Instant Mock Testing With Powermock By Deep Shah

Mockito Essentials
Children of Dune
Building Cloud Native Applications with Go and Java for Docker and Kubernetes
A Study Guide
Real World Java Ee Night Hacks Dissecting the Business Tier
Testing with JUnit
Spring Microservices
I, Robot
Mockito for Spring
A Study Guide Using Spring Framework 5
The Official Documentation for DocBook
DocBook 5: The Definitive Guide
Real World Java Ee Patterns-Rethinking Best Practices
Java Testing with Spock
Simple, Rapid, Effective, and Scalable
A hands-on guide to creating clean web applications with code examples in Java
Mockito Cookbook
Mastering Unit Testing Using Mockito and JUnit
Build modern, cloud-native, and distributed systems using Spring Boot
Real-time apps and microservices with the Kafka Streams API
The Art of Unit Testing
JavaScript Testing with Jasmine
Elements of Reusable Object-Oriented Software
Software Systems Architecture
Build Cloud-Native Enterprise Java Applications and Microservices
Programming for the Java Virtual Machine
Instant Mock Testing with PowerMock
Kafka Streams in Action
Design Patterns
Mastering Spring Boot 2.0
Beginning Quarkus Framework
Mastering C++ Programming
Get Your Hands Dirty on Clean Architecture
The JHipster Mini-Book
JUnit in Action
with examples in C#
JIRA Development Cookbook
Mastering Software Testing with JUnit 5

SWANSON TOMMY

Mockito Essentials Lulu.com
Filled with practical, step-by-step instructions and clear explanations for the most important and useful tasks. A concise guide full of step-by-step recipes, to teach you how you can use PowerMock to unit test code.If you are a beginner who would like to learn how to use PowerMock, this is the perfect guide for you! As the book comprehensively covers both basic and advanced concepts, this guide is also suitable for those of you who have an intermediate knowledge of PowerMock.
Children of Dune Packt Publishing Ltd
Software Systems Architecture is a practitioner-oriented guide to designing and implementing effective architectures for information systems. It is both a readily accessible introduction to software architecture and an invaluable handbook of well-established best practices. It shows why the role of the architect is central to any successful information-systems development project, and, by presenting a set of architectural viewpoints and perspectives, provides specific direction for improving your own and your organization's approach to software systems architecture. With this book you will learn how to Design an architecture that reflects and balances the different needs of its stakeholders Communicate the architecture to stakeholders and demonstrate that it has met their requirements Focus on architecturally significant aspects of design, including frequently overlooked areas such as performance, resilience, and location Use scenarios and patterns to drive the creation and validation of your architecture Document your architecture as a set of related views Use perspectives to ensure that your architecture exhibits important qualities such as performance, scalability, and security The architectural viewpoints and perspectives presented in the book also provide a valuable long-term reference source for new and experienced architects alike. Whether you are an aspiring or practicing software architect, you will find yourself referring repeatedly to the practical advice in this book throughout the lifecycle of your projects. A supporting Web site containing further information can be found at www.viewpoints-and-perspectives.info
Building Cloud Native Applications with Go and Java for Docker and Kubernetes Springer Nature
The surprisingly successful book Real World Java EE Patterns-Rethinking Best Practices [press.adam-bien.com] discusses the rethinking of legacy J2EE patterns. Now, Real World Java EE Night Hacks walks you through the Java EE 6 best practices and patterns used to create a real world application called "x-ray." X-ray is a high-performance blog statistics application built with nothing but vanilla Java EE 6 leveraging the synergies between the JAX-RS, EJB 3.1, JPA 2, and CDI 1.0 APIs. Foreword by James Gosling, Father of Java
A Study Guide Addison-Wesley Professional
Build scalable microservices with Spring, Docker, and Mesos About This Book Learn how to efficiently build and implement microservices in Spring, and how to use Docker and Mesos to push the boundaries of what you thought possible Examine a number of real-world use cases and hands-on code examples. Distribute your microservices in a completely new way Who This Book Is For If you are a Spring developers and want to build cloud-ready, internet-scale applications to meet modern business demands, then this book is for you Developers will understand how to build simple Restful services and organically grow them to truly enterprise grade microservices ecosystems. What You Will Learn Get to know the microservices development lifecycle process See how to implement microservices governance Familiarize yourself with the microservices architecture and its benefits Use Spring Boot to develop microservices Find out how to avoid common pitfalls when developing microservices Be introduced to end-to-end microservices written in Spring Framework and Spring Boot In Detail The Spring Framework is an application framework and inversion of the control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions to build web applications on top of the Java EE platform. This book will help you implement the microservice architecture in Spring Framework, Spring Boot, and Spring Cloud. Written to the latest specifications of Spring, you'll be able to build modern, Internet-scale Java applications in no time. We would start off with the guidelines to implement responsive microservices at scale. We will then deep dive into Spring Boot, Spring Cloud, Docker, Mesos, and Marathon. Next you will understand how Spring Boot is used to deploy autonomous services, server-less by removing the need to have a heavy-weight application server. Later you will learn how to go further by deploying your microservices to Docker and manage it with Mesos. By the end of the book, you'll will gain more clarity on how to implement microservices using Spring Framework and use them in Internet-scale deployments through real-world examples. Style and approach The book follows a step by step approach on how to develop microservices using Spring Framework, Spring Boot, and a set of Spring Cloud components that will help you scale your applications.
Real World Java Ee Night Hacks Dissecting the Business Tier Packt Publishing Ltd
This book is ideal for developers who have some experience in Java application development as well as some basic knowledge of test doubles and JUnit testing. This book also introduces you to the fundamentals of JUnit testing, test doubles, refactoring legacy code, and writing JUnit tests for GWT and web services.
Testing with JUnit Simon and Schuster
Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++,

