

Tpl Dataflow Example Reactive Programming Net

Maximizing the performance of your algorithms and applications is extremely important and can give you a competitive advantage, a lower cost of ownership, and happier users. Pro .NET Performance explains the internals of Windows, the CLR, and the physical hardware that affect the performance of your applications, and gives you the knowledge and tools to measure how your code performs in isolation from external factors. The book is full of C# code samples and tips to help you squeeze every bit of juice from your application—lower memory utilization, consistent CPU usage, and fewer I/O operations across the network and disk. Pro .NET Performance will change the way you think about .NET application development. Guides you through performance measurement with a variety of profilers and other tools Explains how OS and CLR internals affect your application 's performance in unexpected ways Provides you with tips and real-life case studies for improving application performance

Buy the print C# 5.0 Unleashed and get the eBook version for free! See inside the book for access code and details. C# 5.0 Unleashed is for anyone who wants to learn the C# programming language in depth, understanding how language features truly work. While giving you those insights, you learn where and how to use the features to design various kinds of software. This book not only teaches the language 's capabilities, it also looks behind the scenes to build a solid foundation to aid you in understanding the .NET platform as a whole.  Bart De Smet offers exceptional insight into the features of both the language and Microsoft 's broader framework. He doesn 't just cover the " what " and " how " of effective C# programming; He explains the " why , " so you can consistently choose the right language and platform features, maximizing your efficiency and effectiveness.  The early chapters introduce the .NET platform, the tooling ecosystem, and the C# programming language, followed by in-depth coverage of the C# Framework language itself, with immediate application of language features. The last chapters give an overview of the .NET Framework libraries about which every good developer on the platform should know. Understanding the .NET platform: its language support, libraries, tools, and more Learn where C# fits, how it has evolved, and where it 's headed Master essential language features including expressions, operators, types, objects, and methods Efficiently manage exceptions and resources Write more effective C# object-oriented code Make the most of generics, collections, delegates, reflection, and other advanced language features Use LINQ to express queries for any form of data Master dynamic programming techniques built on .NET 's Dynamic Language Runtime (DLR) Work with namespaces, assemblies, and application domains Write more efficient code using threading, synchronization, and advanced parallel programming techniques Learn the Base Class Library (BCL) to quickly perform many common tasks Instrument, diagnose, test, and troubleshoot your C# code Understand how to use the new C# 5.0 asynchronous programming features Leverage interoperability with Windows Runtime to build Windows 8 applications

F# brings the power of functional-first programming to the .NET Framework, a platform for developing software in the Microsoft Windows ecosystem. If you're a traditional .NET developer used to C# and Visual Basic, discovering F# will be a revelation that will change how you code, and how you think about coding. In The Book of F#, Microsoft MVP Dave Fancher shares his expertise and teaches you how to wield the power of F# to write succinct, reliable, and predictable code. As you learn to take advantage of features like default immutability, pipelining, type inference, and pattern matching, you'll be amazed at how efficient and elegant your code can be. You'll also learn how to: Exploit F#'s functional nature using currying, partial application, and delegation Streamline type creation and style with record types and discriminated unions Use collection types and modules to handle data sets more effectively Use pattern matching to decompose complex types and branch your code into a single in-line expression Make your software more responsive with parallel programming and asynchronous workflows Harness object orientation to develop rich frameworks and interact with other .NET languages Use query expressions and type providers to access and manipulate data sets from disparate sources Break free of that old school of programming. The Book of F# will show you how to unleash the expressiveness of F# to create smarter, leaner code. Get up and running with reactive programming paradigms to build fast, concurrent, and powerful applications About This Book Get to grips with the core design principles of reactive programming Learn about Reactive Extensions for .NET through real-world examples Improve your problem-solving ability by applying functional programming Who This Book Is For If you are a .NET developer who wants to implement all the reactive programming paradigm techniques to create better and more efficient code, then this is the book for you. No prior knowledge of reactive programming is expected. What You Will Learn Create, manipulate, and aggregate sequences in a functional-way Query observable data streams using standard LINQ query operators Program reactive observers and observable collections with C# Write concurrent programs with ease, scheduling actions on various workers Debug, analyze, and instrument Rx functions Integrate Rx with CLR events and custom scheduling operators Implement Functional Fit in Detail Reactive programming is an innovative programming paradigm focused on time-based problem solving. It makes your programs better-performing, easier to scale, and more reliable. Want to create fast-running applications to handle complex logic and huge datasets for financial and big-data challenges? Then you have picked up the right book! Starting with the principles of reactive programming and unveiling the power of the pull-programming world, this book is your one-stop solution to get a deep practical understanding of reactive programming techniques. You will gradually learn all about reactive extensions, programming, testing, and debugging observable sequences, and integrating events from CLR data-at-rest or events. Finally, you will dive into advanced techniques such as manipulating time in data-flow, customizing operators and providers, and exploring functional reactive programming. By the end of the book, you'll know how to apply reactive programming to solve complex problems and build efficient programs with reactive user interfaces. Style and approach This is a concise reference manual for reactive programming with Rx for C# and F# using real-world, practical examples.

