

Hands On Design Patterns With C And Net Core Writ

Yeah, reviewing a book **Hands On Design Patterns With C And Net Core Writ** could increase your near friends listings. This is just one of the solutions for you to be successful. As understood, realization does not suggest that you have astounding points.

Comprehending as competently as pact even more than supplementary will offer each success. next-door to, the statement as without difficulty as perception of this Hands On Design Patterns With C And Net Core Writ can be taken as without difficulty as picked to act.

Design Patterns for Embedded Systems in C - Bruce Powel Douglass 2010-11-03

A recent survey stated that 52% of embedded projects are late by 4-5 months. This book can help get those projects in on-time with design patterns. The author carefully takes into account the special concerns found in designing and developing embedded applications specifically concurrency, communication, speed, and memory usage. Patterns are given in UML (Unified Modeling Language) with examples including ANSI C for direct and practical application to C code. A basic C knowledge is a prerequisite for the book while UML notation and terminology is included. General C programming books do not include discussion of the constraints found within embedded system design. The practical examples give the reader an understanding of the use of UML and OO (Object Oriented) designs in a resource-limited environment. Also included are two chapters on state machines. The beauty of this book is that it can help you today. . Design Patterns within these pages are immediately applicable to your project Addresses embedded system design concerns such as concurrency, communication, and memory usage Examples contain ANSI C for ease of use with C programming code

Design Patterns Explained - Alan Shalloway 2004-10-12

"One of the great things about the book is the way the authors explain concepts very simply using analogies rather than programming examples-this has been very inspiring for a product I'm working on: an audio-only introduction to OOP and software development."

-Bruce Eckel "...I would expect that readers with a basic understanding of object-oriented programming and design would find this book useful, before approaching design patterns completely. Design Patterns Explained complements the existing design patterns texts and may perform a very useful role, fitting between introductory texts such as UML Distilled and the more advanced patterns books." -James Noble Leverage the quality and productivity benefits of patterns-without the complexity! Design Patterns Explained, Second Edition is the field's simplest, clearest, most practical introduction to patterns. Using dozens of updated Java examples, it shows programmers and architects exactly how to use patterns to design, develop, and deliver software far more effectively. You'll start with a complete overview of the fundamental principles of patterns, and the role of object-oriented analysis and design in contemporary software development. Then, using easy-to-understand sample code, Alan Shalloway and James Trott illuminate dozens of today's most useful patterns: their underlying concepts, advantages, tradeoffs, implementation techniques, and pitfalls to avoid. Many patterns are accompanied by UML diagrams. Building on their best-selling First Edition, Shalloway and Trott have thoroughly updated this book to reflect new software design trends, patterns, and implementation techniques. Reflecting extensive reader feedback, they have deepened and clarified coverage throughout, and reorganized content for even greater ease of understanding. New and revamped coverage in this edition includes Better ways to start

"thinking in patterns" How design patterns can facilitate agile development using eXtreme Programming and other methods How to use commonality and variability analysis to design application architectures The key role of testing into a patterns-driven development process How to use factories to instantiate and manage objects more effectively The Object-Pool Pattern—a new pattern not identified by the "Gang of Four" New study/practice questions at the end of every chapter Gentle yet thorough, this book assumes no patterns experience whatsoever. It's the ideal "first book" on patterns, and a perfect complement to Gamma's classic Design Patterns. If you're a programmer or architect who wants the clearest possible understanding of design patterns—or if you've struggled to make them work for you—read this book.

Design Patterns in C# - Vaskaran Sarcar
2018-06-21

Get hands-on experience with each Gang of Four design pattern using C#. For each of the patterns, you'll see at least one real-world scenario, a coding example, and a complete implementation including output. In the first part of Design Patterns in C#, you will cover the 23 Gang of Four (GoF) design patterns, before moving onto some alternative design patterns, including the Simple Factory Pattern, the Null Object Pattern, and the MVC Pattern. The final part winds up with a conclusion and criticisms of design patterns with chapters on anti-patterns and memory leaks. By working through easy-to-follow examples, you will understand the concepts in depth and have a collection of programs to port over to your own projects. Along the way, the author discusses the different creational, structural, and behavioral patterns and why such classifications are useful. In each of these chapters, there is a Q&A session that clears up any doubts and covers the pros and cons of each of these patterns. He finishes the book with FAQs that will help you consolidate your knowledge. This book presents the topic of design patterns in C# in such a way that anyone can grasp the idea. What You Will Learn Work with each of the design patterns Implement the design patterns in real-world applications Select an alternative to these patterns by comparing their pros and cons Use Visual Studio

Community Edition 2017 to write code and generate output Who This Book Is For Software developers, software testers, and software architects.

Hands-On Object-Oriented Programming with Kotlin - Abid Khan 2018-10-31

Learn everything you need to know about object-oriented programming with the latest features of Kotlin 1.3 Key Features A practical guide to understand objects and classes in Kotlin Learn to write asynchronous, non-blocking codes with Kotlin coroutines Explore Encapsulation, Inheritance, Polymorphism, and Abstraction in Kotlin Book Description Kotlin is an object-oriented programming language. The book is based on the latest version of Kotlin. The book provides you with a thorough understanding of programming concepts, object-oriented programming techniques, and design patterns. It includes numerous examples, explanation of concepts and keynotes. Where possible, examples and programming exercises are included. The main purpose of the book is to provide a comprehensive coverage of Kotlin features such as classes, data classes, and inheritance. It also provides a good understanding of design pattern and how Kotlin syntax works with object-oriented techniques. You will also gain familiarity with syntax in this book by writing labeled for loop and when as an expression. An introduction to the advanced concepts such as sealed classes and package level functions and coroutines is provided and we will also learn how these concepts can make the software development easy. Supported libraries for serialization, regular expression and testing are also covered in this book. By the end of the book, you would have learnt building robust and maintainable software with object oriented design patterns in Kotlin. What you will learn Get an overview of the Kotlin programming language Discover Object-oriented programming techniques in Kotlin Understand Object-oriented design patterns Uncover multithreading by Kotlin way Understand about arrays and collections Understand the importance of object-oriented design patterns Understand about exception handling and testing in OOP with Kotlin Who this book is for This book is for programmers and developers who wish to learn Object-oriented programming principles and

apply them to build robust and scalable applications. Basic knowledge in Kotlin programming is assumed

