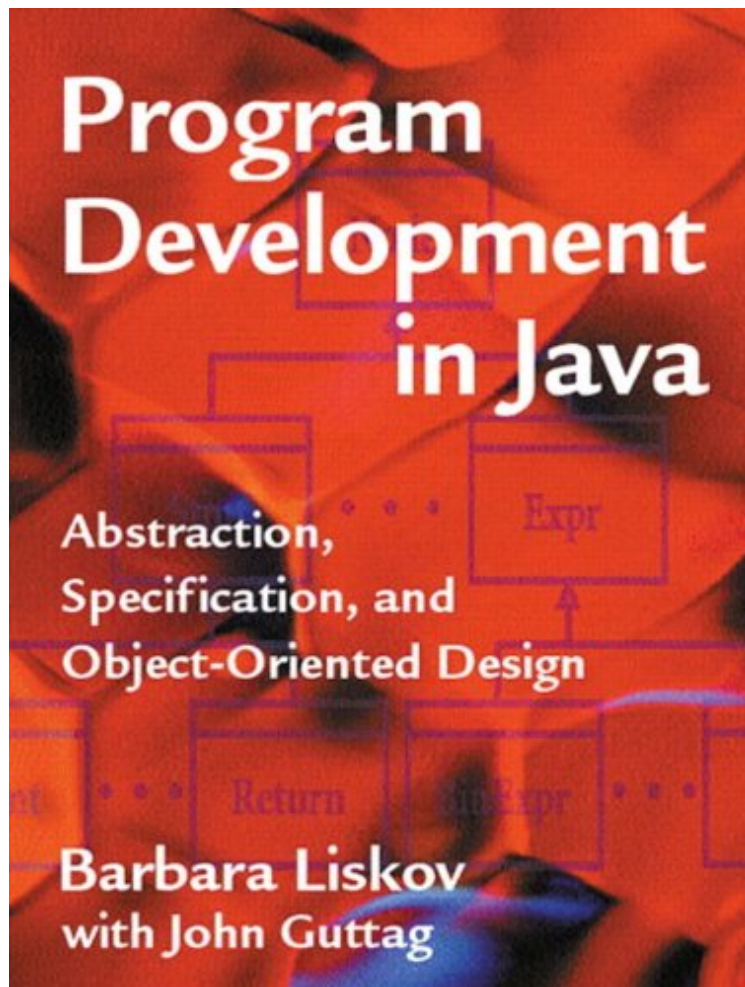


(Free) Program Development in Java: Abstraction, Specification, and Object-Oriented Design

Program Development in Java: Abstraction, Specification, and Object-Oriented Design

Von Barbara Liskov, John Guttag
ebooks | Download PDF | *ePub | DOC | audiobook



 Download

 Read Online

Produktinformation -Verkaufsrank: #792804 in eBooksVerffentlicht am: 2000-06-06Erscheinungsdatum:
2000-06-06File Name: B001FBFHEC | File size: 68.Mb

Von Barbara Liskov, John Guttag : Program Development in Java: Abstraction, Specification, and Object-Oriented Design before purchasing it in order to gage whether or not it would be worth my time, and all praised Program Development in Java: Abstraction, Specification, and Object-Oriented Design:

KundenrezensionenHilfreichste Kundenrezensionen6 von 6 Kunden fanden die folgende Rezension hilfreich.
Excellent OO bookVon John P. BindelEven though "Java" is mentioned in the title of this book, it is not a book on programming in Java. It is much better than that. Plenty of books exist to teach the syntax of Java already; this book provides excellent discussions on designing good programming modules (e.g. classes) that are cohesive units without undesired coupling to other modules in one's design. The discussions of the concepts of mutability, representation-invariants, adequacy, and object specifications are the best I have seen, and Liskov uses these concepts to show how to

build extensible classes. This book also has good information about the whole process of developing software; it does not focus on coding, but it does discuss implementing key concepts using Java as the implementation language. This book is not the Java version of some C++ book, but is an excellent software design and development book updated to use Java instead of a custom language called CLU that was used in the previous version. Those looking for a book on good Java usage should check out Practical Java by Hagggar or something similar. 0 von 0 Kunden fanden die folgende Rezension hilfreich. Excellent OO book Von John P. Bindel Even though "Java" is mentioned in the title of this book, it is not a book on programming in Java. It is much better than that. Plenty of books exist to teach the syntax of Java already; this book provides excellent discussions on designing good programming modules (e.g. classes) that are cohesive units without undesired coupling to other modules in one's design. The discussions of the concepts of mutability, representation-invariants, adequacy, and object specifications are the best I have seen, and Liskov uses these concepts to show how to build extensible classes. This book also has good information about the whole process of developing software; it does not focus on coding, but it does discuss implementing key concepts using Java as the implementation language. This book is not the Java version of some C++ book, but is an excellent software design and development book updated to use Java instead of a custom language called CLU that was used in the previous version. Those looking for a book on good Java usage should check out Practical Java by Hagggar or something similar. 0 von 3 Kunden fanden die folgende Rezension hilfreich. Abstraction Von Ein Kunde I was disappointed by the abstract nature of the book. Is this just an update to the C++ version to use Java? I expected a more concrete presentation but the title does suggest Abstraction. I hesitate to be critical of an expert but it wasn't that useful to me.

