development practices that have emerged in the GWT community. It begins with a rapid-fire introduction to GWT and Ajax to get you up to speed with GWT concepts and tools. Then, you'll explore key concepts like managing events, interacting with the server, creating UI components, building your user interface declaratively using UiBinder ... and more. About the Technology Google Web Toolkit works on a simple idea. Write your web application in Java, and GWT crosscompiles it into JavaScript. It is open source, supported by Google, and version 2.5 now includes a library of high-quality interface components and productivity tools that make using GWT a snap. The JavaScript it produces is really good. About this Book GWT in Action, Second Edition is a revised edition of the best-selling GWT book. In it, you'll explore key concepts like managing events, interacting with the server, and creating UI components. As you move through its engaging examples, you'll absorb the latest thinking in application design and industry-grade best practices, such as implementing MVP, using dependency injection, and code optimization. Written for Java developers, the book requires no prior knowledge of GWT. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. What's Inside Covers GWT 2.4 and up Efficient use of large data sets Optimizing with client bundles, deferred binding, and code splitting Using generators and dependency injection About the Authors Adam Tacy and Robert Hanson coauthored the first edition of GWT in Action. Jason Essington is a Java developer and an active contributor to the GWT mailing list and the GWT IRC channel. Anna Tökke is a programmer and solutions architect working with GWT on a daily basis. Table of Contents PART 1 BASICS GWT Building a GWT application: saying "Hello World!" Building a GWT application: enhancing Hello World PART 2 NEXT STEPS Creating your own widgets Using client bundles Interface design with UiBinder Communicating with GWT-RPC Using RequestFactory The Editor framework Data-presentation (cell) widgets Using JSNI—JavaScript Native Interface Classic Ajax and HTML forms Internationalization, localization, and accessibility PART 3 ADVANCED Advanced event handling and event busses Building MVP-based applications Dependency injection Deferred binding Generators Metrics and code splitting

The new edition of an introduction to multiagent systems that captures the state of the art in both theory and practice, suitable as textbook or reference. Multiagent systems are made up of multiple interacting intelligent agents—computational entities to some degree autonomous and able to cooperate, compete, communicate, act flexibly, and exercise control over their behavior within the frame of their objectives. They are the enabling technology for a wide range of advanced applications relying on distributed and parallel processing of data, information, and knowledge

relevant in domains ranging from industrial manufacturing to e-commerce to health care. This book offers a state-of-the-art introduction to multiagent systems, covering the field in both breadth and depth, and treating both theory and practice. It is suitable for classroom use or independent study. This second edition has been completely revised, capturing the tremendous developments in multiagent systems since the first edition appeared in 1999. Sixteen of the book's seventeen chapters were written for this edition; all chapters are by leaders in the field, with each author contributing to the broad base of knowledge and experience on which the book rests. The book covers basic concepts of computational agency from the perspective of both individual agents and agent organizations; communication among agents; coordination among agents; distributed cognition; development and engineering of multiagent systems; and background knowledge in logics and game theory. Each chapter includes references, many illustrations and examples, and exercises of varying degrees of difficulty. The chapters and the overall book are designed to be self-contained and understandable without additional material. Supplemental resources are available on the book's Web site. Contributors Rafael Bordini, Felix Brandt, Amit Chopra, Vincent Conitzer, Virginia Dignum, Jürgen Dix, Ed Durfee, Edith Elkind, Ulle Endriss, Alessandro Farinelli, Shaheen Fatima, Michael Fisher, Nicholas R. Jennings, Kevin Leyton-Brown, Evangelos Markakis, Lin Padgham, Julian Padget, Iyad Rahwan, Talal Rahwan, Alex Rogers, Jordi Sabater-Mir, Yoav Shoham, Munindar P. Singh, Kagan Tumer, Karl Tuyls, Wiebe van der Hoek, Laurent Vercouter, Meritxell Vinyals, Michael Winikoff, Michael Wooldridge, Shlomo Zilberstein

This volume contains the proceedings of the 22nd International Conference on Medical Informatics Europe 2009 (MIE) in Sarajevo, Bosnia and Herzegovina, September 2009. The scientific topics presented in these proceedings range from national and transnational ehealth roadmaps, health information and electronic health record systems, systems interoperability and communication standards, medical terminology and ontology approaches, and social networks to web, web 2.0, and semantic web solutions for patients, health personnel and researchers.