[Hands-On Design Patterns with C++](#) - Fedor G. Pikus 2019-01-30

A comprehensive guide with extensive coverage on concepts such as OOP, functional programming, generic programming, and STL along with the latest features of C++ Key Features Delve into the core patterns and components of C++ in order to master application design Learn tricks, techniques, and best practices to solve common design and architectural challenges Understand the limitation imposed by C++ and how to solve them using design patterns Book Description C++ is a general-purpose programming language designed with the goals of efficiency, performance, and flexibility in mind. Design patterns are commonly accepted solutions to well-recognized design problems. In essence, they are a library of reusable components, only for software architecture, and not for a concrete implementation. The focus of this book is on the design patterns that naturally lend themselves to the needs of a C++ programmer, and on the patterns that uniquely benefit from the features of C++, in particular, the generic programming. Armed with the knowledge of these patterns, you will spend less time searching for a solution to a common problem and be familiar with the solutions developed from experience, as well as their advantages and drawbacks. The other use of design patterns is as a concise and an efficient way to communicate. A pattern is a familiar and instantly recognizable solution to specific problem; through its use, sometimes with a single line of code, we can convey a considerable amount of information. The code conveys: "This is the problem we are facing, these are additional considerations that are most important in our case; hence, the following well-known solution was chosen." By the end of this book, you will have gained a comprehensive understanding of design patterns to create robust, reusable, and maintainable code. What you will learn Recognize the most common design patterns used in C++ Understand how to use C++ generic programming to solve common design problems Explore the most powerful C++ idioms, their strengths, and

drawbacks Rediscover how to use popular C++ idioms with generic programming Understand the impact of design patterns on the program's performance Who this book is for This book is for experienced C++ developers and programmers who wish to learn about software design patterns and principles and apply them to create robust, reusable, and easily maintainable apps. [Pro Objective-C Design Patterns for iOS](#) - Carlo Chung 2011-08-28

It's time to capitalize on your mastery of Cocoa with Pro Objective-C Design Patterns for iOS. You've developed apps that impressed and performed, and now you're ready to jump into development practices that will leave you with more effective, efficient, and professional level apps. This book is the element you need to make the jump from journeyman to master. All too often, developers grind through building good apps on willpower and a vigorous focus on code development, leaving them unaware of and unable to benefit from the underlying structural and functional design patterns. Pro Objective-C Design Patterns for iOS will teach you those design patterns that have always been present at some level in your code, but were never recognized, acknowledged, or fully utilized. Implementation of specific pattern approaches will prove their value to any developer working in the iOS application arena. You'll learn to master classic patterns like singleton, abstract factory, chain of responsibility, and observer. You'll also discover less well-known but useful patterns like memento, composite, command, and mediator.

[Design Patterns and Best Practices in Java](#) - Kamalmeet Singh 2018-06-27

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.

[Easy Learning Design Patterns C++ \(1 Edition\)](#) - Yang Hu 2021-08-02

Design Patterns allow you to create more flexible, elegant, and ultimately reusable designs without having to rediscover the design solutions. Design Patterns as your guide, you will learn how these patterns fit into the software development process. All patterns are

compiled from real systems and are based on real-world examples. Each pattern also includes code that demonstrates how it may be implemented in object-oriented programming languages like C++. The book is divided into 2 parts: 1. The first part vividly explains the concept of each design pattern through life 2. The second part applies design patterns to GUI and QT Games examples

Go Design Patterns - Mario Castro Contreras 2017-02-24

Learn idiomatic, efficient, clean, and extensible Go design and concurrency patterns by using TDD About This Book A highly practical guide filled with numerous examples unleashing the power of design patterns with Go. Discover an introduction of the CSP concurrency model by explaining GoRoutines and channels. Get a full explanation, including comprehensive text and examples, of all known GoF design patterns in Go. Who This Book Is For The target audience is both beginner- and advanced-level developers in the Go programming language. No knowledge of design patterns is expected. What You Will Learn All basic syntax and tools needed to start coding in Go Encapsulate the creation of complex objects in an idiomatic way in Go Create unique instances that cannot be duplicated within a program Understand the importance of object encapsulation to provide clarity and maintainability Prepare cost-effective actions so that different parts of the program aren't affected by expensive tasks Deal with channels and GoRoutines within the Go context to build concurrent application in Go in an idiomatic way In Detail Go is a multi-paradigm programming language that has built-in facilities to create concurrent applications. Design patterns allow developers to efficiently address common problems faced during developing applications. Go Design Patterns will provide readers with a reference point to software design patterns and CSP concurrency design patterns to help them build applications in a more idiomatic, robust, and convenient way in Go. The book starts with a brief introduction to Go programming essentials and quickly moves on to explain the idea behind the creation of design patterns and how they appeared in the 90's as a common "language" between developers to solve common tasks in object-

oriented programming languages. You will then learn how to apply the 23 Gang of Four (GoF) design patterns in Go and also learn about CSP concurrency patterns, the "killer feature" in Go that has helped Google develop software to maintain thousands of servers. With all of this the book will enable you to understand and apply design patterns in an idiomatic way that will produce concise, readable, and maintainable software. Style and approach This book will teach widely used design patterns and best practices with Go in a step-by-step manner. The code will have detailed examples, to allow programmers to apply design patterns in their day-to-day coding.

[Hands-On Design Patterns with Delphi](#) - Primož Gabrijelčič 2019-02-27

Get up to speed with creational, structural, behavioral and concurrent patterns in Delphi to write clear, concise and effective code Key Features Delve into the core patterns and components of Delphi in order to master your application's design Brush up on tricks, techniques, and best practices to solve common design and architectural challenges Choose the right patterns to improve your program's efficiency and productivity Book Description Design patterns have proven to be the go-to solution for many common programming scenarios. This book focuses on design patterns applied to the Delphi language. The book will provide you with insights into the language and its capabilities of a runtime library. You'll start by exploring a variety of design patterns and understanding them through real-world examples. This will entail a short explanation of the concept of design patterns and the original set of the 'Gang of Four' patterns, which will help you in structuring your designs efficiently. Next, you'll cover the most important 'anti-patterns' (essentially bad software development practices) to aid you in steering clear of problems during programming. You'll then learn about the eight most important patterns for each creational, structural, and behavioral type. After this, you'll be introduced to the concept of 'concurrency' patterns, which are design patterns specifically related to multithreading and parallel computation. These will enable you to develop and improve an interface between items and harmonize shared memories within

threads. Toward the concluding chapters, you'll explore design patterns specific to program design and other categories of patterns that do not fall under the 'design' umbrella. By the end of this book, you'll be able to address common design problems encountered while developing applications and feel confident while building scalable projects. What you will learn Gain insights into the concept of design patterns Study modern programming techniques with Delphi Keep up to date with the latest additions and program design techniques in Delphi Get to grips with various modern multithreading approaches Discover creational, structural, behavioral, and concurrent patterns Determine how to break a design problem down into its component parts Who this book is for Hands-On Design Patterns with Delphi is aimed at beginner-level Delphi developers who want to build scalable and robust applications. Basic knowledge of Delphi is a must.

