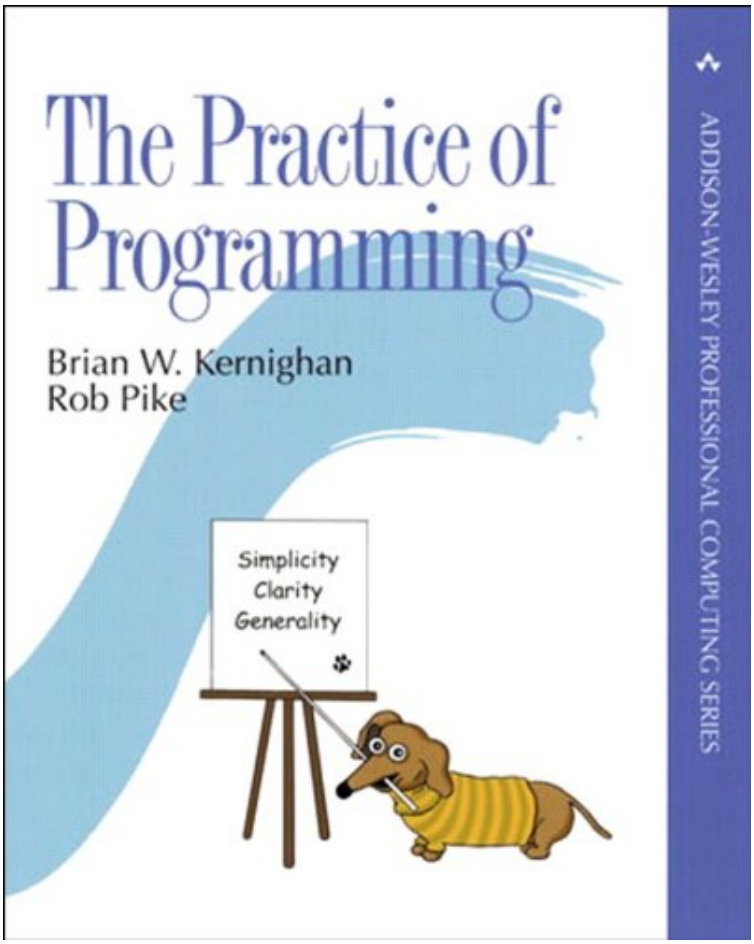
The Practice of Programming (Addison-Wesley Professional Computing Series)

Von Brian W. Kernighan, Rob Pike
*DOC | *audiobook | ebooks | Download PDF | ePub*



Produktinformation -Verkaufsrang: #128932 in eBooksVerffentlicht am: 1999-02-09Erscheinungsdatum: 1999-02-09File Name: B00HU50A12 | File size: 33.Mb

Von Brian W. Kernighan, Rob Pike : The Practice of Programming (Addison-Wesley Professional Computing Series) before purchasing it in order to gage whether or not it would be worth my time, and all praised The Practice of Programming (Addison-Wesley Professional Computing Series):

KundenrezensionenHilfreichste Kundenrezensionen7 von 7 Kunden fanden die folgende Rezension hilfreich. Concise, well written, full of good and long-lasting advice.Von Ein KundeI was surprised by some unfavourable reviews of this book, but they were probably by people expecting something different. This is not a handbook or cookbook. This book is about how to do things wisely... it is about long-lasting principles about how to program in practice. It is not a book about how to program in the latest version of a given compiler and windowing system.I am not a professional programmer, but I do a fair bit of programming and lately software design for our research work. I have also read some of the earlier books by the same authors. Not everything was new to me, but nontheless it was never boring