C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

Spring Microservices Packt Publishing Ltd

Real World Java EE Patterns - Rethinking Best Practices (http://realworldpatterns.com) discusses patterns and best practices in a structured way, with code from real world projects. The rewritten and re-edited version of this book covers: an introduction into the core principles and APIs of Java EE 6, principles of transactions, isolation levels, CAP and BASE, remoting, pragmatic modularization and structure of Java EE applications, discussion of superfluous patterns and outdated best practices, patterns for domain driven and service oriented components, custom scopes, asynchronous processing and parallelization, real time HTTP events, schedulers, REST optimizations, plugins and monitoring tools, and fully functional JCA 1.6 implementation. Real World Java EE Patterns--Rethinking Best Practices will not only help experienced developers and architects to write concise code, but especially help you to shrink the codebase to unbelievably small sizes: -).

I, Robot Prentice Hall Professional

Learn to develop, test, and deploy your Spring Boot distributed application and explore various best practices. Key Features Build and deploy your microservices architecture in the cloud Build event-driven resilient systems using Hystrix and Turbine Explore API management tools such as KONG and API documentation tools such as Swagger Book Description Spring is one of the best frameworks on the market for developing web, enterprise, and cloud ready software. Spring Boot simplifies the building of complex software dramatically by reducing the amount of boilerplate code, and by providing production-ready features and a simple deployment model. This book will address the challenges related to power that come with Spring Boot's great configurability and flexibility. You will understand how Spring Boot configuration works under the hood, how to overwrite default configurations, and how to use advanced techniques to prepare Spring Boot applications to work in production. This book will also introduce readers to a relatively new topic in the Spring ecosystem – cloud native patterns, reactive programming, and applications. Get up to speed with microservices with Spring Boot and Spring Cloud. Each chapter aims to solve a specific problem or teach you a useful skillset. By the end of this book, you will be proficient in building and deploying your Spring Boot application. What you will learn Build logically structured and highly maintainable Spring Boot applications Configure RESTful microservices using Spring Boot Make the application production and operation-friendly with Spring Actuator Build modern, high-performance distributed applications using cloud patterns Manage and deploy your Spring Boot application to the cloud (AWS) Monitor distributed applications using log aggregation and ELK Who this book is for The book is targeted at experienced Spring and Java developers who have a basic knowledge of working with Spring Boot. The reader should be familiar with Spring Boot basics, and aware of its benefits over traditional Spring Framework-based applications.