Learning JavaScript Design Patterns - Addy Osmani 2012-07-08

With Learning JavaScript Design Patterns, you'll learn how to write beautiful, structured, and maintainable JavaScript by applying classical and modern design patterns to the language. If you want to keep your code efficient, more manageable, and up-to-date with the latest best practices, this book is for you. Explore many popular design patterns, including Modules, Observers, Facades, and Mediators. Learn how modern architectural patterns—such as MVC, MVP, and MVVM—are useful from the perspective of a modern web application developer. This book also walks experienced JavaScript developers through modern module formats, how to namespace code effectively, and other essential topics. Learn the structure of design patterns and how they are written Understand different pattern categories, including creational, structural, and behavioral Walk through more than 20 classical and modern design patterns in JavaScript Use several options for writing modular code—including the Module pattern, Asynchronous Module Definition (AMD), and CommonJS Discover design patterns implemented in the jQuery library Learn popular design patterns for writing maintainable jQuery plug-ins "This book should be in every JavaScript

developer's hands. It's the go-to book on JavaScript patterns that will be read and referenced many times in the future."—Andrée Hansson, Lead Front-End Developer, presis!

Design Patterns in Modern C++ - Dmitri Nesteruk 2018-04-18

Apply modern C++17 to the implementations of classic design patterns. As well as covering traditional design patterns, this book fleshes out new patterns and approaches that will be useful to C++ developers. The author presents concepts as a fun investigation of how problems can be solved in different ways, along the way using varying degrees of technical sophistication and explaining different sorts of trade-offs.

Design Patterns in Modern C++ also provides a technology demo for modern C++, showcasing how some of its latest features (e.g., coroutines) make difficult problems a lot easier to solve. The examples in this book are all suitable for putting into production, with only a few simplifications made in order to aid readability. What You Will Learn Apply design patterns to modern C++ programming Use creational patterns of builder, factories, prototype and singleton Implement structural patterns such as adapter, bridge, decorator, facade and more Work with the behavioral patterns such as chain of responsibility, command, iterator, mediator and more Apply functional design patterns such as Monad and more Who This Book Is For Those with at least some prior programming experience, especially in C++.

Hands-On Design Patterns with Kotlin - Alexey Soshin 2018-06-15

Make the most of Kotlin by leveraging design patterns and best practices to build scalable and high performing apps Key Features Understand traditional GOF design patterns to apply generic solutions Shift from OOP to FP; covering reactive and concurrent patterns in a step-by-step manner Choose the best microservices architecture and MVC for your development environment Book Description Design patterns enable you as a developer to speed up the development process by providing you with proven development paradigms. Reusing design patterns helps prevent complex issues that can cause major problems, improves your code base, promotes code reuse, and makes an architecture more robust. The mission of this book is to ease

the adoption of design patterns in Kotlin and provide good practices for programmers. The book begins by showing you the practical aspects of smarter coding in Kotlin, explaining the basic Kotlin syntax and the impact of design patterns. From there, the book provides an in-depth explanation of the classical design patterns of creational, structural, and behavioral families, before heading into functional programming. It then takes you through reactive and concurrent patterns, teaching you about using streams, threads, and coroutines to write better code along the way 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 Get to grips with Kotlin principles, including its strengths and weaknesses Understand classical design patterns in Kotlin Explore functional programming using built-in features of Kotlin Solve real-world problems using reactive and concurrent design patterns Use threads and coroutines to simplify concurrent code flow Understand antipatterns to write clean Kotlin code, avoiding common pitfalls Learn about the design considerations necessary while choosing between architectures Who this book is for This book is for developers who would like to master design patterns with Kotlin to build efficient and scalable applications. Basic Java or Kotlin programming knowledge is assumed

Head First Design Patterns - Eric Freeman 2004-10-25

Using research in neurobiology, cognitive science and learning theory, this text loads patterns into your brain in a way that lets you put them to work immediately, makes you better at solving software design problems, and improves your ability to speak the language of patterns with others on your team.

C++ High Performance - Björn Andrist 2020-12-30

A comprehensive guide to help aspiring and professional C++ developers elevate the performance of their apps by allowing them to run faster and consume fewer resources Key Features Updated to C++20 with completely revised code and more content on error handling, benchmarking, memory allocators, and

concurrent programmingExplore the latest C++20 features including concepts, ranges, and coroutinesUtilize C++ constructs and techniques to carry out effective data structure optimization and memory managementBook Description C++ High Performance, Second Edition guides you through optimizing the performance of your C++ apps. This allows them to run faster and consume fewer resources on the device they're running on without compromising the readability of your codebase. The book begins by introducing the C++ language and some of its modern concepts in brief. Once you are familiar with the fundamentals, you will be ready to measure, identify, and eradicate bottlenecks in your C++ codebase. By following this process, you will gradually improve your style of writing code. The book then explores data structure optimization, memory management, and how it can be used efficiently concerning CPU caches. After laying the foundation, the book trains you to leverage algorithms, ranges, and containers from the standard library to achieve faster execution, write readable code, and use customized iterators. It provides hands-on examples of C++ metaprogramming, coroutines, reflection to reduce boilerplate code, proxy objects to perform optimizations under the hood, concurrent programming, and lock-free data structures. The book concludes with an overview of parallel algorithms. By the end of this book, you will have the ability to use every tool as needed to boost the efficiency of your C++ projects. What you will learnWrite specialized data structures for performance-critical codeUse modern metaprogramming techniques to reduce runtime calculationsAchieve efficient memory management using custom memory allocatorsReduce boilerplate code using reflection techniquesReap the benefits of lock-free concurrent programmingGain insights into subtle optimizations used by standard library algorithmsCompose algorithms using ranges libraryDevelop the ability to apply metaprogramming aspects such as constexpr, constraints, and conceptsImplement lazy generators and asynchronous tasks using C++20 coroutinesWho this book is for If you're a C++ developer looking to improve the efficiency of your code or just keen to upgrade your skills

to the next level, this book is for you.

Hands-On Design Patterns and Best

Practices with Julia - Tom Kwong 2020-01-17

Design and develop high-performance, reusable, and maintainable applications using traditional and modern Julia patterns with this comprehensive guide Key FeaturesExplore useful design patterns along with object-oriented programming in Julia 1.0Implement macros and metaprogramming techniques to make your code faster, concise, and efficientDevelop the skills necessary to implement design patterns for creating robust and maintainable applicationsBook Description Design patterns are fundamental techniques for developing reusable and maintainable code. They provide a set of proven solutions that allow developers to solve problems in software development quickly. This book will demonstrate how to leverage design patterns with real-world applications. Starting with an overview of design patterns and best practices in application design, you'll learn about some of the most fundamental Julia features such as modules, data types, functions/interfaces, and metaprogramming. You'll then get to grips with the modern Julia design patterns for building large-scale applications with a focus on performance, reusability, robustness, and maintainability. The book also covers anti-patterns and how to avoid common mistakes and pitfalls in development. You'll see how traditional object-oriented patterns can be implemented differently and more effectively in Julia. Finally, you'll explore various use cases and examples, such as how expert Julia developers use design patterns in their open source packages. By the end of this Julia programming book, you'll have learned methods to improve software design, extensibility, and reusability, and be able to use design patterns efficiently to overcome common challenges in software development. What you will learnMaster the Julia language features that are key to developing large-scale software applicationsDiscover design patterns to improve overall application architecture and designDevelop reusable programs that are modular, extendable, performant, and easy to maintainWeigh up the pros and cons of using different design patterns for use casesExplore methods for transitioning from object-oriented

programming to using equivalent or more advanced Julia techniques Who this book is for This book is for beginner to intermediate-level Julia programmers who want to enhance their skills in designing and developing large-scale applications.

