

Uml Et Les Design Patterns

Ce livre de ré fé rence en mati è re d'è pens è e objet ' est une introduction pratique à l'analyse et la conception orient é es objet (A/COO) au moyen d'UML et des design patterns. Il propose de suivre une d é marche it é rative et incr é mentale bien d é finie, le Processus Unifié é , qui guide pas à pas utilisateur, de la sp é cification des besoins au code de l'application. Le Processus Unifié é fait appel à UML, le langage standard de repr é sentation graphique pour les projets de conception logicielle. L'auteur propose une description compl è te de ces design patterns, qui constituent les "meilleurs pratiques" que les experts en A/COO appliquent au d é veloppement de syst è mes. Parmi ces patterns, se trouvent les c è l è bres patterns GOF et GRASP, dont les modalit é s d'application sont ici d é tail é es. Cette troisi è me é dition comprend plus particuli è rement : Des d é veloppements plus importants sur la ma trise de l'A/COO au travers des é tudes de cas ; Les am é liorations é es à UML 2, à la mod é lisation agile, au d é veloppement pilot é par les tests, et au refactoring; De nombreuses explications sur la fa ç on de combiner le d é veloppement it é ratif et é volutif à l'A/COO; Des indications pour appliquer le Processus Unifié ; Un nouveau chapitre sur l' é volution des besoins L'ouvrage se caract é rise par l'analyse compl è te de deux é tudes de cas, qui illustrent tout le processus d'A/COO: techniques é s, patterns essentiels, notation UML. Il adresse aux d é veloppeurs et aux é tudiants en informatique ou en g é nie logiciel qui connaissent un langage de programmation objet mais sont novices en mati è re d'A/COO, ainsi qu' à ceux qui d é sirent apprendre ou approfondir la notation UML, l'application des patterns et les principes d'analyse et de conception.

2012 Jolt Award Finalist! Even experienced software professionals find it difficult to apply patterns in ways that deliver substantial value to their organizations. In Elemental Design Patterns, Jason McK. Smith addresses this problem head-on, helping developers harness the true power of patterns, map them to real software implementations more cleanly and directly, and achieve far better results. Part tutorial, part example-rich cookbook, this resource will help developers, designers, architects, and analysts successfully use patterns with a wide variety of languages, environments, and problem domains. Every bit as important, it will give them the deeper appreciation for the work they ' ve chosen to pursue. Smith presents the crucial missing link that patterns practitioners have needed: a foundational collection of simple core patterns that are broken down to their core elements. If you work in software, you may already be using some of these elemental design patterns every day. Presenting them in a comprehensive methodology for the first time, Smith names them, describes them, explains their importance, helps you compare and choose among them, and offers a framework for using them together. He also introduces an innovative Pattern Instance Notation diagramming system that makes it easier to work with patterns at many levels of granularity, regardless of your goals or role. If you ' re new to patterns, this example-rich approach will help you master them piece by piece.

logically and intuitively. If you're already familiar with, explains how his elemental patterns can be composed into conventional design patterns, and introduces highly productive new ways to apply ideas you ' ve already encountered. No matter what your level of experience, this infinitely practical book will help you transform abstract patterns into high-value solutions. This revised and enlarged edition is a classic in OGD. Testament scholarship reflects the most up-to-date research on the prototypic books and offers substantially expanded discussions of important new insight on UML and the other protoghs.

John Hunt's book guides you through the use of the UML, and the Unified Process and their application to Java systems. Key topics focus explicitly on applying the notation and the method to Java. The book is clearly structured and written, making it ideal for practitioners. This second edition is considerably revised and extended and includes examples taken from the latest version of Rational Rose and Together. Considers how Agile Modelling fits with the Unified Process, and presents Design Patterns Self contained -- covers both the Unified Process and UML in one book Includes real-world case studies Written by an experienced author and industry expert Ideal for students on Software Engineering courses

Design Pattern Formalization Techniques

Building Better Software with Archetype Patterns and UML

Les 23 mod è les de conception : descriptions et solutions illustr é es en UML2

Design Patterns in C#

Applying UML and Patterns

A UML Pattern Language

Design Patterns are a type of pattern used in the initial design phase of an object-oriented development project Documents 46 Visual Basic .NET design patterns, including 20 that have never before been published Features case studies that demonstrate how to use design patterns effectively in the real world-and even explains where not to use design patterns Companion Web site includes all code and UML models from the book as well as links to appropriate software downloads

Java developers know that design patterns offer powerful productivity benefits but few books have been specific enough to address their programming challenges. With 'Java Design Patterns', there's finally a hands-on guide focused specifically on real-world Java development. The book covers three main categories of design patterns--creational, structural, and behavioral--and the example programs and useful variations can be found on the accompanying CD-ROM.