Mockito for Spring "O'Reilly Media, Inc."

Harness the power of Quarkus, the supersonic subatomic cloud-native Java platform from Red Hat. This book covers everything you need to know to get started with the platform, which has been engineered from the ground up for superior performance and cloud-native deployment. You'll start with an overview of the Quarkus framework and its features. Next, you'll dive into building your first microservice using Quarkus, including the use of JAX-RS, Swagger, Microprofile, REST, reactive programming, and more. You'll see how to seamlessly add Quarkus to existing Spring framework projects. The book continues with a dive into the dependency injection pattern and how Quarkus supports it, working with annotations and facilities from both Jakarta EE CDI and the Spring framework. You'll also learn about dockerization and serverless technologies to deploy your microservice. Next you'll cover how data access works in Quarkus with Hibernate, JPA, Spring Boot, MongoDB, and more. This will also give you an eye for efficiency with reactive SQL, microservices, and many more reactive components. You'll also see tips and tricks not available in the official documentation for Quarkus. Lastly, you'll test and secure Quarkus-based code and use different deployment scenarios to package and deploy your Quarkus-based microservice for the cloud, using Amazon Web Services as a focus. After reading and using Beginning Quarkus Framework, you'll have the essentials to build and deploy cloud-native microservices and full-fledged applications. Author Tayo Koleoso goes to great lengths to ensure this book has up to date material including brand new and some unreleased features! What You Will Learn Build and deploy cloud-native Java applications with Quarkus Create Java-based microservices Integrate existing technologies such as the Spring framework and vanilla Java EE into the Quarkus framework Work with the Quarkus data layer on persistence with SQL, reactive SQL, and NoSQL Test code in Quarkus with the latest versions of JUnit and Testcontainers Secure your microservices with JWT and other technologies Package your microservices with Docker containers and GraalVM native image tooling Tips and techniques you won't find in the official Quarkus documentation Who This Book Is For Intermediate Java developers familiar with microservices, the cloud in general, and REST web services, but interested in modern approaches.

A Study Guide Using Spring Framework 5 Simon and Schuster

Your one-stop resource for mastering extensions and customizations in JIRA 7+ About This Book Explore the new features of JIRA 7 and best practices for agile development and integration with development tools Customize the look and feel of your JIRA UI to match your specific user needs Create seamless reports that make sense of your data through easy-to-use techniques Who This Book Is For If you are a JIRA developer or administrator, or a project manager who wants to fully exploit the exciting capabilities of JIRA, then this is the perfect book for you. What You Will Learn Create and deploy your own JIRA plugins Manipulate workflows to transform JIRA into a user-friendly system Create custom reports that show statistics for particular people, projects, versions, or other fields within issues Simplify reporting by writing your own JIRA gadgets, which can be added into a user's dashboard Master database handling by extending and retrieving custom field details from the database Deal with custom fields on an issue and program custom field option In Detail JIRA provides issue and project tracking for software development teams to improve code quality and the speed of development. With the new version of JIRA, you can create your own JIRA plugins and customize the look and feel of your JIRA UI easier than ever. JIRA Development Cookbook , Third Edition, is a one-stop resource to master extensions and customizations in JIRA. This book starts with recipes about simplifying the plugin

development process followed by recipes dedicated to the plugin framework. Then, you will move on to writing custom field plugins to create new field types or custom searchers. You will also learn how to program and customize workflows to transform JIRA into a user-friendly system. With so much data spanning different projects, issues, and so on, we will cover how to work on reports and gadgets to get customized data according to our needs. At the end of the book, you will learn how to customize JIRA by adding new tabs, menus, and web items; communicate with JIRA via the REST APIs; and work with the JIRA database. Style and approach The most unique aspect of this book is its approach. This book is recipe-based, with real-world examples, that will empower you to implement agile processes and explore the exciting capabilities of Jira.