Enhancing the Resilience of the Nation's Electricity System

Building Windows 8, Web, and Desktop Applications for the .NET 4.5 Framework

To .NET Performance

NET Design Patterns

Writing High-Performance .Net Code

Tools and Techniques to Debug and Solve Real-World Problems In .NET

Americans' safety, productivity, comfort, and convenience depend on the reliable supply of electric power. The electric power system is a complex "cyber-physical" system composed of a network of millions of components spread out across the continent. These components are owned, operated, and regulated by thousands of different entities. Power system operators work hard to assure safe and reliable service, but large outages occasionally happen. Given the nature of the system, there is simply no way that outages can be completely avoided, no matter how much time and money is devoted to such an effort. The system's reliability and resilience can be improved but never made perfect. Thus, system owners, operators, and regulators must prioritize their investments based on potential benefits. Enhancing the Resilience of the Nation's Electricity System focuses on identifying, developing, and implementing strategies to increase the power system's resilience in the face of events that can cause large-area, long-duration outages; blackouts that extend over multiple service areas and last several days or longer. Resilience is not just about lessening the likelihood that these outages will occur. It is also about limiting the scope and impact of outages when they do occur, restoring power rapidly afterwards, and learning from these experiences to better deal with events in the future.

If you're one of many developers still uncertain about concurrent and multithreaded development, this practical cookbook will change your mind. With more than 85 code-rich recipes in this updated second edition, author Stephen Cleary demonstrates parallel processing and asynchronous programming techniques using libraries and language features in .NET and C# 8.0. Concurrency is now more common in responsive and scalable application development, but it's still extremely difficult to code. The detailed solutions in this cookbook show you how modern tools raise the level of abstraction, making concurrency much easier than before. Conquer concurrency with ready-to-use code and discussions about how and why solutions work, these recipes help you: Get up to speed on concurrency and async and parallel programming Use async and await for asynchronous operations Enhance your code with asynchronous streams Explore parallel programming with .NET's Task Parallel Library Create dataflow pipelines with .NET's TPL Dataflow Library Understand the capabilities that System.Reactive builds on top of LINQ Utilize threadsafe and immutable collections Learn how to conduct unit testing with concurrent code Make the thread pool work for you Enable clean, cooperative cancellation Examine scenarios for combining concurrent approaches Dive into asynchronous-friendly object-oriented programming Recognize and write adapters for code using other asynchronous styles

C# is undeniably one of the most versatile programming languages available to engineers today. With this comprehensive guide, you'll learn just how powerful the combination of C# and .NET can be. Author Ian Griffiths guides you through C# 8.0 fundamentals and techniques for building cloud, web, and desktop applications. Designed for experienced programmers, this book provides many code examples to help you work with the nuts and bolts of C#, such as generics, LINQ, and asynchronous programming features. You'll get up to speed on .NET Core and the latest C# 8.0 additions, including asynchronous streams, nullable references, pattern matching, default interface implementation, ranges and new indexing syntax, and changes in the .NET tool chain. Discover how C# supports fundamental coding features, such as classes, other custom types, collections, and error handling. Learn how to write high-performance memory-efficient code with .NET Core's Span and Memory types Query and process diverse data sources, such as in-memory object models, databases, data streams, and XML documents with LINQ Use .NET's multithreading features to exploit your computer's parallel processing capabilities Learn how asynchronous language features can help improve application responsiveness and scalability

There is a software gap between the hardware potential and the performance that can be attained using today's software parallel program development tools. The tools need manual intervention by the programmer to parallelize the code. Programming a parallel computer requires closely studying the target algorithm or application, more so than in the traditional sequential programming we have all learned. The programmer must be aware of the communication and data dependencies of the algorithm or application. This book provides the techniques to explore the possible ways to program a parallel computer for a given application.