IBM® Content Navigator provides a unified user interface for your Enterprise Content Management (ECM) solutions. It also provides a robust development platform so you can build customized user interface and applications to deliver value and an intelligent, business-centric experience. This IBM Redbooks® publication guides you through the Content Navigator platform, its architecture, and the available programming interfaces. It describes how you can configure and customize the user interface with the administration tools provided, and how you can customize and extend Content Navigator using available development options with sample code. Specifically, the book shows how to set up a development environment, and develop plug-ins that add an action, service, and feature to the user interface. Customization topics include implementing request and response filters, external data services (EDS), creating custom step processors, and using Content Navigator widgets in other applications. This book also covers mobile development, viewer customization, component deployment, and debugging and troubleshooting. This book is intended for IT architects, application designers and developers working with IBM Content Navigator and IBM ECM products. It offers a high-level description of how to extend and customize IBM Content Navigator and also more technical details of how to do implementations with sample code.

This book provides a complete and comprehensive reference/guide to Pyomo (Python Optimization Modeling Objects) for both beginning and advanced modelers, including students at the undergraduate and graduate levels, academic researchers, and practitioners. The text illustrates the breadth of the modeling and analysis capabilities that are supported by the software and support of complex real-world applications. Pyomo is an open source software package for formulating and solving large-scale optimization and operations research problems. The text begins with a tutorial on simple linear and integer programming models. A detailed reference of Pyomo's modeling components is illustrated with extensive examples, including a discussion of how to load data from data sources like spreadsheets and databases. Chapters describing advanced modeling capabilities for nonlinear and stochastic optimization are also included. The Pyomo software provides familiar modeling features within Python, a powerful dynamic programming language that has a very clear, readable syntax and intuitive object orientation. Pyomo includes Python classes for defining sparse sets, parameters, and variables, which can be used to formulate algebraic expressions that define objectives and constraints. Moreover, Pyomo can be used from a command-line interface and within Python's interactive command environment, which makes it easy to create Pyomo models, apply a variety of optimizers, and examine solutions. The software supports a different modeling approach than commercial AML (Algebraic Modeling Languages) tools, and is designed for flexibility, extensibility, portability, and maintainability but also maintains the central ideas in modern AMLs.

A Practical Guide to SysML: The Systems Modeling Language is a comprehensive guide to SysML for systems and software engineers. It provides an advanced and practical resource for modeling systems with SysML. The source describes the modeling language and offers information about employing SysML in transitioning an organization or project to model-based systems engineering. The book also presents various examples to help readers understand the OMG Systems Modeling Professional (OCSMP) Certification Program. The text is organized into four parts. The first part provides an overview of systems engineering. It explains the model-based approach by comparing it with the document-based approach and providing the modeling principles. The overview of SYsML is also discussed. The second part of the book covers a comprehensive description of the language. It discusses the main concepts of model organization, parametrics, blocks, use cases, interactions, requirements, allocations, and profiles. The third part presents examples that illustrate how SysML supports different model-based procedures. The last part discusses how to transition and deploy SysML into an organization or project. It explains the integration of SysML into a systems development environment. Furthermore, it describes the category of data that are exchanged between a SysML tool and other types of tools, and the types of exchange mechanisms that can be used. It also covers the criteria that must be considered when selecting a SysML. Software and systems engineers, programmers, IT practitioners, experts, and non-experts will find

this book useful. *The authoritative guide for understanding and applying SysML *Authored by the foremost experts on the language *Language description, examples, and quick reference guide included

Focussing on the formulation of mathematical models for the light curves of eclipsing binary stars, and on the algorithms for generating such models, this book provides astronomers, both amateur and professional, with a guide for - specifying an astrophysical model for a set of observations - selecting an algorithm to determine the parameters of the model - estimating the errors of the parameters. It is written for readers with knowledge of basic calculus and linear algebra; appendices cover mathematical details on such matters as optimisation, co-ordinate systems, and specific models. While emphasising the physical and mathematical framework, the discussion remains close to the problems of actual implementation. The book concludes with chapters on specific models and approaches and the authors'views on the structure of future light-curve programs.

