

Liskov Substitution Principle C

Object-Oriented Software Design in C++

Learn the fundamentals of Object-Oriented design by investigating good—and bad—code! Well-designed applications run more efficiently, have fewer bugs, and are easier to revise and maintain. Using an engaging “before-and-after” approach, Object-Oriented Software Design in C++ shows you exactly what bad software looks like and how to fix it with good design principles and patterns. In Object-Oriented Software Design in C++, you’ll find: Design-code-test iterations that improve code with each revision Gathering requirements to make sure you’re developing the right application Design principles like encapsulation and delegation that solve programming problems Design patterns including Observer Design Pattern that fix architecture issues Using recursion and multithreading to simplify common solutions Object-Oriented Software Design in C++ is a vital guide to building the kind of high performance applications delivered by the pros—all using industry-proven design principles and patterns. You’ll learn how to gather and analyze requirements so you’re building exactly what your client is looking for, backtrack mistakes with iterative development, and build a toolbox of design patterns that troubleshoot common issues with application architecture. The book’s accessible examples are written in C++ 17, but its universal principles can be applied to any object-oriented language. Purchase of the print book includes a free eBook in PDF and ePub formats from Manning Publications. About the technology Good design is the foundation of great software. Mastering the principles of object-oriented design is the surest way to create applications that run fast, have few bugs, and last well into the future. Written especially for new C++ programmers, this easy-to-read book gently mentors you in the art of designing great software. About the book Object-Oriented Software Design in C++ introduces object-oriented design principles, practices, and patterns in clear, jargon-free language. The instantly-familiar before-and-after examples highlight the benefits of good design. Each chapter is full of friendly conversations that anticipate your questions and help point out the subtleties you might overlook. Along the way, you’ll pick up tips about idiomatic C++ style that will set your code apart. What's inside Design-code-test iterations Design principles for common programming problems Architecture design patterns in plain English Recursion and multithreading About the reader Examples are in C++ 17. About the author Ronald Mak is a former NASA senior scientist. Currently, he teaches computer science at San Jose State University. The technical editor on this book was Juan Rufes. Table of Contents PART 1 1 The path to well-designed software 2 Iterate to achieve good design PART 2 3 Get requirements to build the right application 4 Good class design to build the application right PART 3 5 Hide class implementations 6 Don’t surprise your users 7 Design subclasses right PART 4 8 The Template Method and Strategy Design Patterns 9 The Factory Method and Abstract Factory Design Patterns 10 The Adapter and Façade Design Patterns 11 The Iterator and Visitor Design Patterns 12 The Observer Design Pattern 13 The State Design Pattern 14 The Singleton, Composite, and Decorator Design Patterns PART 5 15 Designing solutions with recursion and backtracking 16 Designing multithreaded programs

Clean Coder

Verhaltensregeln für professionelle Programmierer Erfolgreiche Programmierer haben eines gemeinsam: Die Praxis der Software-Entwicklung ist ihnen eine Herzensangelegenheit. Auch wenn sie unter einem nicht nachlassenden Druck arbeiten, setzen sie sich engagiert ein. Software-Entwicklung ist für sie eine Handwerkskunst. In Clean Coder stellt der legendäre Software-Experte Robert C. Martin die Disziplinen, Techniken, Tools und Methoden vor, die Programmierer zu Profis machen. Dieses Buch steckt voller praktischer Ratschläge und behandelt alle wichtigen Themen vom professionellen Verhalten und Zeitmanagement über die Aufwandsschätzung bis zum Refactoring und Testen. Hier geht es um mehr als nur um Technik: Es geht um die innere Haltung. Martin zeigt, wie Sie sich als Software-Entwickler professionell verhalten, gut und sauber arbeiten und verlässlich kommunizieren und planen. Er beschreibt, wie Sie sich

schwierigen Entscheidungen stellen und zeigt, dass das eigene Wissen zu verantwortungsvollem Handeln verpflichtet. In diesem Buch lernen Sie: Was es bedeutet, sich als echter Profi zu verhalten Wie Sie mit Konflikten, knappen Zeitplänen und unvernünftigen Managern umgehen Wie Sie beim Programmieren im Fluss bleiben und Schreibblockaden überwinden Wie Sie mit unerbittlichem Druck umgehen und Burnout vermeiden Wie Sie Ihr Zeitmanagement optimieren Wie Sie für Umgebungen sorgen, in denen Programmierer und Teams wachsen und sich wohlfühlen Wann Sie Nein sagen sollten – und wie Sie das anstellen Wann Sie Ja sagen sollten – und was ein Ja wirklich bedeutet Großartige Software ist etwas Bewundernswertes: Sie ist leistungsfähig, elegant, funktional und erfreut bei der Arbeit sowohl den Entwickler als auch den Anwender. Hervorragende Software wird nicht von Maschinen geschrieben, sondern von Profis, die sich dieser Handwerkskunst unerschütterlich verschrieben haben. Clean Coder hilft Ihnen, zu diesem Kreis zu gehören. Über den Autor: Robert C. Uncle Bob Martin ist seit 1970 Programmierer und bei Konferenzen in aller Welt ein begehrter Redner. Zu seinen Büchern gehören Clean Code – Refactoring, Patterns, Testen und Techniken für sauberen Code und Agile Software Development: Principles, Patterns, and Practices. Als überaus produktiver Autor hat Uncle Bob Hunderte von Artikeln, Abhandlungen und Blogbeiträgen verfasst. Er war Chefredakteur bei The C++ Report und der erste Vorsitzende der Agile Alliance. Martin gründete und leitet die Firma Object Mentor, Inc., die sich darauf spezialisiert hat, Unternehmen bei der Vollendung ihrer Projekte behilflich zu sein.