This is the only UML book for Visual Basic developers that covers design patterns. It gives readers design techniques that will make their code more modifiable and reusable in all kinds of applications.

This is a practical tutorial to writing Visual Basic (VB6 and VB.NET) programs using some of the most common design patterns. This book also provides a convenient way for VB6 programmers to migrate to VB.NET and use its more powerful object-oriented features. Organized as a series of short chapters that each describe a design pattern, Visual Basic Design Patterns provides one or more complete working visual examples of programs using that pattern, along with UML diagrams illustrating how the classes interact. Each example is a visual program that students can run and study on the companion CD making the process as concrete as possible.

Design Patterns Explained

A Tutorial

C# 3.0 Design Patterns

Guide to the Unified Process featuring UML, Java and Design Patterns

VB 6.0 and VB.NET

An Introduction to Unified Process and Design Patterns

Welcome to our latest book. Tis Introduction will discuss the purpose of this book and the various sections into which it has been divided. Designing an application is a complex task and, overtime, as application have become more complex, certain standards for analyzing business problems and designing the software solutions to fix them have come to the forefront. This book covers two of those standards, design patterns and the Unified Modeling Language(UML) diagrams used to document them.

Software -- Software Engineering

Teaches developers to build J2EE applications using the leading J2EE application server in 21 straightforward, example-driven lessons.

For students learning to use the object environment, this book is a companion to the original Design Patterns text tailored to the C#. This book is an application book, rather than a theoretical one. It is written for students who want to gain a better understanding of the patterns described in the seminal design patterns book by Gamma et al. The book's intent is to give students the confidence and know-how to apply the original 23 patterns identified in the Gamma book, with all code examples provided in C#.

Java Design

A Brain-Friendly Guide

Real-time Design Patterns

Design Patterns pour C#

Design Patterns avec UML 2 et C#6

UML 2 et les design patterns

Ce livre a été conçu comme une présentation simple et efficace des 23 modèles de conception (design patterns). Les modèles de conception répondent à des problèmes de conception de logiciels dans le cadre de la programmation par objets. Ce sont des solutions connues et éprouvées dont la conception provient de l'expérience de programmeurs. Ce livre a une double finalité. En premier, il permet au lecteur d'acquérir une connaissance des éléments essentiels des 23 modèles de conception, notamment leur structure générale sous forme d'un diagramme de classes UML. En second, le lecteur peut affiner ses connaissances en examinant les exemples pratiques de programmation mis en oeuvre (en langage C# dans

l'environnement de développement intégré Visual Studio 2015 Community), et en étudiant les compositions et les variantes expliquées et détaillées.

Ce livre présente de façon concise et pratique les 23 modèles de conception (design patterns) fondamentaux en les illustrant par des exemples pertinents et rapides à appréhender. Chaque exemple est décrit en UML et en Java sous la forme d'un petit programme complet et exécutable. Pour chaque pattern, l'auteur détaille son nom, le problème correspondant, la solution proposée, ses domaines d'application et sa structure générique. Le livre s'adresse aux concepteurs et développeurs en Programmation Orientée Objet. Pour bien appréhender, il est préférable de disposer de connaissances sur les principaux éléments des diagrammes de classes UML et de la version 1.6 ou ultérieure du langage Java. Le livre est organisé en 5 parties. Dans la première l'auteur introduit la notion de pattern de conception. Les trois parties suivantes détaillent chacune des trois familles des patterns de conception : les patterns de construction, les patterns de structuration et les patterns de comportement. Pour finir, la dernière partie présente trois variantes de patterns existants, montrant la grande souplesse de mise en oeuvre de ces modèles. Le pattern composé MVC (Model-View-Controller) y est également présenté. Cette nouvelle édition du livre s'enrichit d'un chapitre sur des concepts avancés de la programmation par objets qui permet au lecteur d'approfondir ses connaissances. Les exemples utilisés dans le livre sont issus d'une application de vente en ligne de véhicules et sont un téléchargement sur le site editions-en.