This book constitutes the thoroughly refereed post-proceedings of the Second International Conference on the Quality of Software Architectures, QoSA 2006, held in Västerås, Sweden in June 2006, co-located with the 9th International Symposium on Component-Based Software Engineering, CBSE 2006. Coverage includes architecture evaluation, managing and applying architectural knowledge, and processes for supporting architecture quality.

EMF : Eclipse Modeling FrameworkEMFEclipse Modeling FrameworkPearson Education

Users can dramatically improve the design, performance, and manageability of object-oriented code without altering its interfaces or behavior. "Refactoring" shows users exactly how to spot the best opportunities for refactoring and exactly how to do it, step by step.

Achieve Breakthrough Productivity and Quality with MDD and Eclipse-Based DSLs Domain-specific languages (DSLs) and model-driven development (MDD) offer software engineers powerful new ways to improve productivity, enhance quality, and insulate systems from rapid technological change. Now, there’s a pragmatic, start-to-finish guide to creating DSLs and using MDD techniques with the powerful open source Eclipse platform. In Eclipse Modeling Project, Richard C. Gronback illuminates both the principles and techniques software professionals need to master, offering insights that will be invaluable to developers working with any tool or platform. As coleader of the Eclipse Modeling Project, Gronback is singularly well-positioned to demonstrate DSLs and MDD at work in Eclipse. Gronback systematically introduces each of the Eclipse technologies that can be used in DSL and MDD development. Throughout, he introduces key concepts and technologies in the context of a complete worked example and presents new best practices and never-before published techniques. He also covers Eclipse projects discussed in no other book, including Query/View/Transformation (QVT) and the Graphical Modeling Framework (GMF)—a project the author personally leads. Eclipse Modeling Project gives software practitioners all the knowledge they need to explore the remarkable potential of DSLs and MDD—and includes coverage of Why a model-based approach enables the rapid customization of high-quality solutions within the product line paradigm How the Eclipse Modeling Project’s capabilities can be used to efficiently create new DSLs Powerful techniques for developing DSL abstract syntax, graphical notation, and textual syntax How to build Model-to-Model (M2M) and Model-to-Text (M2T) transformations—including a powerful new M2M implementation of the Object Management Group’s QVT Operational Mapping Language (OML) Efficiently packaging and deploying DSLs with Eclipse Complete reference sections for the Graphical Editing Framework (GEF), GMF runtime and tooling, QVT OML, Xpand, and more

Learn how to implement a DSL with Xtext and Xtend using easy-to-understand examples and best practices About This Book Leverage the latest features of Xtext and Xtend to develop a domain-specific language. Integrate Xtext with popular third party IDEs and get the best out of both worlds. Discover how to test a DSL implementation and how to customize runtime and IDE aspects of the DSL Who This Book Is For This book is targeted at programmers and developers who want to create a domain-specific language with Xtext. They should have a basic familiarity with Eclipse and its functionality. Previous experience with compiler implementation can be helpful but is not necessary since this book will explain all the development stages of a DSL. What You Will Learn Write Xtext grammar for a DSL; Use Xtend as an alternative to Java to write cleaner, easier-to-read, and more maintainable code; Build your Xtext DSLs easily with Maven/Tycho and Gradle; Write a code generator and an interpreter for a DSL; Explore the Xtext scoping mechanism for symbol resolution; Test most aspects of the DSL implementation with JUnit; Understand best practices in DSL implementations with Xtext and Xtend; Develop your Xtext DSLs using Continuous Integration mechanisms; Use an Xtext editor in a web application In Detail Xtext is an open source Eclipse framework for implementing domain-specific languages together with IDE functionalities. It lets you implement languages really quickly; most of all, it covers all aspects of a complete language infrastructure, including the parser, code generator, interpreter, and more. This book will enable you to implement Domain Specific Languages (DSL) efficiently, together with their IDE tooling, with Xtext and Xtend. Opening with brief coverage of Xtext features involved in DSL implementation, including integration in an IDE, the book will then introduce you to Xtend as this language will be used in all the examples throughout the book. You will then explore the typical programming development workflow with Xtext when we modify the grammar of the DSL. Further, the Xtend programming language (a fully-featured Java-like language tightly integrated with Java) will be introduced. We then explain the main concepts of Xtext, such as validation, code generation, and customizations of runtime and UI aspects. You will have learned how to test a DSL implemented in Xtext with JUnit and will progress to advanced concepts such as type checking and scoping. You will then integrate the typical Continuous Integration systems built in to Xtext DSLs and familiarize yourself with Xbase. By the end of the book, you will manually maintain the EMF model for an Xtext DSL and will see how an Xtext DSL can also be used in IntelliJ. Style and approach A step-by step-tutorial with illustrative examples that will let you master using Xtext and implementing DSLs with its custom language, Xtend.