Search Patterns - Peter Morville 2010-01-14

What people are saying about Search Patterns

"Search Patterns is a delight to read -- very thoughtful and thought provoking. It's the most comprehensive survey of designing effective search experiences I've seen." --Irene Au,

Director of User Experience, Google "I love this book! Thanks to Peter and Jeffery, I now know

that search (yes, boring old yucky who cares search) is one of the coolest ways around of looking at the world." --Dan Roam, author, *The Back of the Napkin* (Portfolio Hardcover)

"Search Patterns is a playful guide to the practical concerns of search interface design. It contains a bonanza of screenshots and

illustrations that capture the best of today's design practices and presents a fresh

perspective on the broader role of search and discovery." --Marti Hearst, Professor, UC

Berkeley and author, *Search User Interfaces* (Cambridge University Press) "It's not often I

come across a book that asks profound questions about a fundamental human activity, and then

proceeds to answer those questions with practical observations and suggestions. Search

Patterns is an expedition into the heart of the web and human cognition, and for me it was a

delightful journey that delivered scores of insights." --Dave Gray, Founder and Chairman,

XPLANE "Search is swiftly transforming everything we know, yet people don't

understand how mavens design search: by stacking breadcrumbs, scenting widgets, and

keeping eyeballs on the engine. I urge you to put your eyeballs on this unique and important

book." --Bruce Sterling, Writer, Futurist, and Co-Founder, The Electronic Frontier Foundation "As

one who searches a lot (and often ends up frustrated), Search Patterns is a revelation." --

Nigel Holmes, Designer, Theorist, and Principal, Explanation Graphics "Search Patterns is a

fabulous must-have book! Inside, you'll learn the whys and wheres of practically every modern

search design trick and technique." --Jared Spool, CEO and Founder, User Interface

Engineering Search is among the most disruptive innovations of our time. It influences what we buy and where we go. It shapes how we

learn and what we believe. In this provocative and inspiring book, you'll explore design

patterns that apply across the categories of web, ecommerce, enterprise, desktop, mobile, social,

and real-time search and discovery. Filled with colorful illustrations and examples, Search

Patterns brings modern information retrieval to life, covering such diverse topics as relevance,

faceted navigation, multi-touch, personalization, visualization, multi-sensory search, and

augmented reality. By drawing on their own experience-as well as best practices and

evidence-based research-the authors not only offer a practical guide to help you build effective

search applications, they also challenge you to imagine the future of discovery. You'll find

Search Patterns intriguing and invaluable, whether you're a web practitioner, mobile

designer, search entrepreneur, or just interested in the topic. Discover a pattern language for

search that embraces user psychology and behavior, information architecture, interaction

design, and emerging technology Boost enterprise efficiency and e-commerce sales

Enable mobile users to achieve goals, complete tasks, and find what they need Drive design

innovation for search interfaces and applications

Game Programming Patterns - Robert Nystrom 2014-11-03

The biggest challenge facing many game programmers is completing their game. Most

game projects fizzle out, overwhelmed by the complexity of their own code. Game

Programming Patterns tackles that exact problem. Based on years of experience in

shipped AAA titles, this book collects proven patterns to untangle and optimize your game,

organized as independent recipes so you can pick just the patterns you need. You will learn

how to write a robust game loop, how to organize your entities using components, and

take advantage of the CPUs cache to improve your performance. You'll dive deep into how

scripting engines encode behavior, how quadrees and other spatial partitions optimize

your engine, and how other classic design patterns can be used in games.

C# 3.0 Design Patterns - Judith Bishop

Downloaded from
omahafoodtruckassociation.org on by
guest

2007-12-10

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 foundational 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.

[Hands-On System Programming with C++](#) - Dr. Rian Quinn 2018-12-26

A hands-on guide to making system programming with C++ easy Key FeaturesWrite system-level code leveraging C++17Learn the internals of the Linux Application Binary Interface (ABI) and apply it to system programmingExplore C++ concurrency to take advantage of server-level constructsBook Description C++ is a general-purpose programming language with a bias toward system programming as it provides ready access to hardware-level resources, efficient compilation, and a versatile approach to higher-level abstractions. This book will help you understand the benefits of system programming

with C++17. You will gain a firm understanding of various C, C++, and POSIX standards, as well as their respective system types for both C++ and POSIX. After a brief refresher on C++, Resource Acquisition Is Initialization (RAII), and the new C++ Guideline Support Library (GSL), you will learn to program Linux and Unix systems along with process management. As you progress through the chapters, you will become acquainted with C++'s support for IO. You will then study various memory management methods, including a chapter on allocators and how they benefit system programming. You will also explore how to program file input and output and learn about POSIX sockets. This book will help you get to grips with safely setting up a UDP and TCP server/client. Finally, you will be guided through Unix time interfaces, multithreading, and error handling with C++ exceptions. By the end of this book, you will be comfortable with using C++ to program high-quality systems. What you will learnUnderstand the benefits of using C++ for system programmingProgram Linux/Unix systems using C++Discover the advantages of Resource Acquisition Is Initialization (RAII)Program both console and file input and outputUncover the POSIX socket APIs and understand how to program themExplore advanced system programming topics, such as C++ allocatorsUse POSIX and C++ threads to program concurrent systemsGrasp how C++ can be used to create performant system applicationsWho this book is for If you are a fresh developer with intermediate knowledge of C++ but little or no knowledge of Unix and Linux system programming, this book will help you learn system programming with C++ in a practical way.

C# Design Pattern Essentials - Tony Bevis 2012

Once you've learned the fundamentals of C#, understanding Design Patterns is essential for writing clear, concise and effective code. This book gives you a step-by-step guide to object-oriented development, using tried and trusted techniques. The examples have been kept simple, enabling you to concentrate on understanding the concepts and application of each pattern. All examples have been designed around a common theme, making it easier to see

how they relate to each other and how you can adapt them to your applications. While the book assumes a basic knowledge of C# you don't need to be a guru. This book is perfect for the programmer wishing to take their skills to the next level, and feel confident about using C# in real applications. Coverage includes all 23 of the patterns from the "Gang of Four" work, additional patterns including Model-View-Controller, and simple UML diagrams.