Parallel Programming

Dataflow and Reactive Programming in .net

Concurrency in .NET

Multithreading with C# Cookbook

Design Patterns in Modern C++

Dataflow and Reactive Programming Systems

Summary Concurrency in .NET teaches you how to build concurrent and scalable programs in .NET using the functional paradigm. This intermediate-level guide is aimed at developers, architects, and passionate computer programmers who are interested in writing code with improved speed and effectiveness by adopting a declarative and pain-free programming style. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Unlock the incredible performance built into your multi-processor machines. Concurrent applications run faster because they spread work across processor cores, performing several tasks at the same time. Modern tools and techniques on the .NET platform, including parallel LINQ, functional programming, asynchronous programming, and the Task Parallel Library, offer powerful alternatives to traditional thread-based concurrency. About the Book Concurrency in .NET teaches you to write code that delivers the speed you need for performance-sensitive applications. Featuring examples in both C# and F#, this book guides you through concurrent and parallel designs that emphasize functional programming in theory and practice. You'll start with the foundations of concurrency and master essential techniques and design practices to optimize code running on modern multiprocessor systems. What's Inside The most important concurrency abstractions Employing the agent programming model Implementing real-time event-stream processing Executing unbounded asynchronous operations Best concurrent practices and patterns that apply to all platforms About the Reader For readers skilled with C# or F#. About the Book Riccardo Ferrero is a senior engineering manager at Microsoft MVP who is passionate about functional programming. He has over 20 years' experience delivering cost-effective technology solutions in a competitive business environment. Table of ContentsPART I - Benefits of functional programming applicable to concurrent programs Functional concurrency foundations Functional programming techniques for concurrency Functional data structures and immutability PART 2 - How to approach the different parts of a concurrent program The types of processing: data, data parallelism, part 2 Real-time event streams: functional reactive programming Task-based functional parallelism Task asynchronicity for the Async/Worker functional programming in F# Functional combinators for fluent concurrent programming

Applying reactive programming everywhere with agents Parallel workflow and agent programming with TPL Dataflow PART 3 - Modern patterns of concurrent programming applied Recipes and design patterns for successful concurrent programming Building a scalable mobile app with concurrent functional programming This book presents cutting-edge applications of, and up-to-date research on, ontology engineering techniques in the physical asset integrity domain. Through a survey of state-of-the-art theory and methods on ontology engineering, the authors emphasize essential topics including data integration modeling, knowledge representation, and semantic interpretation. The book also reflects novel topics dealing with the advanced problems of physical asset integrity applications such as heterogeneity, data inconsistency, and interoperability existing in design and utilization. With a distinctive focus on applications relevant in heavy industry, Ontology Modeling in Physical Asset Integrity Management is ideal for practicing industrial and mechanical engineers working in the field, as well as researchers and graduate concerned with ontology engineering in physical systems life cycles.

Conquer complex and interesting programming challenges by building robust and concurrent applications with caches, cryptography, and parallel programming. Key FeaturesUnderstand how to use .NET frameworks like the Task Parallel Library (TPL)and CryptoAPIDevelop a containerized application based on microservices architectureGain insights into memory management techniques in .NET CoreBook Description This Learning Path shows you how to create high performing applications and solve programming challenges using a wide range of C# features. You 'll begin by learning how to identify the bottlenecks in writing programs, highlight common performance pitfalls, and apply strategies to detect and resolve these issues early. You'll also study the importance of micro-services architecture for building fast applications and implementing resiliency and security in .NET Core. Then, you'll study the importance of defining and testing boundaries, abstracting away third-party code, and working with different types of job double, such as spies, mocks, and fakes. In addition to describing programming trade-offs, this Learning Path will also help you build a useful toolkit of techniques, including value caching, statistical analysis, and geometric algorithms. This Learning Path includes content from the following Packt products: C# 7 and .NET Core 2.0 High Performance by Ovais Mehboub Ahmed KhanPractical Test-Driven Development using C# 7 by John Callaway, Clayton HuntThe Modern C# Challenge by Rod StephensWhat you will learnMeasure application performance using BenchmarkDotNetLeverage the Task Parallel Library (TPL) and Parallel Language Integrated Query (PLINQ)library to perform asynchronous operations using Parallel.Invoke and PLINQ to search directories for files matching patternsFind patterns in strings using regular expressions and pattern matching in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