"This book introduces the fundamentals of software contracts and illustrates how Design by Contract contributes to the optimal use of design patterns in a quality-oriented software engineering process. The Design by Contract approach to software construction provides a methodological guideline for building systems that are robust, modular, and simple." "Readers will find value in the book's overview of the Object Constraint Language, a precise modeling language that allows Design by Contract to be used with the industry standard Unified Modeling Language (UML). Although written in Eiffel, this book makes an excellent companion for developers who are using languages such as Java and UML. Throughout the book the authors discuss specific implementation issues and provide complete, ready-to-be-compiled examples of the use of each pattern." "They introduce design patterns and Design by Contract in the context of software engineering, and show how these tools are used to guide and document system design."--BOOK JACKET.Title Summary field provided by Blackwell North America, Inc. All Rights Reserved

A UML Pattern Language pairs the software design pattern concept with the Unified Modeling Language (UML) to offer a tool set for software professionals practicing both system modeling and software development. This book provides: a collection of patterns in the domain of system modeling, including those that are useful to management, operations, and deployment teams, as well as to software developers; a survey of the development of patterns and the UML; a discussion of the underlying theory of the patterns and instructions for using the language; a thorough exploration of the design process and model-driven development. A UML Pattern Language recognizes that design and modeling have become equal partners with programming and coding in the enterprise of software development. Providing both an understanding of the work of design and the way patterns and the UML combine to facilitate design.

Les 23 modèles de conception : descriptions et solutions illustrées en UML 2 et Java

Les 23 modèles de conception

Use the Power of C# 3.0 to Solve Real-World Problems

Learning Python Design Patterns

Applying UML and Patterns Training Course

Enterprise Patterns and MDA

Learn to combine security theory and code to produce secure systems Security is clearly a crucial issue to consider during the design and implementation of any distributed software architecture. Security patterns are increasingly being used by developers who take security into serious consideration from the creation of their work. Written by the authority on security patterns, this unique book examines the structure and purpose of security patterns, illustrating their use with the help of detailed implementation advice, numerous code samples, and descriptions in UML. Provides an extensive, up-to-date catalog of security patterns logically organized by type of application and use case, and how to use them. Explains when and how to use them. Includes a chapter on how to use them in the context of design patterns. Coverage includes Using Adapter to provide consistent interfaces to clients Using Facade to simplify the use of reusable toolkits Understanding the role of Bridge in Java database connectivity The Observer pattern Author is well known and highly respected in the field of security and an expert on security patterns Security Patterns in Practice shows you how to confidently develop a secure system step by step.

Many formal approaches for pattern specification are emerging as a means to cope with the inherent shortcomings of informal description. Design Pattern Formalization Techniques presents multiple methodological, formal approaches for pattern specification, emphasizing on software development processes for engineering disciplines. Design Pattern Formalization Techniques focuses on formalizing the solution element of patterns, providing tangible benefits to pattern users, researchers, scholars, academicians, practitioners and students working in the field of design patterns and software reuse.Design Pattern Formalization Techniques explains details on several specification languages, allowing readers to choose the most suitable formal technique to solve their specific inquiries.

Social scientists, whether earnest graduate students or tenured faculty members, clearly know the rules that govern good writing. But for some reason they choose to ignore those guidelines and churn out turgid, pompous, and obscure prose. Distinguished sociologist Howard S. Becker, true to his calling, looks for an explanation of this behavior not in the psyches of his colleagues but in the structure of his profession. In this highly personal and inspirational volume he considers academic writing as a social activity. Both the means and the reasons for writing a thesis or article or book are socially structured by the organization of graduate study, the requirements for publication, and the conditions for promotion, and the pressures arising from these situations create the writing style so often lampooned and lamented. Drawing on his thirty-five years' experience as a researcher, writer, and teacher, Becker exposes the foibles of the academic profession to the light of sociological analysis and gentle humor. He also offers eminently useful suggestions for ways to make social scientists better and more productive writers. Among the topics discussed are how to overcome the paralyzing fears of chaos and ridicule that lead to writer's block; how to rewrite and revise, again and again; how to adopt a persona compatible with lucid prose; how to deal with that academic bugaboo, "the literature." There is also a chapter by Pamela Richards on the personal and professional risks involved in scholarly writing. In recounting his own trials and errors Becker offers his readers not a model to be slavishly imitated but an example to inspire. Throughout, his focus is on the elusive work habits that contribute to good writing, not the more easily learned rules of grammar and punctuation. Although his examples are drawn from sociological literature, his conclusions apply to all fields of social science, and is indeed to all areas of scholarly endeavor. The message is clear: you don't have to write like a social scientist to be one.

This workbook approach deepens understanding, builds confidence, and strengthens readers' skills. It covers all five categories of design pattern intent: interfaces, responsibility, construction, operations, and extensions.