Kurzbeschreibung Written by a world-renowned expert on programming methodology, and the winner of the 2008 Turing Award, this book shows how to build production-quality programs--programs that are reliable, easy to maintain, and quick to modify. Its emphasis is on modular program construction: how to get the modules right and how to organize a program as a collection of modules. The book presents a methodology effective for either an individual programmer, who may be writing a small program or a single module in a larger one; or a software engineer, who may be part of a team developing a complex program comprised of many modules. Both audiences will acquire a solid foundation for object-oriented program design and component-based software development from this methodology. Because each module in a program corresponds to an abstraction, such as a collection of documents or a routine to search the collection for documents of interest, the book first explains the kinds of abstractions most useful to programmers: procedures; iteration abstractions; and, most critically, data abstractions. Indeed, the author treats data abstraction as the central paradigm in object-oriented program design and implementation. The author also shows, with numerous examples, how to develop informal specifications that define these abstractions--specifications that describe what the modules do--and then discusses how to implement the modules so that they do what they are supposed to do with acceptable performance. Other topics discussed include: Encapsulation and the need for an implementation to provide the behavior defined by the specification Tradeoffs between simplicity and performance Techniques to help readers of code understand and reason about it, focusing on such properties as rep invariants and abstraction functions Type hierarchy and its use in defining families of related data abstractions Debugging, testing, and requirements analysis Program design as a top-down, iterative process, and design patterns The Java programming language is used for the book's examples. However, the techniques presented are language independent, and an introduction to key Java concepts is included for programmers who may not be familiar with the language.

Kurzbeschreibung Written by a world-renowned expert on programming methodology, and the winner of the 2008 Turing Award, this book shows how to build production-quality programs--programs that are reliable, easy to maintain, and quick to modify. Its emphasis is on modular program construction: how to get the modules right and how to organize a program as a collection of modules. The book presents a methodology effective for either an individual programmer, who may be writing a small program or a single module in a larger one; or a software engineer, who may be part of a team developing a complex program comprised of many modules. Both audiences will acquire a solid foundation for object-oriented program design and component-based software development from this methodology. Because each module in a program corresponds to an abstraction, such as a collection of documents or a routine to search the collection for documents of interest, the book first explains the kinds of abstractions most useful to programmers: procedures; iteration abstractions; and, most critically, data abstractions. Indeed, the author treats data abstraction as the central paradigm in object-oriented program design and implementation. The author also shows, with numerous examples, how to develop informal specifications that define these abstractions--specifications that describe what the modules do--and then discusses how to implement the modules so that they do what they are supposed to do with acceptable performance. Other topics discussed include: Encapsulation and the need for an implementation to provide the behavior defined by the specification Tradeoffs between simplicity and performance Techniques to help readers of code understand and reason about it, focusing on such properties as rep invariants and abstraction functions Type hierarchy and its use in defining families of related data abstractions Debugging, testing,

and requirements analysis Program design as a top-down, iterative process, and design patterns The Java programming language is used for the book's examples. However, the techniques presented are language independent, and an introduction to key Java concepts is included for programmers who may not be familiar with the language. Synopsis Of all the books that have sought to explain object-oriented, component-based development, this one is unique in its clarity. MIT Professor Barbara Liskov presents a start-to-finish methodology for constructing reliable, understandable, easy-to-maintain Java software. Developers will learn exactly how to decompose a programming problem into its key elements, and then build an optimal program from independent modules -- crucial for success in large-scale, team development projects. Liskov masterfully introduces key concepts of abstraction, specification, and object-oriented design in the context of small programs; then scales the techniques up to enterprise-class development, where they work equally well. The book includes detailed coverage of developing and writing accurate specifications; optimizing tradeoffs between simplicity and performance; and making code easier to evaluate, debug, and test. In the second half of the book, Liskov focuses on modular development of large-scale programs: discovering useful abstractions, program design, analysis, implementation, testing, and the value of design patterns. All examples are based on Java (though the concepts are equally applicable to any object-based language).