Object-Oriented Design And Patterns - Cay Horstmann 2009-08

Cay Horstmann offers readers an effective means for mastering computing concepts and developing strong design skills. This book introduces object-oriented fundamentals critical to designing software and shows how to implement design techniques. The author's clear, hands-on presentation and outstanding writing style help readers to better understand the material. · A Crash Course in Java · The Object-Oriented Design Process · Guidelines for Class Design · Interface Types and Polymorphism · Patterns and GUI Programming · Inheritance and Abstract Classes · The Java Object Model · Frameworks · Multithreading · More Design Patterns

Mastering Python Design Patterns - Kamon Ayeva 2018-08-31

Exploit various design patterns to master the art of solving problems using Python Key Features Master the application design using the core design patterns and latest features of Python 3.7 Learn tricks to solve common design and architectural challenges Choose the right plan to improve your programs and increase their productivity Book Description Python is an object-oriented scripting language that is used in a wide range of categories. In software engineering, a design pattern is an elected solution for solving software design problems. Although they have been around for a while, design patterns remain one of the top topics in software engineering, and are a ready source for software developers to solve the problems they face on a regular basis. This book takes you through a variety of design patterns and explains them with real-world examples. You will get to grips with low-level details and concepts that show you how to write Python code, without focusing on common solutions as enabled in Java

and C++. You'll also find sections on corrections, best practices, system architecture, and its designing aspects. This book will help you learn the core concepts of design patterns and the way they can be used to resolve software design problems. You'll focus on most of the Gang of Four (GoF) design patterns, which are used to solve everyday problems, and take your skills to the next level with reactive and functional patterns that help you build resilient, scalable, and robust applications. By the end of the book, you'll be able to efficiently address commonly faced problems and develop applications, and also be comfortable working on scalable and maintainable projects of any size. What you will learn Explore Factory Method and Abstract Factory for object creation Clone objects using the Prototype pattern Make incompatible interfaces compatible using the Adapter pattern Secure an interface using the Proxy pattern Choose an algorithm dynamically using the Strategy pattern Keep the logic decoupled from the UI using the MVC pattern Leverage the Observer pattern to understand reactive programming Explore patterns for cloud-native, microservices, and serverless architectures Who this book is for This book is for intermediate Python developers. Prior knowledge of design patterns is not required to enjoy this book.

Design Patterns in Ruby (Adobe Reader) - Russ Olsen 2007-12-10

Praise for Design Patterns in Ruby " Design Patterns in Ruby documents smart ways to resolve many problems that Ruby developers commonly encounter. Russ Olsen has done a great job of selecting classic patterns and augmenting these with newer patterns that have special relevance for Ruby. He clearly explains each idea, making a wealth of experience available to Ruby developers for their own daily work." —Steve Metsker, Managing Consultant with Dominion Digital, Inc. "This book provides a great demonstration of the key 'Gang of Four' design patterns without resorting to overly technical explanations. Written in a precise, yet almost informal style, this book covers enough ground that even those without prior exposure to design patterns will soon feel confident applying them using Ruby. Olsen has done a great job to make a book about a classically 'dry' subject into such an engaging and even

occasionally humorous read." —Peter Cooper
"This book renewed my interest in understanding patterns after a decade of good intentions. Russ picked the most useful patterns for Ruby and introduced them in a straightforward and logical manner, going beyond the GoF's patterns. This book has improved my use of Ruby, and encouraged me to blow off the dust covering the GoF book." —Mike Stok
"Design Patterns in Ruby is a great way for programmers from statically typed object-oriented languages to learn how design patterns appear in a more dynamic, flexible language like Ruby." —Rob Sanheim, Ruby Ninja, Relevance
Most design pattern books are based on C++ and Java. But Ruby is different—and the language's unique qualities make design patterns easier to implement and use. In this book, Russ Olsen demonstrates how to combine Ruby's power and elegance with patterns, and write more sophisticated, effective software with far fewer lines of code. After reviewing the history, concepts, and goals of design patterns, Olsen offers a quick tour of the Ruby language—enough to allow any experienced software developer to immediately utilize patterns with Ruby. The book especially calls attention to Ruby features that simplify the use of patterns, including dynamic typing, code closures, and "mixins" for easier code reuse. Fourteen of the classic "Gang of Four" patterns are considered from the Ruby point of view, explaining what problems each pattern solves, discussing whether traditional implementations make sense in the Ruby environment, and introducing Ruby-specific improvements. You'll discover opportunities to implement patterns in just one or two lines of code, instead of the endlessly repeated boilerplate that conventional languages often require. Design Patterns in Ruby also identifies innovative new patterns that have emerged from the Ruby community. These include ways to create custom objects with metaprogramming, as well as the ambitious Rails-based "Convention Over Configuration" pattern, designed to help integrate entire applications and frameworks. Engaging, practical, and accessible, Design Patterns in Ruby will help you build better software while making your Ruby programming experience more rewarding.

Reactive Design Patterns - Jamie Allen

2017-02-21

Summary Reactive Design Patterns is a clearly written guide for building message-driven distributed systems that are resilient, responsive, and elastic. In this book you'll find patterns for messaging, flow control, resource management, and concurrency, along with practical issues like test-friendly designs. All patterns include concrete examples using Scala and Akka. Foreword by Jonas Bonér. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Modern web applications serve potentially vast numbers of users - and they need to keep working as servers fail and new ones come online, users overwhelm limited resources, and information is distributed globally. A Reactive application adjusts to partial failures and varying loads, remaining responsive in an ever-changing distributed environment. The secret is message-driven architecture - and design patterns to organize it. About the Book Reactive Design Patterns presents the principles, patterns, and best practices of Reactive application design. You'll learn how to keep one slow component from bogging down others with the Circuit Breaker pattern, how to shepherd a many-staged transaction to completion with the Saga pattern, how to divide datasets by Sharding, and more. You'll even see how to keep your source code readable and the system testable despite many potential interactions and points of failure. What's Inside The definitive guide to the Reactive Manifesto Patterns for flow control, delimited consistency, fault tolerance, and much more Hard-won lessons about what doesn't work Architectures that scale under tremendous load About the Reader Most examples use Scala, Java, and Akka. Readers should be familiar with distributed systems. About the Author Dr. Roland Kuhn led the Akka team at Lightbend and coauthored the Reactive Manifesto. Brian Hanafée and Jamie Allen are experienced distributed systems architects. Table of Contents PART 1 - INTRODUCTION Why Reactive? A walk-through of the Reactive Manifesto Tools of the trade PART 2 - THE PHILOSOPHY IN A NUTSHELL Message passing Location transparency Divide and conquer Principled

failure handling Delimited consistency
Nondeterminism by need Message flow PART 3 -
PATTERNS Testing reactive applications Fault
tolerance and recovery patterns Replication
patterns Resource-management patterns
Message flow patterns Flow control patterns
State management and persistence patterns

Holub on Patterns - Allen Holub 2004-09-27

* Allen Holub is a highly regarded instructor for the University of California, Berkeley, Extension. He has taught since 1982 on various topics, including Object-Oriented Analysis and Design, Java, C++, C. Holub will use this book in his Berkeley Extension classes. * Holub is a regular presenter at the Software Development conferences and is Contributing Editor for the online magazine JavaWorld, for whom he writes the Java Toolbox. He also wrote the OO Design Process column for IBM DeveloperWorks. * This book is not time-sensitive. It is an extremely well-thought out approach to learning design patterns, with Java as the example platform, but the concepts presented are not limited to just Java programmers. This is a complement to the Addison-Wesley seminal "Design Patterns" book by the "Gang of Four".