Visual Basic Design Patterns

Elements of Reusable Object-Oriented Software with Applying Uml and Patterns:An Introduction to Object-Oriented Analysis and Design and the Unified Process

Security Patterns in Practice

A UML Pattern Language

A comprehensive guide to building smart and reusable code in Java

Object-Oriented Analysis and Design Using UML

Understand Gang of Four, architectural, functional, and reactive design patterns and how to implement them on modern Java platforms, such as Java 12 and beyond Key Features Learn OOP, functional, and reactive patterns for creating readable and maintainable code Explore architectural patterns and practices for building scalable and reliable applications Tackle all kinds of performance-related issues and streamline development using design patterns Book Description Java design patterns are reusable and proven solutions to software design problems. This book covers over 60 battle-tested design patterns used by developers to create functional, reusable, and flexible software. Hands-On Design Patterns with Java starts with an introduction to the Unified Modeling Language (UML), and delves into class and object diagrams with the help of detailed examples. You'll study concepts and approaches to object-oriented programming (OOP) and OOP design patterns to build robust applications. As you advance, you'll explore the categories of GOF design patterns, such as behavioral, creational, and structural, that help you improve code readability and enable large-scale reuse of software. You'll also discover how to work effectively with microservices and service-oriented architectures, each of which is thoroughly explained and accompanied by real-world programming solutions. By the end of the book, you'll be able to speed up your software development process using the right design patterns, and you'll be comfortable working on scalable and maintainable projects of any size. What you will learn Understand the significance of design patterns for software engineering Visualize software design with UML diagrams Strengthen your understanding of OOP to create reusable software systems Discover GOF design patterns to develop scalable applications Examine programming challenges and the design patterns that solve them Explore architectural patterns for microservices and cloud development Who this book is for If you are a developer who wants to learn how to write clear, concise, and effective code for building production-ready applications, this book is for you.

Familiarity with the fundamentals of Java is assumed.

UML et les Design Patterns fournit aux développeurs et aux étudiants les connaissances indispensables pour comprendre et maîtriser l'analyse et la conception orientées objet (A/COO). Son objectif ne se limite pas à la création de diagrammes et à la notation UML : il vise à leur application réelle dans le contexte de la conception de systèmes logiciels. Craig Larman, expert de renom en technologie objet et en méthodes itératives, présente ici une unique étude de cas homogène, exposant progressivement les techniques capitales de l'analyse et de la conception orientées objet, tout en mettant l'accent sur les activités, les outils et les plus essentiels. Grâce à ses nombreux exemples et à une étude approfondie des patterns, des cas d'utilisation, du Processus Unifié et de l'analyse structurale, cet ouvrage constitue un exposé clair et pratique sur la façon de penser et de concevoir en termes d'objets.

This Three-Volume-Set constitutes the refereed proceedings of the Second International Conference on Software Engineering and Computer Systems, ICSRECS 2011, held in Kuantan, Malaysia, in June 2011. The 190 revised full papers presented together with invited papers in the three volumes were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on software engineering; network; bioinformatics and e-health; biometrics technologies; web engineering; neural network; parallel and distributed; e-learning; ontology; image processing; information and data management; engineering; software security; graphics and multimedia; databases; algorithms; signal processing; software design/testing; e- technology; ad hoc networks; social networks; software process modeling; miscellaneous topics in software engineering and computer systems.

Design Patterns in Java" gives you the hands-on practice and deep insight you need to fully leverage the significant power of design patterns in any Java software project. The perfect complement to the classic Design Patterns, this learn-by-doing workbook applies the latest Java features and best practices to all of the original 23 patterns identified in that groundbreaking text. Drawing on their extensive experience as Java instructors and programmers, Steve Metsker and Bill Wake illuminate each pattern with real Java programs, clear UML diagrams, and compelling exercises. You'll move quickly from theory to application-learning how to improve your code and refactor existing code for simplicity, manageability, and performance. Coverage includes Using Adapter to provide consistent interfaces to clients Using Facade to simplify the use of reusable toolkits Understanding the role of Bridge in Java database connectivity The Observer pattern

Model-View-Controller, and GUI behavior Java Remote Method Invocation (RMI) and the Proxy pattern Streamlining designs using the Chain of Responsibility pattern Using patterns to go beyond Java's built-in constructor features Implementing Undo capabilities with Memento Using the State pattern to manage state more cleanly and simply Optimizing existing codebases with extension patterns Providing thread-safe iteration with the Iterator pattern Using Visitor to define new operations without changing hierarchy classes If you're a Java programmer wanting to save time while writing better code, this book's techniques, tips, and clear explanations and examples will help you harness the power of patterns to improve every program you write, design, or maintain. All source code is available for download at http://www.ozinoz.com.