Die C++-Programmiersprache

"The puzzles and problems in Exceptional C++ not only entertain, they will help you hone your skills to become the sharpest C++ programmer you can be. - Many of these problems are culled from the famous Guru of the Week feature of the Internet newsgroup comp.lang.c++, moderated, expanded and updated to conform to the official ISO/ANSI C++ Standard."--BOOK JACKET. - "Try your skills against the C++ masters and come away with the insight and experience to create more efficient, effective, robust, and portable C++ code."--Jacket.

Clean Architecture

Improve readability and understandability of code using C++ best practices Key Features Enrich your coding skills using features from the modern C++ standard and industry approved libraries Implement refactoring techniques and SOLID principles in C++ Apply automated tools to improve your code quality Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionDespite the prevalence of higher-level languages, C++ is still running the world, from bare-metal embedded systems to distributed cloud-native systems. C++ is on the frontline whenever there is a need for a performance-sensitive tool supporting complex data structures. The language has been actively evolving for the last two decades. This book is a comprehensive guide that shows you how to implement SOLID principles and refactor legacy code using the modern features and approaches of C++, the standard library, Boost library collection, and Guidelines Support Library by Microsoft. The book begins by describing the essential elements of writing clean code and discussing object-oriented programming in C++. You'll explore the design principles of software testing with examples of using popular unit testing frameworks such as Google Test. The book also guides you through applying automated tools for static and dynamic code analysis using Clang Tools. By the end of this book, you'll be proficient in applying industry-approved coding practices to design clean, sustainable, and readable real-world C++ code. What you will learn Leverage the rich type system of C++ to write safe and elegant code Create advanced object-oriented designs using the unique features of C++ Minimize code duplication by using metaprogramming Refactor code safely with the help of unit tests Ensure code conventions and format with clang-format Facilitate the usage of modern features automatically with clang-tidy Catch complex bugs such as memory leakage and data races with Clang AddressSanitizer and ThreadSanitizer Who this book is for This book will benefit experienced C++ programmers the most, but is also suitable for technical leaders, software architects, and senior software engineers who want to save on costs and improve software development process efficiency by using modern C++ features and automated tools.

Effektiv C++ programmieren

Exploring C++ uses a series of self-directed lessons to divide C++ into bite-sized chunks that you can digest as rapidly as you can swallow them. The book assumes only a basic understanding of fundamental programming concepts (variables, functions, expressions, statements) and requires no prior knowledge of C or any other particular language. It reduces the usually considerable complexity of C++. The included lessons allow you to learn by doing, as a participant of an interactive education session. You'll master each step in one sitting before you proceed to the next. Author Ray Lischner has designed questions to promote learning new material. And by responding to questions throughout the text, you'll be engaged every step of the way.

Exceptional C++

h2\u003e Kommentare, Formatierung, Strukturierung Fehler-Handling und Unit-Tests Zahlreiche Fallstudien, Best Practices, Heuristiken und Code Smells Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code Aus dem Inhalt: Lernen Sie, guten Code von schlechtem zu unterscheiden Sauberen Code schreiben und schlechten Code in guten umwandeln Aussagekräftige Namen sowie gute Funktionen, Objekte und Klassen erstellen Code so formatieren, strukturieren und kommentieren, dass er bestmöglich lesbar ist Ein vollständiges Fehler-Handling implementieren, ohne die Logik des Codes zu verschleiern Unit-Tests schreiben und Ihren Code testgesteuert entwickeln Selbst schlechter Code kann funktionieren. Aber wenn der Code nicht sauber ist, kann er ein Entwicklungsunternehmen in die Knie zwingen. Jedes Jahr gehen unzählige Stunden und beträchtliche Ressourcen verloren, weil Code schlecht geschrieben ist. Aber das muss nicht sein. Mit Clean Code präsentiert Ihnen der bekannte Software-Experte Robert C. Martin ein revolutionäres Paradigma, mit dem er Ihnen aufzeigt, wie Sie guten Code schreiben und schlechten Code überarbeiten. Zusammen mit seinen Kollegen von Object Mentor destilliert er die besten Praktiken der agilen Entwicklung von sauberem Code zu einem einzigartigen Buch. So können Sie sich die Erfahrungswerte der Meister der Software-Entwicklung aneignen, die aus Ihnen einen besseren Programmierer machen werden – anhand konkreter Fallstudien, die im Buch detailliert durchgearbeitet werden. Sie werden in diesem Buch sehr viel Code lesen. Und Sie werden aufgefordert, darüber nachzudenken, was an diesem Code richtig und falsch ist. Noch wichtiger: Sie werden herausgefordert, Ihre professionellen Werte und Ihre Einstellung zu Ihrem Beruf zu überprüfen. Clean Code besteht aus drei Teilen: Der erste Teil beschreibt die Prinzipien, Patterns und Techniken, die zum Schreiben von sauberem Code benötigt werden. Der zweite Teil besteht aus mehreren, zunehmend komplexeren Fallstudien. An jeder Fallstudie wird aufgezeigt, wie Code gesäubert wird – wie eine mit Problemen behaftete Code-Basis in eine solide und effiziente Form umgewandelt wird. Der dritte Teil enthält den Ertrag und den Lohn der praktischen Arbeit: ein umfangreiches Kapitel mit Best Practices, Heuristiken und Code Smells, die bei der Erstellung der Fallstudien zusammengetragen wurden. Das Ergebnis ist eine Wissensbasis, die beschreibt, wie wir denken, wenn wir Code schreiben, lesen und säubern. Dieses Buch ist ein Muss für alle Entwickler, Software-Ingenieure, Projektmanager, Team-Leiter oder Systemanalytiker, die daran interessiert sind, besseren Code zu produzieren. Über den Autor: Robert C. »Uncle Bob« Martin entwickelt seit 1970 professionell Software. Seit 1990 arbeitet er international als Software-Berater. Er ist Gründer und Vorsitzender von Object Mentor, Inc., einem Team erfahrener Berater, die Kunden auf der ganzen Welt bei der Programmierung in und mit C++, Java, C#, Ruby, OO, Design Patterns, UML sowie Agilen Methoden und eXtreme Programming helfen.