reading...I do not mind paying for a book full of ideas, and few instructions. This is a book I will not open every day, but having read it has made me a better and wiser programmer every day.4 von 4 Kunden fanden die folgende Rezension hilfreich. A great book if you want to be a professional programmerVon Lars Tafjord (lars.tafjord@siemens.no)There is no doubt, this is a good book. In my opinion it focuses on the right aspects of programming and it does it in a concise and enjoyable way, and even better it does it on about 250 pages only. So, in fact I can manage to read it during hectic working days, not just buy it and decorate my bookshelf with it. I would recommend it to any beginner who wants to turn into a professional programmer. For the professionals who have been in the game for some years, it works very well as a reminder of what we should focus on to stay professionals. However, I feel it lacks something on how to handle really big software systems. But, together with the books by Steve Maguire and The Mythical Man-Month by Frederick Brooks, I would say you have a very good ballast as a professional programmer.1 von 1 Kunden fanden die folgende Rezension hilfreich. No waste of time hereVon Charles AshbacherIn computing, the learning curve is doubly steep. Not only do we have to learn very complex operations, but we have to learn them at a pace unrivaled in any other field. Furthermore, the equipment improves at a rate that simply boggles the mind. In this frantic environment, we rarely have time to read our code twice, much less read a book about code. Therefore, when we do read, we must make every minute count. This is one book where your count of wasted minutes would be a very small one. Some of the tips in this book are obvious in retrospect, yet ones that you probably would not think of. My favorite is the fact that due to the changes in processors, a double precision floating point arithmetic operation can be faster than the equivalent one for integers. In the "old" days, the gospel was that you must avoid floating point operations unless absolutely necessary, to avoid the degradation of performance. Other tips, such as methods to assure you comment what is necessary, taking a few minutes to learn simple performance features, debugging and testing guidelines; portability issues and basic algorithm analysis should cause you to pause for a moment. Even in our hectic development environments, stopping and analyzing your code is a necessity. It is difficult to conceive of someone who will not find a tip in here that will justify the cost of the book. Unless of course, you are one of the authors. I listed it as one of the top books of the year in my On Books column that appeared in the September, 1999 issue of _Journal of Object-Oriented Programming_.

KurzbeschreibungWith the same insight and authority that made their book The Unix Programming Environment a classic, Brian Kernighan and Rob Pike have written The Practice of Programming to help make individual programmers more effective and productive. The practice of programming is more than just writing code. Programmers must also assess tradeoffs, choose among design alternatives, debug and test, improve performance, and maintain software written by themselves and others. At the same time, they must be concerned with issues like compatibility, robustness, and reliability, while meeting specifications. The Practice of Programming covers all these topics, and more. This book is full of practical advice and real-world examples in C, C++, Java, and a variety of special-purpose languages. It includes chapters on: debugging: finding bugs quickly and methodically testing: guaranteeing that software works correctly and reliably performance: making programs faster and more compact portability: ensuring that programs run everywhere without change design: balancing goals and constraints to decide which algorithms and data structures are best interfaces: using abstraction and information hiding to control the interactions between components style: writing code that works well and is a pleasure to read notation: choosing languages and tools that let the machine do more of the work Kernighan and Pike have distilled years of experience writing programs, teaching, and working with other programmers to create this book. Anyone who writes software will profit from the principles and guidance in The Practice of Programming . .deCoauthored by Brian Kernighan, one of the pioneers of the C programming language, The Practice of Programming is a manual of good programming style that will help any C/C++ or Java developer create faster, more maintainable code. Early sections look at some of the pitfalls of C/C++, with numerous real-world excerpts of confusing or incorrect code. The authors offer many tips and solutions, including a guide for variable names and commenting styles. Next, they cover algorithms, such as binary and quick sorting. Here, the authors show how to take advantage of the built-in functions in standard C/C++. When it comes to data structures, such as arrays, linked lists, and trees, the authors compare the options available to C, C++, Java, and even Perl developers with a random-text-generation program (using a sophisticated Markov chain algorithm) written for each language. Subsequent sections cover debugging tips (including how to isolate errors with debugging statements) and testing strategies (both white-box and black-box testing) for verifying the correctness of code. Final sections offer tips on creating more portable C/C++ code, with the last chapter suggesting that programmers can take advantage of interpreters (and regular expressions) to gain better control over their code. A handy appendix summarizes the dozens of tips offered throughout the book. With its commonsense expertise and range of examples drawn from C, C++, and Java, The Practice of Programming is an excellent resource for improving the style and performance of your code base. --Richard DraganKurzbeschreibungWith the same insight and authority that made their book The Unix Programming Environment a classic, Brian Kernighan and Rob Pike have written The Practice of

Programming to help make individual programmers more effective and productive. The practice of programming is more than just writing code. Programmers must also assess tradeoffs, choose among design alternatives, debug and test, improve performance, and maintain software written by themselves and others. At the same time, they must be concerned with issues like compatibility, robustness, and reliability, while meeting specifications. The Practice of Programming covers all these topics, and more. This book is full of practical advice and real-world examples in C, C++, Java, and a variety of special-purpose languages. It includes chapters on: debugging: finding bugs quickly and methodically testing: guaranteeing that software works correctly and reliably performance: making programs faster and more compact portability: ensuring that programs run everywhere without change design: balancing goals and constraints to decide which algorithms and data structures are best interfaces: using abstraction and information hiding to control the interactions between components style: writing code that works well and is a pleasure to read notation: choosing languages and tools that let the machine do more of the work Kernighan and Pike have distilled years of experience writing programs, teaching, and working with other programmers to create this book. Anyone who writes software will profit from the principles and guidance in The Practice of Programming .