Design Patterns and Contracts

Learning PHP Design Patterns

An Introduction to Object-oriented Analysis and Design and Iterative Development

Hands-On Design Patterns with Java

Design Patterns en PHP

Objects, UML, and Process

Leverage the power of Python design patterns to solve real-world problems in software architecture and design About This Book Understand the structural, creational, and behavioral Python design patterns Get to know the context and application of design patterns to solve real-world problems in software architecture, design, and application development Get practical exposure through sample implementations in Python v3.5 For the design patterns featured Who This Book Is For This book is for Software architects and Python application developers who are passionate about software design. It will be very useful to engineers with beginner level proficiency in Python and who love to work with Python 3.5 What You Will Learn Enhance your skills to create better software architecture Understand proven solutions to commonly occurring design issues Explore the design principles that form the basis of software design, such as loose coupling, the Hollywood principle and the Open Close principle among others Delve into the object-oriented programming concepts and find out how they are used in software applications Develop an understanding of Creational Design Patterns and the different object creation methods that help you solve issues in software development Use Structural Design Patterns and find out how objects and classes interact to build larger applications Focus on the interaction between objects with the command and observer patterns Improve the productivity and code base of your application using Python design patterns In Detail With the increasing focus on optimized software architecture and design it is important that software architects think about optimizations in object creation, code structure, and interaction between objects at the architecture or design level. This makes sure that the cost of software maintenance is low and code can be easily reused or is adaptable to change. The key to this is reusability and low maintenance in design patterns. Building on the success of the previous edition, Learning Python Design Patterns, Second Edition will help you implement real-world scenarios with Python's latest release, Python v3.5. We start by introducing design patterns from the Python perspective. As you progress through the book, you will learn about Singleton patterns, Factory patterns, and Facade patterns in detail. After this, we'll look at how to control object access with proxy patterns. It also covers observer patterns, command patterns, and compound patterns. By the end of the book, you will have enhanced your professional abilities in software architecture, design, and development. Style and approach This is an easy-to-follow guide to design patterns with hands-on examples of real-world scenarios and their implementation in Python v3.5. Each topic is explained and placed in context, and for the more inquisitive, there are more details on the concepts used.

Discusses how the unified modeling language (UML) can be used during the implementation stage of the Java software development lifecycle. The book focuses on refactoring or cleaning up the design of existing code, and addresses the most common and significant decisions made during enterprise Java development. The author identifies initial analysis classes, introduces the UML sequence diagram, and demonstrates architectural modeling. Annotation copyrighted by Book News Inc., Portland, OR.

If you want to speed up the development of your .NET applications, you're ready for C# design patterns: elegant, accepted and proven ways to tackle common programming problems. This practical guide offers you a clear introduction to the classic object-oriented design patterns, and explains how to use the latest features of C# 3.0 to code them. C# Design Patterns draws on new C# 3.0 language and .NET 3.5 framework features to implement the 23 fundamental patterns known to working developers. You get plenty of case studies that reveal how each pattern is used in practice, and an insightful comparison of patterns and where they would be best used or combined. This well-organized and illustrated book includes: An explanation of design patterns and why they're used, with tables and guidelines to help you choose one pattern over another Illustrated coverage of each classic Creational, Structural, and Behavioral design pattern, including its representation in UML and the roles of its various players C# 3.0 features introduced by example and summarized in sidebars for easy reference Examples of each pattern at work in a real .NET 3.5 program available for download from O'Reilly and the author's companion web site Quizzes and exercises to test your understanding of the material. With C# 3.0 Design Patterns, you learn to make code correct, extensible and efficient to save time up front and eliminate problems later. If your business relies on efficient application development and quality code, you need C# Design Patterns.

Second Edition of the UML video course based on the book Applying UML and Patterns. This VTC will focus on object-oriented analysis and design, not just drawing UML.

Learn Design Patterns That Enable the Building of Large-Scale Software Architectures

Teach Yourself BEA WebLogic Server 7.0 in 21 Days

Elemental Design Patterns

Java Design Patterns

Design Patterns en Java

Design Patterns

Enterprise Patterns and MDA teaches you how to customize any archetype pattern--such as Customer, Product, and Order--to reflect the idiosyncrasies of your own business environment. Because all the patterns work harmoniously together and have clearly documented relationships to each other, you'll come away with a host of reusable solutions to common problems in business--software design. This book shows you how using a pattern or a fragment of a pattern can save you months of work and help you avoid costly errors. You'll also discover how--when used in literate modeling--patterns can solve the difficult challenge of communicating UML models to broad audiences. The configurable patterns can be used manually to create executable code. However, the authors draw on their extensive experience to show you how to tap the significant power of MDA and UML for maximum automation. Not surprisingly, the patterns included in this book are highly valuable: a blue-chip company recently valued a similar, but less mature, set of patterns at hundreds of thousands of dollars. Use this practical guide to increase the efficiency of your designs and to create robust business applications that can be applied immediately in a business setting.