Refactoring with C++

This book contains a collection of thoroughly refereed papers presented at the 6th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2011, held in Beijing, China, in June 2011. The 18 revised and extended full papers presented together with 10 revised short papers were carefully reviewed and selected from 75 initial submissions. The papers cover a wide range of topics, such as software quality and testing, requirements engineering, programming, software processes and methods, software tools and environments, business process and services modeling, software components, software effort and

processes, and socio-technical aspects of software development.

Exploring C++

Improve your existing C++ competencies quickly and efficiently with this advanced volume Professional C++, 5th Edition raises the bar for advanced programming manuals. Complete with a comprehensive overview of the new capabilities of C++20, each feature of the newly updated programming language is explained in detail and with examples. Case studies that include extensive, working code round out the already impressive educational material found within. Without a doubt, the new 5th Edition of Professional C++ is the leading resource for dedicated and knowledgeable professionals who desire to advance their skills and improve their abilities. This book contains resources to help readers: Maximize the capabilities of C++ with effective design solutions Master little-known elements of the language and learn what to avoid Adopt new workarounds and testing/debugging best practices Utilize real-world program segments in your own applications Notoriously complex and unforgiving, C++ requires its practitioners to remain abreast of the latest developments and advancements. Professional C++, 5th Edition ensures that its readers will do just that.

Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code

Annotation This work provides a comprehensive overview of research and practical issues relating to component-based information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application, including the selection and trading of commercial off-the shelf products (COTS).

Entwurfsmuster

This work provides a comprehensive overview of research and practical issues relating to component-based development information systems (CBIS). Spanning the organizational, developmental, and technical aspects of the subject, the original research included here provides fresh insights into successful CBIS technology and application. Part I covers component-based development methodologies and system architectures. Part II analyzes different aspects of managing component-based development. Part III investigates component-based development versus commercial off-the-shelf products (COTS), including the selection and trading of COTS products.

Evaluation of Novel Approaches to Software Engineering

"Masters of Modern C++" is the ultimate guide for aspiring programmers and experienced developers alike, seeking to master the intricacies of this powerful and versatile language. Written in a clear and engaging style, this comprehensive book provides a thorough exploration of C++, empowering you to create sophisticated software applications that redefine the boundaries of possibility. Within these pages, you will embark on a journey through the core concepts of C++, gaining a deep understanding of object-oriented programming, data structures, algorithms, and memory management. Through a blend of theoretical explanations, real-world examples, and thought-provoking exercises, you will develop the skills and knowledge necessary to conquer the challenges of modern software development. As you delve deeper into the chapters of this book, you will discover the secrets of crafting robust and maintainable software applications, embracing the principles of object-oriented design and utilizing advanced programming techniques. You will learn to harness the power of templates, lambdas, and exceptions, unlocking the full potential of C++ to tackle complex problems with elegance and efficiency. Whether you are a seasoned programmer seeking to expand your skillset or a newcomer eager to embark on a career in software development, "Masters of Modern C++" is your ultimate companion. This book will guide you through the intricacies of C++, transforming you into a confident and capable software engineer, ready to conquer the world of programming. With "Masters of Modern C++" as your trusted guide, you will unlock the gateway

to a world of possibilities, where innovation and problem-solving collide. Embrace the challenge, conquer the complexities of C++, and unlock your true potential as a software developer. If you like this book, write a review on google books!

Professional C++

API Design for C++ provides a comprehensive discussion of Application Programming Interface (API) development, from initial design through implementation, testing, documentation, release, versioning, maintenance, and deprecation. It is the only book that teaches the strategies of C++ API development, including interface design, versioning, scripting, and plug-in extensibility. Drawing from the author's experience on large scale, collaborative software projects, the text offers practical techniques of API design that produce robust code for the long term. It presents patterns and practices that provide real value to individual developers as well as organizations. API Design for C++ explores often overlooked issues, both technical and non-technical, contributing to successful design decisions that product high quality, robust, and long-lived APIs. It focuses on various API styles and patterns that will allow you to produce elegant and durable libraries. A discussion on testing strategies concentrates on automated API testing techniques rather than attempting to include end-user application testing techniques such as GUI testing, system testing, or manual testing. Each concept is illustrated with extensive C++ code examples, and fully functional examples and working source code for experimentation are available online. This book will be helpful to new programmers who understand the fundamentals of C++ and who want to advance their design skills, as well as to senior engineers and software architects seeking to gain new expertise to complement their existing talents. Three specific groups of readers are targeted: practicing software engineers and architects, technical managers, and students and educators.

- The only book that teaches the strategies of C++ API development, including design, versioning, documentation, testing, scripting, and extensibility
- Extensive code examples illustrate each concept, with fully functional examples and working source code for experimentation available online
- Covers various API styles and patterns with a focus on practical and efficient designs for large-scale long-term projects

Development of Component-based Information Systems

Comprehensive C++23 resource offering deep coverage from syntax basics to advanced concurrency and standard library usage. Learn best practices to write secure, efficient, and modular C++ code with expert guidance.

Key Features

- In-depth coverage of modern C++23 concepts ensuring comprehensive understanding of language features
- Focus on writing secure, maintainable, and efficient code for professional and scalable projects
- Practical examples and real-world scenarios illustrating advanced techniques and best practices

Book Description

This book begins by grounding readers in the essentials of modern C++23, covering syntax, compiling, and core programming concepts. Early chapters introduce building blocks like data types, functions, and statements, ensuring a solid foundation. Readers also learn coding best practices focused on readability and modularization. As the journey progresses, the focus shifts to object-oriented programming, exploring classes, inheritance, namespaces, and lifecycle management. The text includes advanced topics such as templates, macros, and the integration of C libraries. Readers develop skills in designing secure, maintainable, and extensible code while mastering error handling and testing. The final sections dive into concurrency, standard library features like containers and algorithms, and advanced stream handling. Practical guidance on thread management, synchronization, and modern concurrency tools prepares readers for real-world applications. Concluding chapters present C++ guidelines, emphasizing sustainable and quality code development, completing a comprehensive path from fundamentals to expert-level mastery.