Software Architecture with C++ - Adrian Ostrowski 2021-04-23

Apply business requirements to IT infrastructure and deliver a high-quality product by understanding architectures such as microservices, DevOps, and cloud-native using modern C++ standards and features Key Features Design scalable large-scale applications with the C++ programming language Architect software solutions in a cloud-based environment with continuous integration and continuous delivery (CI/CD) Achieve architectural goals by leveraging design patterns, language features, and useful tools Book Description Software architecture refers to the high-level design of complex applications. It is evolving just like the languages we use, but there are architectural concepts and patterns that you can learn to write high-performance apps in a high-level language without sacrificing readability and maintainability. If you're working with modern C++, this practical guide will help you put your knowledge to work and design distributed, large-scale apps. You'll start by getting up to speed with architectural concepts, including

established patterns and rising trends, then move on to understanding what software architecture actually is and start exploring its components. Next, you'll discover the design concepts involved in application architecture and the patterns in software development, before going on to learn how to build, package, integrate, and deploy your components. In the concluding chapters, you'll explore different architectural qualities, such as maintainability, reusability, testability, performance, scalability, and security. Finally, you will get an overview of distributed systems, such as service-oriented architecture, microservices, and cloud-native, and understand how to apply them in application development. By the end of this book, you'll be able to build distributed services using modern C++ and associated tools to deliver solutions as per your clients' requirements. What you will learn Understand how to apply the principles of software architecture Apply design patterns and best practices to meet your architectural goals Write elegant, safe, and performant code using the latest C++ features Build applications that are easy to maintain and deploy Explore the different architectural approaches and learn to apply them as per your requirement Simplify development and operations using application containers Discover various techniques to solve common problems in software design and development Who this book is for This software architecture C++ programming book is for experienced C++ developers looking to become software architects or develop enterprise-grade applications.

Hands-On Design Patterns with C# and .NET Core - Gaurav Arora 2019-07-05

Apply design patterns to solve problems in software architecture and programming using C# 7.x and .NET Core 2 Key Features Enhance your programming skills by implementing efficient design patterns for C# and .NET Explore design patterns for functional and reactive programming to build robust and scalable applications Discover how to work effectively with microservice and serverless architectures Book Description Design patterns are essentially reusable solutions to common programming problems. When used correctly, they meet crucial software requirements with ease and reduce costs. This book will uncover

effective ways to use design patterns and demonstrate their implementation with executable code specific to both C# and .NET Core. Hands-On Design Patterns with C# and .NET Core begins with an overview of object-oriented programming (OOP) and SOLID principles. It provides an in-depth explanation of the Gang of Four (GoF) design patterns such as creational, structural, and behavioral. The book then takes you through functional, reactive, and concurrent patterns, helping you write better code with streams, threads, and coroutines. Toward the end of the book, you'll learn about the latest trends in architecture, exploring design patterns for microservices, serverless, and cloud native applications. You'll even understand the considerations that need to be taken into account when choosing between different architectures such as microservices and MVC. By the end of the book, you will be able to write efficient and clear code and be comfortable working on scalable and maintainable projects of any size. What you will learn

Make your code more flexible by applying SOLID principles
Follow the Test-driven development (TDD) approach in your .NET Core projects
Get to grips with efficient database migration, data persistence, and testing techniques
Convert a console application to a web application using the right MVP
Write asynchronous, multithreaded, and parallel code
Implement MVVM and work with RxJS and AngularJS to deal with changes in databases
Explore the features of microservices, serverless programming, and cloud computing

Who this book is for
If you have a basic understanding of C# and the .NET Core framework, this book will help you write code that is easy to reuse and maintain with the help of proven design patterns that you can implement in your code.

Introduction to Design Patterns in C++ with Qt - Alan Ezust 2011-08-29

Master C++ "The Qt Way" with Modern Design Patterns and Efficient Reuse This fully updated, classroom-tested book teaches C++ "The Qt Way," emphasizing design patterns and efficient reuse. Readers will master both the C++ language and Qt libraries, as they learn to develop maintainable software with well-defined code layers and simple, reusable classes and

functions. Every chapter of this edition has been improved with new content, better organization, or both. Readers will find extensively revised coverage of QObjects, Reflection, Widgets, Main Windows, Models and Views, Databases, Multi-Threaded Programming, and Reflection. This edition introduces the powerful new Qt Creator IDE; presents new multimedia APIs; and offers extended coverage of Qt Designer and C++ Integration. It has been restructured to help readers start writing software immediately and write robust, effective software sooner. The authors introduce several new design patterns, add many quiz questions and labs, and present more efficient solutions relying on new Qt features and best practices. They also provide an up-to-date C++ reference section and a complete application case study. Master C++ keywords, literals, identifiers, declarations, types, and type conversions. Understand classes and objects, organize them, and describe their interrelationships. Learn consistent programming style and naming rules. Use lists, functions, and other essential techniques. Define inheritance relationships to share code and promote reuse. Learn how code libraries are designed, built, and reused. Work with QObject, the base class underlying much of Qt. Build graphical user interfaces with Qt widgets. Use templates to write generic functions and classes. Master advanced reflective programming techniques. Use the Model-View framework to cleanly separate data and GUI classes. Validate input using regular expressions and other techniques. Parse XML data with SAX, DOM, and QDomStreamReader. Master today's most valuable creational and structural design patterns. Create, use, monitor, and debug processes and threads. Access databases with Qt's SQL classes. Manage memory reliably and efficiently. Understand how to effectively manage QThreads and use QtConcurrent algorithms. Click here to obtain supplementary materials for this book.

Hands-On Design Patterns with Java - Dr. Edward Lavieri 2019-04-27

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 codeExplore architectural patterns and practices for building scalable and reliable applicationsTackle all kinds of performance-related issues and streamline development using design patternsBook 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 serverless architectures by using cloud design patterns, 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 learnUnderstand the significance of design patterns for software engineeringVisualize software design with UML diagramsStrengthen your understanding of OOP to create reusable software systemsDiscover GOF design patterns to develop scalable applicationsExamine programming challenges and the design patterns that solve themExplore architectural patterns for microservices and cloud developmentWho 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.