Create various design patterns to master the art of solving problems using Java Key Features This book demonstrates the shift from OOP to functional programming and covers reactive and functional patterns in a clear and step-by-step manner All the design patterns come with a practical use case as part of the explanation, which will improve your productivity Tackle all kinds of performance-related issues and streamline your development Book Description Having a knowledge of design patterns enables you, as a developer, to improve your code base, promote code reuse, and make the architecture more robust. As languages evolve, new features take time to fully understand before they are adopted en masse. The mission of this book is to ease the adoption of the latest trends and provide good practices for programmers. We focus on showing you the practical aspects of smarter coding in Java. We'll start off by going over object-oriented (OOP) and functional programming (FP) paradigms, moving on to describe the most frequently used design patterns in their classical format and explain how Java's functional programming features are changing them. You will learn to enhance implementations by mixing OOP and FP, and finally get to know about the reactive programming model, where FP and OOP are used in conjunction with a view to writing better code.

Gradually, the book will show you the latest trends in architecture, moving from MVC to microservices and serverless architecture. We will finish off by highlighting the new Java features and best practices. By the end of the book, you will be able to efficiently address common problems faced while developing applications and be comfortable working on scalable and maintainable projects of any size. What you will learn Understand the OOP and FP paradigms Explore the traditional Java design patterns Get to know the new functional features of Java See how design patterns are changed and affected by the new features Discover what reactive programming is and why is it the natural augmentation of FP Work with reactive design patterns and find the best ways to solve common problems using them See the latest trends in architecture and the shift from MVC to serverless applications Use best practices when working with the new features Who this book is for This book is for those who are familiar with Java development and want to be in the driver's seat when it comes to modern development techniques. Basic OOP Java programming experience and elementary familiarity with Java is expected.

Design patterns are elegant, adaptable, and reusable solutions to everyday software development problems. Programmers use design patterns to organize objects in programs, making them easier to write and modify. C# Design Patterns: A Tutorial is a practical guide to writing C# programs using the most common patterns. This tutorial begins with clear and concise introductions to C#, object-oriented programming and inheritance, and UML diagrams. Each chapter that follows describes one of twenty-three design patterns, recommends when to use it, and explains the impact that it will have on the larger design. The use of every pattern is demonstrated with simple example programs. These programs are illustrated with screen shots and UML diagrams displaying how the classes interact. Design patterns will have an immediate impact on your work as you learn the following: Applying design patterns effectively in your day-to-day programming Using patterns to create sophisticated, robust C# programs The interaction of classes as demonstrated by UML diagrams Advancing your programming skills using design patterns Design patterns will not only enhance your productivity, but once you see how quickly and easily object-oriented code can be recycled, they will become an everyday part of your C# programming.

Build server-side applications more efficiently--and improve your PHP programming skills in the process--by learning how to use design patterns in your code. This book shows you how to apply several object-oriented patterns through simple examples, and demonstrates many of how to use the latest features of PHP 5.4 to code them. C# Design Patterns draws on new C# 3.0 language and .NET 3.5 framework features to implement the 23 fundamental patterns known to working developers. You get plenty of case studies that reveal how each pattern is used in practice, and an insightful comparison of patterns and where they would be best used or combined. This well-organized and illustrated book includes: An explanation of design patterns and why they're used, with tables and guidelines to help you choose one pattern over another Illustrated coverage of each classic Creational, Structural, and Behavioral design pattern, including its representation in UML and the roles of its various players C# 3.0 features introduced by example and summarized in sidebars for easy reference Examples of each pattern at work in a real .NET 3.5 program available for download from O'Reilly and the author's companion web site Quizzes and exercises to test your understanding of the material. With C# 3.0 Design Patterns, you learn to make code correct, extensible and efficient to save time up front and eliminate problems later. If your business relies on efficient application development and quality code, you need C# Design Patterns.

Second Edition of the UML video course based on the book Applying UML and Patterns. This VTC will focus on object-oriented analysis and design, not just drawing UML.