The Official Documentation for DocBook Packt Publishing Ltd

The things you need to do to set up a new software project can be daunting. First, you have to select the back-end framework to create your API, choose your database, set up security, and choose your build tool. Then you have to choose the tools to create your front end: select a UI framework, configure a build tool, set up Sass processing, configure your browser to auto-refresh when you make changes, and configure the client and server so they work in unison. If you're building a new application using Spring Boot and Angular, you can save days by using JHipster. JHipster generates a complete and modern web app, unifying: - A high-performance and robust Java stack on the server side with Spring Boot - A sleek, modern, mobile-first front-end with Angular and Bootstrap - A robust microservice architecture with the JHipster Registry, Netflix OSS, the ELK stack, and Docker - A powerful workflow to build your application with Yeoman, Webpack, and Maven/Gradle

DocBook 5: The Definitive Guide Simon and Schuster

This book explains in detail how to implement unit tests using two very popular open source Java technologies: JUnit and Mockito. It presents a range of techniques necessary to write high quality unit tests - e.g. mocks, parametrized tests and matchers. It also discusses trade-offs related to the choices we have to make when dealing with some real-life code issues. The book stresses the importance of writing readable and maintainable unit tests, and puts a lot of stress on code quality. It shows how to achieve testable code and to eliminate common mistakes by following the Test Driven Development approach. Every topic discussed in the book is illustrated with code examples, and each chapter is accompanied by some exercises. By reading this book you will: Grasp the role and purpose of unit tests Write high-quality, readable and maintainable unit tests Learn how to use JUnit and Mockito (but also other useful tools) Avoid common pitfalls when writing unit tests Recognize bad unit tests, and fix them in no time Develop code following the Test Driven Development (TDD) approach Use mocks, stubs and test-spies intelligently Measure the quality of your tests using code coverage and mutation testing Learn how to improve your tests' code so it is an asset and not a burden Test collections, expected exceptions, time-dependent methods and much more Customize test reports so that they show you what you really need to know Master tools and techniques your team members have never even heard of (priceless!):) Nowadays every developer is expected to write unit tests. While simple in theory, in practice writing high-quality unit tests can turn out to be a real challenge. This book will help.

Real World Java Ee Patterns-Rethinking Best Practices Dune

Summary Effective Unit Testing is written to show how to write good tests—tests that are concise and to the point, expressive, useful, and maintainable. Inspired by Roy Osherove's bestselling The Art of Unit Testing, this book focuses on tools and practices specific to the Java world. It introduces you to emerging techniques like behavior-driven development and specification by example, and shows you how to add robust practices into your toolkit. About Testing Test the components before you assemble them into a full application, and you'll get better software. For Java developers, there's now a decade of experience with well-crafted tests that anticipate problems, identify known and unknown dependencies in the code, and allow you to test components both in isolation and in the context of a full application. About this Book Effective Unit Testing teaches Java developers how to write unit tests that are concise, expressive, useful, and maintainable. Offering crisp explanations and easy-to-absorb examples, it introduces emerging techniques like behavior-driven development and specification by example. Programmers who are already unit testing will learn the current state of the art. Those who are new to the game will learn practices that will serve them well for the rest of their career. 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. About the Author Lasse Koskela is a coach, trainer, consultant, and programmer. He hacks on open source projects, helps companies improve their productivity, and speaks frequently at conferences around the world. Lasse is the author of Test Driven, also published by Manning. What's Inside A thorough introduction to unit testing Choosing best-of-breed tools Writing tests using dynamic languages Efficient test automation Table of Contents PART 1 FOUNDATIONS The promise of good tests In search of good Test doubles PART 2 CATALOG Readability Maintainability Trustworthiness PART 3 DIVERSIONS Testable design Writing tests in other JVM languages Speeding up test execution