The Rust Programming Language (Covers Rust 2018) - Steve Klabnik 2019-09-03

The official book on the Rust programming language, written by the Rust development team at the Mozilla Foundation, fully updated for Rust

2018. The Rust Programming Language is the official book on Rust: an open source systems programming language that helps you write faster, more reliable software. Rust offers control over low-level details (such as memory usage) in combination with high-level ergonomics, eliminating the hassle traditionally associated with low-level languages. The authors of The Rust Programming Language, members of the Rust Core Team, share their knowledge and experience to show you how to take full advantage of Rust's features--from installation to creating robust and scalable programs. You'll begin with basics like creating functions, choosing data types, and binding variables and then move on to more advanced concepts, such as:

- Ownership and borrowing, lifetimes, and traits
- Using Rust's memory safety guarantees to build fast, safe programs
- Testing, error handling, and effective refactoring
- Generics, smart pointers, multithreading, trait objects, and advanced pattern matching
- Using Cargo, Rust's built-in package manager, to build, test, and document your code and manage dependencies
- How best to use Rust's advanced compiler with compiler-led programming techniques

You'll find plenty of code examples throughout the book, as well as three chapters dedicated to building complete projects to test your learning: a number guessing game, a Rust implementation of a command line tool, and a multithreaded server. New to this edition: An extended section on Rust macros, an expanded chapter on modules, and appendixes on Rust development tools and editions.

[Hands-On Design Patterns with React Native](#) - Mateusz Grzesiukiewicz 2018-09-29

Learn how to write cross platform React Native code by using effective design patterns in the JavaScript world. Get to know industry standard patterns as well as situational patterns.

Decouple your application with these set of "Idea patterns". Key FeaturesMobile development in React Native should be done in a reusable way. Learn how to build scalable applications using JavaScript patterns that are battle tested.Try effective techniques on your own using over 80 standalone examples.Book Description React Native helps developers reuse code across different mobile platforms like iOS and Android. This book will show you effective

design patterns in the React Native world and will make you ready for professional development in big teams. The book will focus only on the patterns that are relevant to JavaScript, ECMAScript, React and React Native. However, you can successfully transfer a lot of the skills and techniques to other languages. I call them "Idea patterns". This book will start with the most standard development patterns in React like component building patterns, styling patterns in React Native and then extend these patterns to your mobile application using real world practical examples. Each chapter comes with full, separate source code of applications that you can build and run on your phone. The book is also diving into architectural patterns. Especially how to adapt MVC to React environment. You will learn Flux architecture and how Redux is implementing it. Each approach will be presented with its pros and cons. You will learn how to work with external data sources using libraries like Redux thunk and Redux Saga. The end goal is the ability to recognize the best solution for a given problem for your next mobile application. What you will learn

- Explore the design Patterns in React Native
- Learn the best practices for React Native development
- Explore common React patterns that are highly used within React Native development
- Learn to decouple components and use dependency injection in your applications
- Explore the best ways of fetching data from the backend systems
- Learn the styling patterns and how to implement custom mobile designs
- Explore the best ways to organize your application code in big codebases

Who this book is for The ideal target audience for this book are people eager to learn React Native design patterns who already know the basics of JavaScript. We can assume that the target audience already knows how to write Hello World in JavaScript and know what are the functions, recursive functions, JavaScript types and loops.

Java Design Patterns - Vaskaran Sarcar
2018-12-06

Get hands-on experience implementing 26 of the most common design patterns using Java and Eclipse. In addition to Gang of Four (GoF) design patterns, you will also learn about alternative design patterns, and understand the criticisms of

design patterns with an overview of anti-patterns. For each pattern you will see at least one real-world scenario, a computer-world example, and a complete implementation including output. This book has three parts. The first part covers 23 Gang of Four (GoF) design patterns. The second part includes three alternative design patterns. The third part presents criticisms of design patterns with an overview of anti-patterns. You will work through easy-to-follow examples to understand the concepts in depth and you will have a collection of programs to port over to your own projects. A Q&A session is included in each chapter and covers the pros and cons of each pattern. The last chapter presents FAQs about the design patterns. The step-by-step approach of the book helps you apply your skills to learn other patterns on your own, and to be familiar with the latest version of Java and Eclipse. What You'll Learn

- Work with each of the design patterns
- Implement design patterns in real-world applications
- Choose from alternative design patterns by comparing their pros and cons
- Use the Eclipse IDE to write code and generate output
- Read the in-depth Q&A session in each chapter with pros and cons for each design pattern

Who This Book Is For Software developers, architects, and programmers

C++ Programming with Design Patterns Revealed - Tomasz Muldner 2002

C++ Programming with Design Patterns Revealed introduces C++ syntax alongside current object-oriented tools such as design patterns, and the Unified Modeling Language (UML), which are essential for the production of well-designed C++ software. Through this book, readers will attain mastery of many C++ features, as well as the object-oriented design techniques that facilitate and optimize their use. This book uses an example-based approach. First, a technique is presented alongside a piece of code that implements that technique. Next, a component is shown that uses the technique. Finally, an entire running example that incorporates the technique is presented. The book balances a systematic discussion of object-oriented design alongside the introduction of C++ syntax. It introduces twelve basic design patterns early on and uses them throughout, and describes design patterns via use of basic UML.

Downloaded from
omahafoodtruckassociation.org on by
guest

Numerous reference appendices are included for the idioms, design patterns, and programming guidelines in the book. Portability tips, common programming errors, idioms, and programming style tips are also highlighted in each chapter. This book is designed for readers who have been exposed to Java, as well as to basic object-oriented ideas, and are looking to gain familiarity with C++.

The Art of Writing Efficient Programs -

Fedor G. Pikus 2021-10-22

Become a better programmer with performance improvement techniques such as concurrency, lock-free programming, atomic operations, parallelism, and memory management

Key Features Learn proven techniques from a heavyweight and recognized expert in C++ and high-performance computing Understand the limitations of modern CPUs and their performance impact Find out how you can avoid writing inefficient code and get the best optimizations from the compiler Learn the tradeoffs and costs of writing high-performance programs

Book Description The great free lunch of "performance taking care of itself" is over. Until recently, programs got faster by themselves as CPUs were upgraded, but that doesn't happen anymore. The clock frequency of new processors has almost peaked, and while new architectures provide small improvements to existing programs, this only helps slightly. To write efficient software, you now have to know how to program by making good use of the available computing resources, and this book will teach you how to do that. The Art of Efficient Programming covers all the major aspects of writing efficient programs, such as using CPU resources and memory efficiently, avoiding unnecessary computations, measuring performance, and how to put concurrency and multithreading to good use. You'll also learn about compiler optimizations and how to use the programming language (C++) more efficiently. Finally, you'll understand how design decisions impact performance. By the end of this book, you'll not only have enough knowledge of processors and compilers to write efficient programs, but you'll also be able to understand which techniques to use and what to measure while improving performance. At its core, this book is about learning how to learn. What you

will learn Discover how to use the hardware computing resources in your programs effectively Understand the relationship between memory order and memory barriers Familiarize yourself with the performance implications of different data structures and organizations Assess the performance impact of concurrent memory accessed and how to minimize it Discover when to use and when not to use lock-free programming techniques Explore different ways to improve the effectiveness of compiler optimizations Design APIs for concurrent data structures and high-performance data structures to avoid inefficiencies

Who this book is for This book is for experienced developers and programmers who work on performance-critical projects and want to learn new techniques to improve the performance of their code. Programmers in algorithmic trading, gaming, bioinformatics, computational genomics, or computational fluid dynamics communities will get the most out of the examples in this book, but the techniques are fairly universal. Although this book uses the C++ language, the concepts demonstrated in the book can be easily transferred or applied to other compiled languages such as C, Java, Rust, Go, and more.