What you will learn

- Understand C++23 syntax and semantics effectively
- Apply object-oriented programming principles with clarity
- Utilize the standard library for data structures and algorithms
- Implement concurrent programming with threads and synchronization
- Write modular and maintainable code following best practices
- Master templates and generic programming techniques

Who this book is for

Ideal for intermediate programmers and software developers with some familiarity in programming concepts, looking to master modern C++23. Readers should have basic understanding of programming logic and syntax. Prior

experience in any procedural or object-oriented language helps, but beginners motivated to learn C++ thoroughly will also benefit.

The Development of Component-based Information Systems

If you need a free PDF practice set of this book for your studies, feel free to reach out to me at cbsenet4u@gmail.com, and I'll send you a copy! THE SOFTWARE DESIGN MCQ (MULTIPLE CHOICE QUESTIONS) SERVES AS A VALUABLE RESOURCE FOR INDIVIDUALS AIMING TO DEEPEN THEIR UNDERSTANDING OF VARIOUS COMPETITIVE EXAMS, CLASS TESTS, QUIZ COMPETITIONS, AND SIMILAR ASSESSMENTS. WITH ITS EXTENSIVE COLLECTION OF MCQS, THIS BOOK EMPOWERS YOU TO ASSESS YOUR GRASP OF THE SUBJECT MATTER AND YOUR PROFICIENCY LEVEL. BY ENGAGING WITH THESE MULTIPLE-CHOICE QUESTIONS, YOU CAN IMPROVE YOUR KNOWLEDGE OF THE SUBJECT, IDENTIFY AREAS FOR IMPROVEMENT, AND LAY A SOLID FOUNDATION. DIVE INTO THE SOFTWARE DESIGN MCQ TO EXPAND YOUR SOFTWARE DESIGN KNOWLEDGE AND EXCEL IN QUIZ COMPETITIONS, ACADEMIC STUDIES, OR PROFESSIONAL ENDEAVORS. THE ANSWERS TO THE QUESTIONS ARE PROVIDED AT THE END OF EACH PAGE, MAKING IT EASY FOR PARTICIPANTS TO VERIFY THEIR ANSWERS AND PREPARE EFFECTIVELY.

Masters of Modern C++

Unlock the potential of design patterns to write better code in C# 11 and .NET 7 KEY FEATURES ? Learn the essentials of C# and object-oriented programming. ? Gain insights into best practices for quality coding. ? Learn how to use design patterns to write code that is reusable, flexible, and maintainable. DESCRIPTION This book is a complete guide to design patterns and object-oriented programming (OOP) in C# and .NET. It covers everything from the basics of C# and Visual Studio to advanced topics like software architecture and best coding practices, including the SOLID principles. The book starts with the basics of C#, .NET, the SOLID principles, and the OOP paradigm. Then, it introduces widely-used design patterns with hands-on examples in C# and .NET. These examples include real-world scenarios and step-by-step instructions. In addition, the book provides an overview of advanced features in the .NET ecosystem, insights into current market solutions for software strategy, and guidance on when to use a design pattern-centric approach. The book concludes with valuable recommendations and best practices for .NET applications, especially when using design patterns. WHAT YOU WILL LEARN ? Learn how to use the Singleton pattern to ensure that only one instance of a class exists in your application. ? Learn how to use the Prototype pattern to create new objects by copying existing objects. ? Learn how to use the Factory Method pattern to create objects without specifying their concrete classes. ? Learn how to use the Adapter pattern to make incompatible interfaces work together. ? Learn how to use the Proxy pattern to control access to objects. ? Learn how to use the Strategy pattern to encapsulate algorithms. WHO THIS BOOK IS FOR This book is invaluable for software developers switching to .NET, experienced .NET developers learning about advanced design patterns, object-oriented programming paradigms, and SOLID principles, and .NET Core enthusiasts looking for information on core functionalities and recent platform advancements. TABLE OF CONTENTS 1. C# Fundamentals 2. .NET Fundamentals 3. Basic Concepts of Object-Oriented Programming in C# 4. SOLID Principles in C# 5. Introduction to Design Patterns 6. Singleton Pattern in .NET Applications 7. Abstract Factory Pattern with Blazor 8. Prototype Pattern with ASP.NET Razor 9. Factory Method Pattern Using New Features on C# 11 10. Adapter Pattern with Entity Framework Core 11. Composite Pattern with ASP.NET MVC 12. Proxy Pattern with GRPC 13. Command Pattern Using MediatR 14. Strategy Pattern Using Azure C# and Azure Functions 15. Observer Pattern

API Design for C++

Embark on a transformative journey into the world of C++ programming with this comprehensive guide, meticulously crafted to empower you with the skills and knowledge necessary to master this powerful and

versatile language. Discover the intricacies of C++, from the fundamentals of data types and control structures to the advanced concepts of object-oriented programming, templates, and memory management. With clear explanations, illustrative examples, and hands-on exercises, this book guides you through the complexities of C++, enabling you to create sophisticated and impactful applications. Whether you're a budding programmer eager to explore the world of C++ or an experienced developer seeking to expand your skillset, this book is your ideal companion. Delve into the core concepts of C++, gaining a thorough understanding of its syntax, semantics, and design principles. Learn how to harness the language's power to create efficient, maintainable, and scalable software applications. With each chapter, you'll delve deeper into the world of C++, gaining the knowledge and skills to tackle increasingly complex programming tasks. Explore a diverse range of topics, including object-oriented programming, data structures, algorithms, file handling, and advanced programming techniques. Discover the practical applications of C++ in various domains, such as game development, web development, operating systems, and machine learning. With its comprehensive coverage, clear explanations, and engaging examples, this book equips you with the skills and knowledge necessary to embark on a successful career as a C++ developer. Whether you aspire to work on cutting-edge software projects, develop innovative mobile apps, or delve into the world of artificial intelligence, this book provides the foundation you need to excel. Master the art of C++ programming and unlock a world of possibilities with this comprehensive guide. Become a proficient C++ developer and create software applications that transform industries and empower users. The journey to C++ mastery begins here. If you like this book, write a review on google books!