Learn Design Patterns That Enable the Building of Large-Scale Software Architectures

Teach Yourself BEA WebLogic Server 7.0 in 21 Days

Elemental Design Patterns

Java Design Patterns

Design Patterns en Java

Design Patterns

Enterprise Patterns and MDA teaches you how to customize any archetype pattern--such as Customer, Product, and Order--to reflect the idiosyncrasies of your own business environment. Because all the patterns work harmoniously together and have clearly documented relationships to each other, you'll come away with a host of reusable solutions to common problems in business--software design. This book shows you how using a pattern or a fragment of a pattern can save you months of work and help you avoid costly errors. You'll also discover how--when used in literate modeling--patterns can solve the difficult challenge of communicating UML models to broad audiences. The configurable patterns can be used manually to create executable code. However, the authors draw on their extensive experience to show you how to tap the significant power of MDA and UML for maximum automation. Not surprisingly, the patterns included in this book are highly valuable: a blue-chip company recently valued a similar, but less mature, set of patterns at hundreds of thousands of dollars. Use this practical guide to increase the efficiency of your designs and to create robust business applications that can be applied immediately in a business setting.

Create various design patterns to master the art of solving problems using Java Key Features This book demonstrates the shift from OOP to functional programming and covers reactive and functional patterns in a clear and step-by-step manner All the design patterns come with a practical use case as part of the explanation, which will improve your productivity Tackle all kinds of performance-related issues and streamline your development Book Description Having a knowledge of design patterns enables you, as a developer, to improve your code base, promote code reuse, and make the architecture more robust. As languages evolve, new features take time to fully understand before they are adopted en masse. The mission of this book is to ease the adoption of the latest trends and provide good practices for programmers. We focus on showing you the practical aspects of smarter coding in Java. We'll start off by going over object-oriented (OOP) and functional programming (FP) paradigms, moving on to describe the most frequently used design patterns in their classical format and explain how Java's functional programming features are changing them. You will learn to enhance implementations by mixing OOP and FP, and finally get to know about the reactive programming model, where FP and OOP are used in conjunction with a view to writing better code.

Gradually, the book will show you the latest trends in architecture, moving from MVC to microservices and serverless architecture. We will finish off by highlighting the new Java features and best practices. By the end of the book, you will be able to efficiently address common problems faced while developing applications and be comfortable working on scalable and maintainable projects of any size. What you will learn Understand the OOP and FP paradigms Explore the traditional Java design patterns Get to know the new functional features of Java See how design patterns are changed and affected by the new features Discover what reactive programming is and why is it the natural augmentation of FP Work with reactive design patterns and find the best ways to solve common problems using them See the latest trends in architecture and the shift from MVC to serverless applications Use best practices when working with the new features Who this book is for This book is for those who are familiar with Java development and want to be in the driver's seat when it comes to modern development techniques. Basic OOP Java programming experience and elementary familiarity with Java is expected.

Design patterns are elegant, adaptable, and reusable solutions to everyday software development problems. Programmers use design patterns to organize objects in programs, making them easier to write and modify. C# Design Patterns: A Tutorial is a practical guide to writing C# programs using the most common patterns. This tutorial begins with clear and concise introductions to C#, object-oriented programming and inheritance, and UML diagrams. Each chapter that follows describes one of twenty-three design patterns, recommends when to use it, and explains the impact that it will have on the larger design. The use of every pattern is demonstrated with simple example programs. These programs are illustrated with screen shots and UML diagrams displaying how the classes interact. Design patterns will have an immediate impact on your work as you learn the following: Applying design patterns effectively in your day-to-day programming Using patterns to create sophisticated, robust C# programs The interaction of classes as demonstrated by UML diagrams Advancing your programming skills using design patterns Design patterns will not only enhance your productivity, but once you see how quickly and easily object-oriented code can be recycled, they will become an everyday part of your C# programming.

Build server-side applications more efficiently--and improve your PHP programming skills in the process--by learning how to use design patterns in your code. This book shows you how to apply several object-oriented patterns through simple examples, and demonstrates many of how to use the latest features of PHP 5.4 to code them. C# Design Patterns draws on new C# 3.0 language and .NET 3.5 framework features to implement the 23 fundamental patterns known to working developers. You get plenty of case studies that reveal how each pattern is used in practice, and an insightful comparison of patterns and where they would be best used or combined. This well-organized and illustrated book includes: An explanation of design patterns and why they're used, with tables and guidelines to help you choose one pattern over another Illustrated coverage of each classic Creational, Structural, and Behavioral design pattern, including its representation in UML and the roles of its various players C# 3.0 features introduced by example and summarized in sidebars for easy reference Examples of each pattern at work in a real .NET 3.5 program available for download from O'Reilly and the author's companion web site Quizzes and exercises to test your understanding of the material. With C# 3.0 Design Patterns, you learn to make code correct, extensible and efficient to save time up front and eliminate problems later. If your business relies on efficient application development and quality code, you need C# Design Patterns.

Second Edition of the UML video course based on the book Applying UML and Patterns. This VTC will focus on object-oriented analysis and design, not just drawing UML.