The analysis and sorting of large numbers of cells with a fluorescence-activated cell sorter (FACS) was first achieved some 30 years ago. Since then, this technology has been rapidly developed and is used today in many laboratories. A Springer Lab Manual Review of the First Edition: "This is a most useful volume which will be a welcome addition for personal use and also for laboratories in a wide range of disciplines. Highly recommended." CYTOBIOS

Asynchronous, Parallel, and Multithreaded Programming

Multithreading with C# Cookbook Second Edition

PHP Architect's Guide to PHP Design Patterns

For Multicore and Cluster Systems

Professional C# 7 and .NET Core 2.0

Introducing Erlang

Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a graphical music player and work with fast, reliable networking software using Tokio, the scalable and productive asynchronous IO Rust library. Over the course of this book, you'll explore various features of Rust Programming including its SDL features, event loop, File I/O, and the famous GTK+ widget toolkit. Through these projects, you'll see how well Rust performs in terms of concurrency—including parallelism, reliability, improved performance, generics, macros, and thread safety. We'll also cover some asynchronous and reactive programming aspects of Rust. By the end of the book, you'll be comfortable building various real-world applications in Rust. What you will learn Compile and run the Rust projects using the Cargo-Rust Package manager Use Rust-SDL features such as the event loop, windows, infinite loops, pattern matching, and more Create a graphical interface using GTK+ and Rust-SDL Incorporate concurrency mechanism and multi-threading along with thread safety and locks Implement the FTP protocol using an asynchronous approach Implement a chat application using LINQ and PLINQ to search directories for files matching patternsFind patterns in strings using the System.Linq.Expressions namespaceRandomize arrays and lists with extension methodsUse cryptographic techniques to encrypt and decrypt strings and filesWho this book is for If you want to improve the speed of your code and optimize the performance of your applications, or are simply looking for a practical resource on test driven development, this is the ideal Learning Path for you. Some familiarity with C# and .NET will be beneficial.

Asynchronous, Parallel, and Multithreaded Programming PHP Architect's Guide to PHP Design Patterns For Multicore and Cluster Systems Professional C# 7 and .NET Core 2.0 Introducing Erlang Do you want your .NET code to have the absolute best performance it can? This book describes the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works-it teaches you exactly what you need to do now to bring the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team...and much more.

Discover the world of Rust programming through real-world examples Key Features Implement various features of Rust to build blazingly fast applications Learn to build GUI applications using GTK+ Explore the multi-threading aspect of Rust to tackle problems in concurrency and in distributed environments Book Description Rust is an open source, safe, concurrent, practical language created by Mozilla. It runs blazingly fast, prevents segfaults, and guarantees safety. This book gets you started with essential software development by guiding you through the different aspects of Rust programming. With this approach, you can bridge the gap between learning and implementing immediately. Beginning with an introduction to Rust, you'll learn the basic concepts such as its syntax, data types, functions, generics, control flows, and more. After this, you'll jump straight into building your first project, a Tetris game. Next you'll build a

Over 70 recipes to get you writing powerful and efficient multithreaded, asynchronous, and parallel programs in C# 6.0 About This Book Rewritten and updated to take advantage of the latest C# 6 features Learn about multithreaded, asynchronous, and parallel programming through hands-on, code-first examples Use these recipes to build fast, scalable, and reliable applications in C# Who This Book Is For This book is aimed at those who are new to multithreaded programming, and who are looking for a quick and easy way to get started. It is assumed that you have some experience in C# and .NET already, and you should also be familiar with basic computer science terminology and basic algorithms and data structures. What You Will Learn Use C# 6.0 asynchronous language features Work with raw threads, synchronize threads, and coordinate their work Develop your own asynchronous API with Task Parallel Library Work effectively with a thread pool Scale up your server application with I/O threads Parallelize your LINQ queries with PLINQ Use common concurrent collections Apply different parallel programming patterns Use Reactive Extensions to run asynchronous operations and manage their options In Detail Multi-core processors are synonymous with computing speed and power in today's world, which is why multithreading has become a key concern for C# developers. Multithreaded code helps you create effective, scalable, and responsive applications. This is an easy-to-follow guide that will show you difficult programming problems in context. You will learn how to solve them with practical, hands-on, recipes. With these recipes, you'll be able to start creating your own scalable and reliable multithreaded applications. Starting from learning what a thread is, we guide you through the basics and then move on to more advanced concepts such as task parallel libraries, C# asynchronous functions, and much more. Rewritten to the latest C# specification, C# 6, and updated with new and modern recipes to help you make the most of the hardware you have available, this book will help you push the boundaries of what you thought possible in C#. Style and approach This is an easy-to-follow guide full of hands-on examples of real-world multithreading tasks. Each topic is explained and placed in context, and for the more inquisitive, there are also more in-depth details of the concepts used.