C++

- Umfassend überarbeitete und aktualisierte Neuauflage des Standardwerks in vollständig neuer Übersetzung
- Verbesserungsmöglichkeiten von bestehender Software anhand von Code-Smells erkennen und Code effizient überarbeiten
- Umfassender Katalog von Refactoring-Methoden mit Code-Beispielen in JavaScript

Seit mehr als zwanzig Jahren greifen erfahrene Programmierer rund um den Globus auf dieses Buch zurück, um bestehenden Code zu verbessern und leichter lesbar zu machen sowie Software besser warten und erweitern zu können. In diesem umfassenden Standardwerk zeigt Ihnen Martin Fowler, was die Vorteile von Refactoring sind, wie Sie verbesserungsbedürftigen Code erkennen und wie Sie ein Refactoring – unabhängig von der verwendeten Programmiersprache – erfolgreich durchführen. In einem umfangreichen Katalog gibt Fowler Ihnen verschiedene Refactoring-Methoden mit ausführlicher Erläuterung, Motivation, Vorgehensweise und einfachen Beispielen in JavaScript an die Hand. Darüber hinaus behandelt er insbesondere folgende Schwerpunkte:

- Allgemeine Prinzipien und Durchführung des Refactorings
- Refactoring anwenden, um die Lesbarkeit, Wartbarkeit und Erweiterbarkeit von Programmen zu verbessern
- Code-Smells erkennen, die auf Verbesserungsmöglichkeiten durch Refactoring hinweisen
- Entwicklung zuverlässiger Tests für das Refactoring
- Erkennen von Fallstricken und notwendigen Kompromissen bei der Durchführung eines Refactorings

Diese vollständig neu übersetzte Ausgabe wurde von Grund auf überarbeitet, um den maßgeblichen Veränderungen der modernen Programmierung Rechnung zu tragen. Sie enthält einen aktualisierten Katalog von Refactoring-Methoden sowie neue Beispiele für einen funktionalen Programmieransatz.

SOFTWARE DESIGN

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++.

Implementing Design Patterns in C# 11 and .NET 7

Gain the confidence you need to apply machine learning in your daily work. With this practical guide, author Matthew Kirk shows you how to integrate and test machine learning algorithms in your code, without the academic subtext. Featuring graphs and highlighted code examples throughout, the book features tests with Python's Numpy, Pandas, Scikit-Learn, and SciPy data science libraries. If you're a software engineer or business analyst interested in data science, this book will help you: Reference real-world examples to test each algorithm through engaging, hands-on exercises Apply test-driven development (TDD) to write and run tests before you start coding Explore techniques for improving your machine-learning models with data extraction and feature development Watch out for the risks of machine learning, such as underfitting or overfitting data Work with K-Nearest Neighbors, neural networks, clustering, and other algorithms

Mastering the Craft of C++: Practical Programming Made Easy

An integrated guide to C++ and computational finance This complete guide to C++ and computational finance is a follow-up and major extension to Daniel J. Duffy's 2004 edition of Financial Instrument Pricing Using C++. Both C++ and computational finance have evolved and changed dramatically in the last ten years and this book documents these improvements. Duffy focuses on these developments and the advantages for the quant developer by: Delving into a detailed account of the new C++11 standard and its applicability to computational finance. Using de-facto standard libraries, such as Boost and Eigen to improve developer productivity. Developing multiparadigm software using the object-oriented, generic, and functional programming styles. Designing flexible numerical algorithms: modern numerical methods and multiparadigm design patterns. Providing a detailed explanation of the Finite Difference Methods through six chapters, including new developments such as ADE, Method of Lines (MOL), and Uncertain Volatility Models. Developing applications, from financial model to algorithmic design and code, through a coherent approach. Generating interoperability with Excel add-ins, C#, and C++/CLI. Using random number generation in C++11 and Monte Carlo simulation. Duffy adopted a spiral model approach while writing each chapter of Financial Instrument Pricing Using C++ 2e: analyse a little, design a little, and code a little. Each cycle ends with a working prototype in C++ and shows how a given algorithm or numerical method works. Additionally, each chapter contains non-trivial exercises and projects that discuss improvements and extensions to the material. This book is for designers and application developers in computational finance, and assumes the reader has some fundamental experience of C++ and derivatives pricing. HOW TO RECEIVE THE SOURCE CODE Once you have purchased a copy of the book please send an email to the author dduffy@datasim.nl requesting your personal and non-transferable copy of the source code. Proof of purchase is needed. The subject of the mail should be "C++ Book Source Code Request". You will receive a reply with a zip file attachment.

Refactoring

Write maintainable, extensible, and durable software with modern C++. This book is a must for every developer, software architect, or team leader who is interested in good C++ code, and thus also wants to save development costs. If you want to teach yourself about writing clean C++, Clean C++ is exactly what you need. It is written to help C++ developers of all skill levels and shows by example how to write understandable, flexible, maintainable, and efficient C++ code. Even if you are a seasoned C++ developer, there are nuggets and data points in this book that you will find useful in your work. If you don't take care with your code, you can produce a large, messy, and unmaintainable beast in any programming language. However, C++ projects in particular are prone to be messy and tend to slip into bad habits. Lots of C++ code that is written today looks as if it was written in the 1980s. It seems that C++ developers have been forgotten

by those who preach Software Craftsmanship and Clean Code principles. The Web is full of bad, but apparently very fast and highly optimized C++ code examples, with cruel syntax that completely ignores elementary principles of good design and well-written code. This book will explain how to avoid this scenario and how to get the most out of your C++ code. You'll find your coding becomes more efficient and, importantly, more fun. What You'll Learn Gain sound principles and rules for clean coding in C++ Carry out test driven development (TDD) Discover C++ design patterns and idioms Apply these design patterns Who This Book Is For Any C++ developer and software engineer with an interest in producing better code.