Java Testing with Spock Packt Publishing Ltd

Gain insight into how hexagonal architecture can help to keep the cost of development low over the complete lifetime of an application Key Features Explore ways to make your software flexible, extensible, and adaptable Learn new concepts that you can easily blend with your own software development style Develop the mindset of building maintainable solutions instead of taking shortcuts Book Description We would all like to build software architecture that yields adaptable and flexible software with low development costs. But, unreasonable deadlines and shortcuts make it very hard to create such an architecture. Get Your Hands Dirty on Clean Architecture starts with a discussion about the conventional layered architecture style and its disadvantages. It also talks about the advantages of the domain-centric architecture styles of Robert C. Martin's Clean Architecture and Alistair Cockburn's Hexagonal Architecture. Then, the book dives into hands-on chapters that show you how to manifest a hexagonal architecture in actual code. You'll learn in detail about different mapping strategies between the layers of a hexagonal architecture and see how to assemble the architecture elements into an application. The later chapters demonstrate how to enforce architecture boundaries. You'll also learn what shortcuts produce what types of technical debt and how, sometimes, it is a good idea to willingly take on those debts. After reading this book, you'll have all the knowledge you need to create applications using the hexagonal architecture style of web development. What you will learn Identify potential shortcomings of using a layered architecture Apply methods to enforce architecture boundaries

Find out how potential shortcuts can affect the software architecture Produce arguments for when to use which style of architecture Structure your code according to the architecture Apply various types of tests that will cover each element of the architecture Who this book is for This book is for you if you care about the architecture of the software you are building. To get the most out of this book, you must have some experience with web development. The code examples in this book are in Java. If you are not a Java programmer but can read object-oriented code in other languages, you will be fine. In the few places where Java or framework specifics are needed, they are thoroughly explained.
Simple, Rapid, Effective, and Scalable Pragmatic Bookshelf
Master high quality software development driven by unit tests About This Book Design and implement robust system components by means of the de facto unit testing standard in Java Reduce defect rate and maintenance effort, plus simultaneously increase code quality and development pace Follow a step-by-step tutorial imparting the essential techniques based on real-world scenarios and code walkthroughs Who This Book Is For No matter what your specific background as a Java developer, whether you're simply interested in building up a safety net to reduce regressions of your desktop application or in improving your server-side reliability based on robust and reusable components, unit testing is the way to go. This book provides you with a comprehensive but concise entrance advancing your knowledge step-wise to a professional level. What You Will Learn Organize your test infrastructure and resources reasonably Understand and write well structured tests Decompose your requirements into small and independently testable units Increase your testing efficiency with on-the-fly generated stand-in components and deal with the particularities of exceptional flow Employ runners to adjust to specific test demands Use rules to increase testing safety and reduce boilerplate Use third party supplements to improve the expressiveness of your verification statements In Detail JUnit has matured to become the most important tool when it comes to automated developer tests in Java. Supported by all IDEs and build systems, it empowers programmers to deliver software features reliably and efficiently. However, writing good unit tests is a skill that needs to be learned; otherwise it's all too easy to end up in gridlocked development due to messed up production and testing code. Acquiring the best practices for unit testing will help you to prevent such problems and lead your projects to success with respect to quality and costs. This book explains JUnit concepts and best practices applied to the test first approach, a foundation for high quality Java components delivered in time and budget. From the beginning you'll be guided continuously through a practically relevant example and pick up background knowledge and development techniques step by step. Starting with the basics of tests organization you'll soon comprehend the necessity of well structured tests and delve into the relationship of requirement decomposition and the many-faceted world of test double usage. In conjunction with third-party tools you'll be trained in writing your tests efficiently, adapt your test case environment to particular demands and increase the expressiveness of your verification statements. Finally, you'll experience continuous integration as the perfect complement to support short feedback cycles and quality related reports for your whole team. The tutorial gives a profound entry point in the essentials of unit testing with JUnit and prepares you for test-related daily work challenges. Style and approach This is an intelligible tutorial based on an ongoing and non-trivial development example. Profound introductions of concepts and techniques are provided stepwise as the programming challenges evolve. This allows you to reproduce and practice the individual skills thoroughly.
A hands-on guide to creating clean web applications with code examples in Java Packt Publishing Ltd
Summary The Art of Unit Testing, Second Edition guides you step by step from writing your first simple tests to developing robust test sets that are maintainable, readable, and trustworthy. You'll master the foundational ideas and quickly move to high-value subjects like mocks, stubs, and isolation, including frameworks such as Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, working with legacy code, and even "untestable" code. Along the way, you'll learn about integration testing and techniques and tools for testing databases and other technologies. About this Book You know you should be unit testing, so why aren't you doing it? If you're new to unit testing, if you find unit testing tedious, or if you're just not getting enough payoff for the effort you put into it, keep reading. The Art of Unit Testing, Second Edition guides you step by step from writing your first simple unit tests to building complete test sets that are maintainable, readable, and trustworthy. You'll move quickly to more complicated subjects like mocks and stubs, while learning to use isolation (mocking) frameworks like Moq, FakeItEasy, and Typemock Isolator. You'll explore test patterns and organization, refactor code applications, and learn how to test "untestable" code. Along the way, you'll learn about integration testing and techniques for testing with databases. The examples in the book use C#, but will benefit anyone using a statically typed language such as Java or C++. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. What's Inside Create readable, maintainable, trustworthy tests Fakes, stubs, mock objects, and isolation (mocking) frameworks Simple dependency injection techniques Refactoring legacy code About the Author Roy Osherove has been coding for over 15 years, and he consults and trains teams worldwide on the gentle art of unit testing and test-driven development. His blog is at ArtOfUnitTesting.com. Table of Contents PART 1 GETTING STARTED The basics of unit testing A first unit test PART 2 CORE TECHNIQUES Using stubs to break dependencies Interaction testing using mock objects Isolation (mocking) frameworks Digging deeper into isolation frameworks PART 3 THE TEST CODE Test hierarchies and organization The pillars of good unit tests PART 4 DESIGN AND PROCESS Integrating unit testing into the organization Working with legacy code Design and testability
Mockito Cookbook Simon and Schuster
The desert planet has begun to grow green and lush. The life-giving spice is abundant. The nine-year-old royal twins, possessing their father's supernatural powers, are being groomed as Messiahs. But there are those who think the Imperium does not need Messiahs ...
Mastering Unit Testing Using Mockito and JUnit Apress
If you are an application developer with some experience in software testing and want to learn more about testing frameworks, then this technology and book is for you. Mockito for Spring will be perfect as your next step towards becoming a competent software tester with Spring and Mockito.
Build modern, cloud-native, and distributed systems using Spring Boot Packt Publishing Ltd
If you need a reliable tool for technical documentation, this clear