After a dozen years of incremental changes, C# has become one of the most versatile programming languages available. With this comprehensive guide, you'll learn just how powerful the combination of C# 5.0 and .NET 4.5 can be. Author Ian Griffiths guides you through C# 5.0 fundamentals and teaches you techniques for building web and desktop applications, including Windows 8-style apps. Completely rewritten for experienced programmers, this book provides many code examples to help you work with the nuts and bolts of C# code, such as generics, dynamic typing, and the new asynchronous programming features. You'll also get up to speed on XAML, ASP.NET, LINQ, and other .NET tools. Discover how C# supports fundamental coding features such as classes, other custom types, collections, and error handling Understand the differences between dynamic and static typing in C# Query and process diverse data sources such as in-memory object models, databases, and XML documents with LINQ Use .NET's multithreading features to exploit your computer's parallel processing capabilities Learn how the new asynchronous language features can help improve application responsiveness and scalability Use XAML to create Windows 8-style, phone, and classic desktop applications

C# 5 First Look
Design Patterns for Decomposition and Coordination on Multicore Architectures
Unleash the Power of Async
Practical Debugging for .NET Developers
Optimize Your C# Applications
Programming C# 5.0

A practice-oriented guide to using C# to design and program pricing and trading models In this step-by-step guide to software development for financial analysts, traders, developers and quants, the authors show both novice and experienced practitioners how to develop robust and accurate pricing models and employ them in real environments. Traders will learn how to design and implement applications for curve and surface modeling, fixed income products, hedging strategies, plain and exotic option modeling, interest rate options, structured bonds, unfunded structured products, and more. A unique mix of modern software technology and quantitative finance, this book is both timely and practical. The approach is thorough and comprehensive and the authors use a combination of C# language features, design patterns, mathematics and finance to produce efficient and maintainable software. Designed for quant developers, traders and MSc/MFE students, each chapter has numerous exercises and the book is accompanied by a dedicated companion website. <http://www.datasimfinancial.com/forum/viewforum.php?f=196&aid=f30022095850dee48c7db5ff62192b34>, providing all source code, alongside audio, support and discussion forums for readers to comment on the code and obtain new versions of the software.

Design patterns are comprehensive, well-tested solutions to common problems that developers everywhere encounter each day. Although designed for solving general programming issues, some of them have been successfully adapted to the specific needs of Web development.php architect's Guide to PHP Design Patterns is the first comprehensive guide to the application of design patterns to the PHP development language. Designed to satisfy the need of enterprise-strength development, you will find this book an excellent way to learn about design patterns and an irreplaceable reference for your day-to-day programming. With coverage of more than 16 different types of patterns, including Model-View-Controller, Iterator, MockObject, Register, Proxy, ActiveRecord, DataMapper and many, many others, this book is the ideal resource for your enterprise development with PHP 4 and PHP 5.* Includes over 16 design patterns* Each pattern is discussed in detail with practical code applications* Covers both PHP 4 and PHP 5* Provides a thorough test-driven approach to design patterns* Code is available online

Recently developed genomic tools, like SNP-genotyping and whole genome sequencing, and their analysis, offer great opportunities for the conservation and utilisation of animal genetic diversity, both among and within breeds. These genomic tools can be used to detect potentially valuable rare alleles and haplotypes. They are important parts of the genetic diversity we need to conserve now for possible utilisation in the future. This book describes the use of genomic technology to define breeds, to measure diversity and to assess important features in the history of breeds affecting the present genetic diversity. The management of genetic diversity with genomic tools is outlined both in vivo: small populations of rare breeds or large populations with small effective population sizes and in vitro: genebanks. Special attention is given to the genomic management of populations of animals with high incidences of genetic defects. This book is intended for MSc and PhD students, scientists working with small populations in animal breeding and in conservation programmes for rare breeds.

Modern patterns of concurrent and parallel programming
Rust Programming By Example
Mastering Kafka Streams and Ksqldb