Design Patterns in Modern C++

Innerhalb kürzester Zeit hat Java die Programmierung revolutioniert. Java ist eine Sprache, die basierend auf C und C++ die Tradition dieser führenden Programmiersprachen in einer neuartigen und überraschenden Weise fortsetzt. So kann ein Java-Programm auf jedem beliebigen Rechner weltweit ohne irgendeine Änderung laufen, wenn für den entsprechenden Rechner eine virtuelle Maschine verfügbar ist. Für die Programmierung im Internet bietet Java die Technik der Applets, d.h. von kleinen Programmen, die in einer Web-Seite laufen. Diese Technik erlaubt es, von irgendeinem Rechner des Internets auf seinen eigenen Rechner Web-Seiten mit Applets zu laden und diese Applets lokal im Browser des eigenen Rechners auszuführen. Java findet seinen Einsatz jedoch nicht nur in Netzen, sondern auch als universell einsetzbare Programmiersprache für die verschiedensten Aufgabenstellungen. Das vorliegende Buch setzt weder C noch C++ Kenntnisse voraus. Es erlaubt dem Einsteiger, Java ohne Vorkenntnisse anderer Programmiersprachen zu erlernen. Daher der Titel „Java als erste Programmiersprache“. Dabei hat dieses Buch das ehrgeizige Ziel, dem Neuling die Sprachkonzepte von Java und die Grundkonzepte der objektorientierten Programmierung so präzise wie möglich und dennoch in leicht verständlicher Weise vorzustellen. Für den erfahrenen C++ Programmierer lohnt sich ein Blick in Anhang D, wo die wesentlichen Unterschiede zwischen C++ und Java übersichtlich dargestellt sind.

Thoughtful Machine Learning with Python

Über den Umbau von IT-Landschaften mit Domain-Driven Design kompakter, tiefgehender Einblick in Domain-Driven Design (DDD) und die Verwendung der vielfältigen DDD-Techniken in der Praxis – Fokussierung auf Legacy-Systeme und Migration in Richtung gut strukturierter Monolithen und Microservices Zusammenhang zwischen Transformation der Architektur und der Teamorganisation In den letzten Jahrzehnten ist sehr viel Software entwickelt worden, die wir heute modernisieren und zukunftsfähig machen wollen. Domain-Driven Design (DDD) eignet sich hervorragend, um große Legacy-Systeme in Microservices zu zerlegen oder zu wartbaren Monolithen umzubauen. In diesem Transformationsprozess wird der fachliche Kern des Softwaresystems herausgearbeitet. Auf der Makro-Architektur-Ebene wird das System in unabhängige DDD-Module aufgeteilt. Auf der Mikro-Architektur-Ebene werden zusätzlich DDD-Muster eingesetzt, um den Code weiter zu strukturieren. Dieses Buch kombiniert Anleitungen, Erkenntnisse und Beispiele aus der Beraterpraxis des Autorenteam. Es veranschaulicht, wie Techniken aus DDD und Refactorings verwendet werden können, um Softwaresysteme zu transformieren, die über Jahre gewachsen und möglicherweise ohne Architekturverständnis entwickelt wurden. Die Leser*innen lernen einen Prozess der Transformation kennen, den sie als Ganzes oder schrittweise in ihre Alltagspraxis übernehmen können, um die Wartbarkeit ihrer Legacy-Systeme effektiv und schnell zu verbessern. Sie erhalten: einen detaillierten Einblick in DDD und die Verwendung der vielfältigen DDD-Techniken in der Praxis, eine Anleitung, wie DDD eingesetzt werden kann, damit Legacy-Systeme wartbar bleiben oder wartbarer werden, eine Anleitung, wie vorzugehen ist, falls eine Zerlegung der Monolithen und eine Neustrukturierung der Microservices geplant ist, Argumente an die Hand, wann eine Zerlegung in Microservices für einen Monolithen sinnvoll ist und wann nicht.

Financial Instrument Pricing Using C++

Der erfolgreiche "Grundkurs Java" in der 6. Auflage. Das Buch führt Sie schrittweise durch die wichtigsten

Aspekte von Java, von den elementaren Grundlagen über generische Klassen und Methoden, Dateiverarbeitung, grafische Benutzungsoberflächen bis zu Datenbankzugriffen und zur Netzwerkkommunikation über TCP/IP. Zahlreiche ergänzende Erläuterungen zu den behandelten Themen wurden aufgenommen sowie neue Themen hinzugefügt: u. a. eine Einführung in das Persistenz-API zur Speicherung von Objekten in relationalen Datenbanken und eine Einführung in das objektorientierte Datenbanksystem db4o. Das Fallbeispiel wurde überarbeitet und neu implementiert. Der Quellcode von über 290 Programmbeispielen (inkl. Lösungen zu den Aufgaben) ist im Internet verfügbar.

Clean C++

Mit Java hat sich in der Industrie eine Programmiersprache durchgesetzt, die weit über die Konzepte traditioneller Programmiersprachen hinausgeht. Dieses Buch setzt keine Kenntnisse in anderen Programmiersprachen voraus, sondern richtet sich an jene Schüler, Studenten und Praktiker, die nicht nur in Java schnuppern, sondern die Grundlagen von Java und vielleicht auch schon die fortgeschrittenen Themen professionell erlernen wollen. Behandelt werden alle grundlegenden Sprachmittel, die zur Erstellung von Java-Programmen erforderlich sind. Alle zum Verständnis erforderlichen Hintergrundinformationen werden anschaulich und präzise dargestellt.