and concise reference will help you take advantage of DocBook, the popular XML schema originally developed to document computer and hardware projects. DocBook 5.0 has been expanded and simplified to address documentation needs in other fields, and it's quickly becoming the tool of choice for many content providers. DocBook 5: The Definitive Guide is the complete, official documentation of DocBook 5.0. You'll find everything you need to know to use DocBook 5.0's features-including its improved content model-whether you're new to DocBook or an experienced user of previous versions. Learn how to write DocBook XML documents Understand DocBook 5.0's elements and attributes, and how they fit together Determine whether your documents conform to the DocBook schema Learn about options for publishing DocBook to various output formats Customize the DocBook schema to meet your needs Get additional information about DocBook editing and processing

Real-time apps and microservices with the Kafka Streams API

"O'Reilly Media, Inc."

In test driven development, you first write an executable test ofwhat your application code must do. Only then do you write thecode itself and, with the test spurring you on, you improve yourdesign. In acceptance test driven development (ATDD), you usethe same technique to implement product features, benefiting fromiterative development, rapid feedback cycles, and better-definedrequirements. TDD and its supporting tools and techniques leadto better software faster. Test Driven brings under one cover practical TDD techniquesdistilled from several years of community experience. With examplesin Java and the Java EE environment, it explores both the techniquesand the mindset of TDD and ATDD. It uses carefully chosen examplesto illustrate TDD tools and design patterns, not in the abstractbut concretely in the context of the technologies you face at work.It is accessible to TDD beginners, and it offers effective and less wellknown techniques to older TDD hands. 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 Learn hands-on to test drive Java code How to avoid common TDD adoption pitfalls Acceptance test driven development and the Fit framework How to test Java EE components-Servlets, JSPs, and SpringControllers Tough issues like multithreaded programs and data access code