Learn Design Patterns That Enable the Building of Large-Scale Software Architectures

Teach Yourself BEA WebLogic Server 7.0 in 21 Days

Elemental Design Patterns

Java Design Patterns

Design Patterns en Java

Design Patterns

Enterprise Patterns and MDA teaches you how to customize any archetype pattern--such as Customer, Product, and Order--to reflect the idiosyncrasies of your own business environment. Because all the patterns work harmoniously together and have clearly documented relationships to each other, you'll come away with a host of reusable solutions to common problems in business--software design. This book shows you how using a pattern or a fragment of a pattern can save you months of work and help you avoid costly errors. You'll also discover how--when used in literate modeling--patterns can solve the difficult challenge of communicating UML models to broad audiences. The configurable patterns can be used manually to create executable code. However, the authors draw on their extensive experience to show you how to tap the significant power of MDA and UML for maximum automation. Not surprisingly, the patterns included in this book are highly valuable: a blue-chip company recently valued a similar, but less mature, set of patterns at hundreds of thousands of dollars. Use this practical guide to increase the efficiency of your designs and to create robust business applications that can be applied immediately in a business setting.

Create various design patterns to master the art of solving problems using Java Key Features This book demonstrates the shift from OOP to functional programming and covers reactive and functional patterns in a clear and step-by-step manner All the design patterns come with a practical use case as part of the explanation, which will improve your productivity Tackle all kinds of performance-related issues and streamline your development Book Description Having a knowledge of design patterns enables you, as a developer, to improve your code base, promote code reuse, and make the architecture more robust. As languages evolve, new features take time to fully understand before they are adopted en masse. The mission of this book is to ease the adoption of the latest trends and provide good practices for programmers. We focus on showing you the practical aspects of smarter coding in Java. We'll start off by going over object-oriented (OOP) and functional programming (FP) paradigms, moving on to describe the most frequently used design patterns in their classical format and explain how Java's functional programming features are changing them. You will learn to enhance implementations by mixing OOP and FP, and finally get to know about the reactive programming model, where FP and OOP are used in conjunction with a view to writing better code.

Gradually, the book will show you the latest trends in architecture, moving from MVC to microservices and serverless architecture. We will finish off by highlighting the new Java features and best practices. By the end of the book, you will be able to efficiently address common problems faced while developing applications and be comfortable working on scalable and maintainable projects of any size. What you will learn Understand the OOP and FP paradigms Explore the traditional Java design patterns Get to know the new functional features of Java See how design patterns are changed and affected by the new features Discover what reactive programming is and why is it the natural augmentation of FP Work with reactive design patterns and find the best ways to solve common problems using them See the latest trends in architecture and the shift from MVC to serverless applications Use best practices when working with the new features Who this book is for This book is for those who are familiar with Java development and want to be in the driver's seat when it comes to modern development techniques. Basic OOP Java programming experience and elementary familiarity with Java is expected.

Design patterns are elegant, adaptable, and reusable solutions to everyday software development problems. Programmers use design patterns to organize objects in programs, making them easier to write and modify. C# Design Patterns: A Tutorial is a practical guide to writing C# programs using the most common patterns. This tutorial begins with clear and concise introductions to C#, object-oriented programming and inheritance, and UML diagrams. Each chapter that follows describes one of twenty-three design patterns, recommends when to use it, and explains the impact that it will have on the larger design. The use of every pattern is demonstrated with simple example programs. These programs are illustrated with screen shots and UML diagrams displaying how the classes interact. Design patterns will have an immediate impact on your work as you learn the following: Applying design patterns effectively in your day-to-day programming Using patterns to create sophisticated, robust C# programs The interaction of classes as demonstrated by UML diagrams Advancing your programming skills using design patterns Design patterns will not only enhance your productivity, but once you see how quickly and easily object-oriented code can be recycled, they will become an everyday part of your C# programming.

Build server-side applications more efficiently--and improve your PHP programming skills in the process--by learning how to use design patterns in your code. This book shows you how to apply several object-oriented patterns through simple examples, and demonstrates many of how to use the latest features of PHP 5.4 to code them. C# Design Patterns draws on new C# 3.0 language and .NET 3.5 framework features to implement the 23 fundamental patterns known to working developers. You get plenty of case studies that reveal how each pattern is used in practice, and an insightful comparison of patterns and where they would be best used or combined. This well-organized and illustrated book includes: An explanation of design patterns and why they're used, with tables and guidelines to help you choose one pattern over another Illustrated coverage of each classic Creational, Structural, and Behavioral design pattern, including its representation in UML and the roles of its various players C# 3.0 features introduced by example and summarized in sidebars for easy reference Examples of each pattern at