Java als erste Programmiersprache

Unlock the power of C++, a cornerstone language in software development, with this comprehensive guide. Whether you're starting your programming journey or looking to solidify your understanding, this book provides a thorough exploration of C++ from foundational concepts to modern features. Begin by setting up your development environment and writing your first program. Master the essentials, including variables, data types, memory management, operators, and controlling program flow with conditional statements and loops. Learn to build modular and reusable code with functions, exploring parameter passing techniques like pass-by-value. Understand how to handle collections of data effectively using arrays and gain crucial insights into the power and pitfalls of pointers. Dive into Object-Oriented Programming (OOP) concepts. Discover how to define classes and objects, encapsulating data and behavior. Explore the mechanisms of inheritance and polymorphism to create flexible and extensible applications. Master constructors and destructors for effective object lifecycle management. Navigate the Standard Template Library (STL), harnessing the power of containers like vectors, deques, lists, sets, and maps, along with generic algorithms for efficient data manipulation. Learn to interact with files for persistent data storage using C++ streams. Finally, get acquainted with modern C++ features like auto type deduction, range-based for loops, smart pointers for automatic resource management (RAII), lambda expressions, and move semantics, which enhance code safety, readability, and performance. This book equips you with the knowledge and skills to write robust, efficient, and modern C++ code.

Domain-Driven Transformation

C++ For Artists The Art, Philosophy, and Science of Object-Oriented Programming takes a refreshing and sometimes controversial approach to the complex topic of object-oriented programming and the C++ language. Intended as both a classroom and reference t

Grundkurs JAVA

ETAPS 2001 is the fourth instance of the European Joint Conferences on Theory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprises ve conferences (FOSSACS, FASE, ESOP, CC, TACAS), ten satellite workshops (CMCS, ETI Day, JOSES, LDTA, MMAABS, PFM, ReMiS, UNIGRA, WADT, WTUML), seven invited lectures, a debate, and ten tutorials. The events that comprise ETAPS address various aspects of the system - velopment process, including speci cation, design, implementation, analysis

and improvement. The languages, methodologies and tools which support these - tivities are all well within its scope. Di erent blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive.

Java als erste Programmiersprache

- Architekturmuster - Technische Konzepte - Architekturanalyse und -bewertung - Dokumentation von Architekturen - Modernisierung bestehender Systeme - Beispiele realer Softwarearchitekturen - iSAQB Curriculum - Neu in der 10. Auflage: C4, Cloud, DataContracts - Ihr exklusiver Vorteil: E-Book inside beim Kauf des gedruckten Buches Softwarearchitekt*innen müssen komplexe fachliche und technische Anforderungen an IT-Systeme umsetzen und sie müssen diese Systeme durch nachvollziehbare Strukturen flexibel und erweiterbar gestalten. Dieser Praxisleitfaden zeigt Ihnen, wie Sie Softwarearchitekturen effektiv und systematisch entwickeln können. Gernot Starke unterstützt Sie mit konkreten praktischen Tipps aus seiner Erfahrung. Sie finden Antworten auf zentrale Fragen: - Welche Aufgaben gehören zur Softwarearchitektur? - Wie kann ich beim Entwurf vorgehen? - Wie kommuniziere und dokumentiere ich Softwarearchitekturen? - Wie helfen Architekturmuster? - Wie analysiere und bewerte ich Softwarearchitekturen? - Wie setze ich Persistenz, grafische Benutzeroberflächen, Geschäftsregeln, Integration, Verteilung, Fehlerbehandlung und andere Konzepte ein? - Was muss ich über Domain-Driven Design, arc42, Microservices, C4 und die Cloud wissen? - Wie verbessere ich bestehende Systeme?

Learn C++

Implement robust applications by applying efficient Design Patterns with .NET 5 and C# KEY FEATURES ? Detailed theoretical concepts covered, including the use of encapsulation, interfaces, and inheritance. ? Access to solutions applied for software strategy and final product output. ? Simplified demonstration of real applications implementing numerous design patterns. DESCRIPTION This book covers detailed aspects of Design Patterns and Object-Oriented Programming concepts using the most modern version of the C# language and .NET platform, including many real-world examples and good practice guidelines that help developers in building robust and extensible applications. The book begins with the essential concepts of C# programming and the .NET platform. You get your foundation strong by understanding SOLID Principles and the actual implementation of reliable applications. You will be working on most common Design Patterns such as Abstract Factory, Adapter, Composite, Proxy, Command, Strategy, Observer, Factory Method, Singleton, Builder, Interpreter, Mediator, and many other patterns that will help you to create solid enterprise applications. You will also witness the performance of these design patterns in a real software development environment with the help of practical examples. After learning the most common Design Patterns practiced in .NET enterprise applications, the reader will be able to understand and apply good practices of software development based on the object-oriented paradigm to develop complex enterprise applications efficiently and simply. WHAT YOU WILL LEARN ? Fine-tune your knowledge about interfaces, polymorphism, and encapsulation. ? Learn to practice implementing design patterns in enterprise applications. ? Implement rich design patterns: Observer, Strategy, Command, Proxy, and more. ? Get to learn the latest additional design patterns such as Builder, Bridge, and Decorator. ? Includes illustrations, examples, and real use-cases of .NET 5.0 applications. WHO THIS BOOK IS FOR This book is for .NET developers, application developers, and software engineers who want to develop .NET applications with proven techniques and build error-free applications. This book also attracts fresh graduates and entry-level developers as long as basic knowledge about .NET is known to them. TABLE OF CONTENTS 1. C# Fundamentals 2. Introduction to .NET 5 3. Basic Concepts of Object-Oriented Programming 4. Interfaces in C# 5. Encapsulation and Polymorphism in C# 6. SOLID Principles in C# 7. Abstract Factory 8. Abstract Factory 9. Prototype 10. Factory Method 11. Adapter 12. Composite 13. Proxy 14. Command 15. Strategy 16. Observer 17. Good Practices and Additional Design Patterns