Hands-On Design Patterns with Swift -

Florent Vilmart 2018-12-24

From learning about the most sought-after design patterns to a comprehensive coverage of architectural patterns and code testing, this book is all you need to write clean, reusable code

Key Features Write clean, reusable and maintainable code, and make the most of the latest Swift version. Analyze case studies of some of the popular open source projects and give your workflow a huge boost Choose patterns such as MVP, MVC, and MVVM depending on the application being built

Book Description Swift keeps gaining traction not only amongst Apple developers but also as a server-side language. This book demonstrates how to apply design patterns and best practices in real-life situations, whether that's for new or already existing projects. You'll begin with a quick refresher on Swift, the compiler, the standard library, and the foundation, followed by the Cocoa design patterns - the ones at the core of many cocoa libraries - to follow up with the

creational, structural, and behavioral patterns as defined by the GoF. You'll get acquainted with application architecture, as well as the most popular architectural design patterns, such as MVC and MVVM, and learn to use them in the context of Swift. In addition, you'll walk through dependency injection and functional reactive programming. Special emphasis will be given to techniques to handle concurrency, including callbacks, futures and promises, and reactive programming. These techniques will help you adopt a test-driven approach to your workflow in order to use Swift Package Manager and integrate the framework into the original code base, along with Unit and UI testing. By the end of the book, you'll be able to build applications that are scalable, faster, and easier to maintain. What you will learn Work efficiently with Foundation and Swift Standard library Understand the most critical GoF patterns and use them efficiently Use Swift 4.2 and its unique capabilities (and limitations) to implement and improve GoF patterns Improve your application architecture and optimize for maintainability and performance Write efficient and clean concurrent programs using futures and promises, or reactive programming techniques Use Swift Package Manager to refactor your program into reusable components Leverage testing and other techniques for writing robust code Who this book is for This book is for intermediate developers who want to apply design patterns with Swift to structure and scale their applications. You are expected to have basic knowledge of iOS and Swift.

Design Patterns in .NET - Dmitri Nesteruk
2019-05-11

Implement design patterns in .NET using the latest versions of the C# and F# languages. This book provides a comprehensive overview of the field of design patterns as they are used in today's developer toolbox. Using the C# programming language, Design Patterns in .NET explores the classic design pattern implementation and discusses the applicability and relevance of specific language features for the purpose of implementing patterns. You will learn by example, reviewing scenarios where patterns are applicable. MVP and patterns expert Dmitri Nesteruk demonstrates possible implementations of patterns, discusses

alternatives and pattern inter-relationships, and illustrates the way that a dedicated refactoring tool (ReSharper) can be used to implement design patterns with ease. What You'll Learn Know the latest pattern implementations available in C# and F# Refer to researched and proven variations of patterns Study complete, self-contained examples including many that cover advanced scenarios Use the latest implementations of C# and Visual Studio/ReSharper Who This Book Is For Developers who have some experience in the C# language and want to expand their comprehension of the art of programming by leveraging design approaches to solving modern problems

Hands-On Parallel Programming with C# 8 and .NET Core 3 - Shakti Tanwar 2019-12-20

Enhance your enterprise application development skills by mastering parallel programming techniques in .NET and C# Key FeaturesWrite efficient, fine-grained, and scalable parallel code with C# and .NET CoreExperience how parallel programming works by building a powerful applicationLearn the fundamentals of multithreading by working with IIS and KestrelBook Description In today's world, every CPU has a multi-core processor. However, unless your application has implemented parallel programming, it will fail to utilize the hardware's full processing capacity. This book will show you how to write modern software on the optimized and high-performing .NET Core 3 framework using C# 8. Hands-On Parallel Programming with C# 8 and .NET Core 3 covers how to build multithreaded, concurrent, and optimized applications that harness the power of multi-core processors. Once you've understood the fundamentals of threading and concurrency, you'll gain insights into the data structure in .NET Core that supports parallelism. The book will then help you perform asynchronous programming in C# and diagnose and debug parallel code effectively. You'll also get to grips with the new Kestrel server and understand the difference between the IIS and Kestrel operating models. Finally, you'll learn best practices such as test-driven development, and run unit tests on your parallel code. By the end of the book, you'll have developed a deep understanding of the core concepts of

concurrency and asynchrony to create responsive applications that are not CPU-intensive. What you will learn

Analyze and break down a problem statement for parallelism

Explore the APM and EAP patterns and how to move legacy code to TaskApply

reduction techniques to get aggregated results

Create PLINQ queries and study the factors that impact their performance

Solve concurrency problems caused by producer-consumer race conditions

Discover the synchronization primitives available in .NET Core

Understand how the threading model works with IIS and Kestrel

Find out how you can make the most of server resources

Who this book is for

If you want to learn how task parallelism is used to build robust and scalable enterprise architecture, this book is for you. Whether you are a beginner to parallelism in C# or an experienced architect, you'll find this book useful to gain insights into the different threading models supported in .NET Standard and .NET Core. Prior knowledge of C# is required to understand the concepts covered in this book.

Design Patterns in Modern C++20 - Dmitri Nesteruk 2021-11-20

Apply the latest editions of the C++ standard to

the implementation of design patterns. As well as covering traditional design patterns, this book fleshes out new design patterns and approaches that will be useful to modern C++ developers. Author Dmitri Nesteruk presents concepts as a fun investigation of how problems can be solved in different ways, along the way using varying degrees of technical sophistication and explaining different sorts of trade-offs. *Design Patterns in Modern C++20, Second Edition* also provides a technology demo for modern C++, showcasing how some of its latest features (e.g., coroutines, modules and more) make difficult problems a lot easier to solve. The examples in this book are all suitable for putting into production, with only a few simplifications made in order to aid readability.

What You Will Learn

Use creational patterns such as builder, factories, prototype and singleton

Implement structural patterns such as adapter, bridge, decorator, facade and more

Work with the behavioral patterns such as chain of responsibility, command, iterator, mediator and more

Apply functional design patterns such as the Maybe Monad

Who This Book Is For

This book is for both beginner and experienced C++ developers.

Design Patterns - Erich Gamma 1995
Software -- Software Engineering.