See how Domain-Driven Design (DDD) combines with Jakarta EE MicroProfile or Spring Boot to offer a complete suite for building enterprise-grade applications. In this book you will see how these all come together in one of the most efficient ways to develop complex software. Practical Domain-Driven Design in Enterprise Java starts by building out the Cargo Tracker reference application as a monolithic application using the Jakarta EE platform. By doing so, you will map concepts of DDD (bounded contexts, language, and aggregates) to the corresponding available tools (CDI, JAX-RS, and JPA) within the Jakarta EE platform. Once you have completed the monolithic application, you will walk through the complete conversion of the monolith to a microservices-based architecture, again mapping the concepts of DDD and the corresponding available tools within the

MicroProfile platform (config, discovery, and fault tolerance). To finish this section, you will examine the same microservices architecture on the Spring Boot platform. The final set of chapters looks at what the application would be like if you used the CQRS and event sourcing patterns. Here you'll use the Axon framework as the base framework. What You Will Learn Discover the DDD architectural principles and use the DDD design patterns Use the new Eclipse Jakarta EE platform Work with the Spring Boot framework Implement microservices design patterns, including context mapping, logic design, entities, integration, testing, and security Carry out event sourcing Apply CQRS Who This Book Is For Junior developers intending to start working on enterprise Java; senior developers transitioning from monolithic- to microservices-based architectures; and architects transitioning to a DDD philosophy of building applications.

The importance of databases and information systems to the functioning of 21st century life is indisputable. This book presents papers from the 13th International Baltic Conference on Databases and Information Systems, held in Trakai, Lithuania, from 1- 4 July 2018. Since the first of these events in 1994, the Baltic DB&IS has proved itself to be an excellent forum for researchers, practitioners and PhD students to deliver and share their research in the field of advanced information systems, databases and related areas. For the 2018 conference, 69 submissions were received from 15 countries. Each paper was assigned for review to at least three referees from different countries. Following review, 24 regular papers were accepted for presentation at the conference, and from these presented papers the 14 best-revised papers have been selected for publication in this volume, together with a preface and three invited papers written by leading experts. The selected revised and extended papers present original research results in a number of subject areas: information systems, requirements and ontology engineering; advanced database systems; internet of things; big data analysis; cognitive computing; and applications and case studies. These results will contribute to the further development of this fast-growing field, and will be of interest to all those working with advanced information systems, databases and related areas.

This book constitutes the refereed proceedings of the 13th International Baltic Conference on Databases and Information Systems, DB&IS 2018, held in Trakai, Lithuania, in July 2018. The 24 revised papers presented were carefully reviewed and selected from 69 submissions. The papers are centered around topics like information systems engineering, enterprise information systems, business process management, knowledge representation, ontology engineering, systems security, information systems applications, database systems, machine learning, big data analysis, big data processing, cognitive computing.

Want to build apps for Android devices? This book is the perfect way to master the fundamentals. Written by experts who have taught this mobile platform to hundreds of developers in large organizations and startups alike, this gentle introduction shows experienced object-oriented programmers how to use Android's basic building blocks to create user interfaces, store data, connect to the network, and more. Throughout the book, you'll build a Twitter-like application, adding new features with each chapter. You'll also create your own toolbox of code patterns to help you program any type of Android application with ease. Become familiar with the Android platform and how it fits into the mobile ecosystem Dive into the Android stack, including its application framework and the APK application package Learn Android's building blocks: Activities, Intents, Services, Content Providers, and Broadcast Receivers Create basic Android user interfaces and organize UI elements in Views and Layouts Build a service that uses a background process to update data in your application