C++ for Artists

This comprehensive guide to C++ programming will equip you with the knowledge and skills you need to create robust, maintainable software applications. Whether you are a beginner or an experienced programmer, this book will take you from the basics of C++ to advanced concepts and techniques. With clear explanations, hands-on examples, and in-depth coverage of C++ features, this book will help you:

- * Master the fundamentals of C++, including variables, data types, operators, control flow statements, and functions
- * Understand object-oriented programming concepts such as classes, inheritance, and polymorphism
- * Explore advanced C++ techniques such as templates, lambda expressions, and multithreading
- * Design and implement efficient algorithms, handle errors and exceptions, and optimize your code for performance
- * Learn about the latest trends and developments in C++ programming, including its applications in artificial intelligence, machine learning, and cloud computing

This book is the perfect resource for anyone who wants to master the art of C++ programming. With its comprehensive coverage of C++ concepts and its focus on practical application, this book will help you build the skills you need to succeed in today's competitive software development landscape. Whether you are a student learning C++ for the first time or a seasoned programmer looking to expand your skills, this book is the perfect companion on your journey to C++ mastery. If you like this book, write a review on google books!

Fundamental Approaches to Software Engineering

This book constitutes the refereed proceedings of the 8th International Symposium on Software Composition, SC 2009, held in Zurich, Switzerland, in July 2009. The workshop has been organized as an event co-located with the TOOLS Europe 2009 conference. The 10 revised full papers presented together with 2 invited lectures were carefully reviewed and selected from 34 submissions. The papers reflect current research in software composition to foster developing of composition models and techniques by using aspect- and service-oriented programming, specification of component contracts and protocols, methods of correct components composition, as well as verification, validation and testing techniques - even in pervasive computing environments and for the Web.

Effektive Softwarearchitekturen

Die Einführung in die Programmiersprachen C++, C++/CLI und die Windows-Programmierung verbindet das Buch mit Visual Studio und richtet sich damit an Anfänger wie an erfahrene C++-Programmierer. Alle Sprachelemente und Konzepte von Standard-C++ sind ausführlich und praxisnah dargestellt. Weitere Themen sind objektorientierte Analyse und Design, Templates sowie die Programmierung mit der .NET-Klassenbibliothek. Das Buch enthält viele Übungen und eignet sich gleichermaßen für Autodidakten wie für Studierende an Fachhochschulen und Universitäten.

Implementing Design Patterns in C# and .NET 5

Unlock the full potential of software development with \"Mastering Object-Oriented Design Patterns in Modern C++: Unlock the Secrets of Expert-Level Skills.\" This comprehensive guide is meticulously crafted for experienced programmers eager to deepen their understanding of design patterns and how they revolutionize software architecture. With a focus on modern C++ advancements, this book equips you with the knowledge to create robust, scalable, and efficient applications tailored to the challenges of today's fast-paced digital landscape. Embodying a blend of theoretical insight and practical application, this book delves into the intricacies of object-oriented principles and the strategic implementation of creational, structural, and behavioral patterns. Each chapter is designed to enhance your proficiency, from advanced template metaprogramming to concurrent programming strategies. Moreover, nuanced discussions on memory management, best practices, and anti-patterns further prepare you to craft streamlined code that not only meets, but exceeds, industry standards. Dive into expertly curated content that demystifies complex programming concepts and empowers you to elevate your software development approach. Through clear

explanations, real-world examples, and insightful advice, \"Mastering Object-Oriented Design Patterns in Modern C++\" transforms theoretical knowledge into practical mastery. Whether you are architecting applications for personal or enterprise needs, this book will serve as your definitive guide to mastering design excellence in the realm of modern C++.

The Art of Effective C++: Building Robust Software with Precision

EduGorilla Publication is a trusted name in the education sector, committed to empowering learners with high-quality study materials and resources. Specializing in competitive exams and academic support, EduGorilla provides comprehensive and well-structured content tailored to meet the needs of students across various streams and levels.

Software Composition

C++ mit Microsoft Visual C++ 2008

<https://forumalternance.cergyponoise.fr/19955698/fpromptc/svisitk/alimitj/toyota+camry+repair+manual.pdf>
<https://forumalternance.cergyponoise.fr/60450719/rcommencej/igotow/qarisel/2000+audi+a6+quattro+repair+guide>
<https://forumalternance.cergyponoise.fr/58008092/jrescueh/mexeg/bthankd/academic+culture+jean+brick+2011.pdf>
<https://forumalternance.cergyponoise.fr/66545918/mhoper/pgog/killustratee/philip+kotler+marketing+management->
<https://forumalternance.cergyponoise.fr/19456063/runiteg/plinkz/vsmashl/1998+yamaha+xt350+service+repair+ma>
<https://forumalternance.cergyponoise.fr/98782142/mspecifyt/gdla/nconcernv/das+us+amerikanische+discovery+ver>
<https://forumalternance.cergyponoise.fr/27093794/lspecifyz/amirrorf/wembodyu/libri+elettrotecnica+ingegneria.pdf>
<https://forumalternance.cergyponoise.fr/35894504/lsoundd/fniche/yawardi/scholarships+grants+prizes+2016+peter>
<https://forumalternance.cergyponoise.fr/53458761/ptesta/yuploadl/eeditg/psoriasis+treatment+with+homeopathy+sc>
<https://forumalternance.cergyponoise.fr/22263304/dcovere/fgotoq/spractiser/fundamentals+of+engineering+econom>