THIS IS A
USED BOOK

This book was originally distributed as a sample copy by the publisher, for academic review. It was (then) purchased by a used book dealer and resold as used. This allows you a substantial savings. All the chapters and pages are included.

SOFTWARE ENGINEERING WITH Ada®

SECOND EDITION

Grady Booch

PRESENTED BY:
BENJAMIN/CUMMINGS PUBLISHING COMPANY

TO: CAL ST UNIV-LOS ANGELES
COMPUTER SCIENCE DEPT
ID# 0402861-370-07

TP31
B724
E.2

9760655

SOFTWARE ENGINEERING WITH Ada®

SECOND EDITION

Grady Booch
Rational

The Benjamin/Cummings Publishing Company, Inc.
Menlo Park, California • Reading, Massachusetts
Don Mills, Ontario • Wokingham, U.K. • Amsterdam • Sydney
Singapore • Tokyo • Madrid • Bogota • Santiago • San Juan

To Jan

Sponsoring Editor: Alan Apt
Production: Merry Finley, Betsy Dilernia
Copy Editor: Toni Murray
Interior Design: R. Kharibian, Marilyn Langfeld
Cover Art and Design: R. Kharibian
Illustration: Renaissance Studios, Marilyn Langfeld
Composition: Interactive Composition Corporation

Copyright © 1983, 1987 by the Benjamin/Cummings Publishing Company, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

The programs presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs.

"Ada" is a registered trademark of the U.S. Government, Ada Joint Program Office. Appendices C, D, and E and the Glossary are taken, with permission of the Ada Joint Program Office, Office of the Undersecretary of Defense for Research and Engineering, U.S. Department of Defense, from the "Reference Manual for the Ada® Programming Language" (1983), and include additional text as indicated.

Library of Congress Cataloging in Publication Data

Booch, Grady.
Software engineering with Ada.

Bibliography: p.
Includes index.
1. Ada (Computer program language) 2. Electronic digital computers—Programming. I. Title.
QA76.73.A35B66 1987 005.13'3 86-12955
ISBN 0-8053-0604-8

ABCDEFGHIJK-DO-89876

The Benjamin/Cummings Publishing Company, Inc.
2727 Sand Hill Road
Menlo Park, CA 94025

贈閱

SOFTWARE ENGINEERING WITH Ada®

SECOND EDITION

Grady Booch

The Benjamin / Cummings Series in Ada and Software Engineering

Grady Booch, Series Editor

Booch, **Software Engineering with Ada, Second Edition (1986)**

Forthcoming Titles

Booch, **Software Components with Ada: Structures, Tools, and Subsystems (1987)**

Brandon, **Introduction to Ada (1988)**

EVB, Inc., **Object Oriented Design Handbook (1987)**

Other Titles of Interest

Conte/Dunsmore/Shen, **Software Engineering: Metrics and Models (1986)**

DeMillo/McCracken(Martin/Passafiume, **Software Engineering: Testing and Evaluation (1987)**

Kelley/Pohl, **A Book on C (1984)**

Kelley/Pohl, **C by Dissection: Essentials of C Programming (1987)**

Kerschberg, **Expert Database Systems (1986)**

Sobell, **A Practical Guide to Unix System V (1985)**

此为试读,需要完整PDF请访问: www.ertongbook.com

FOREWORD

LEARNING TO PROGRAM IN Ada should be viewed as an exciting opportunity because Ada is more than just another programming language. It is the basis for a modern perspective not only of programming but also of software engineering. Ada is a programming *system* from which a new software culture is evolving.

The effect of computers on society has been compared in importance to the Industrial Revolution. They have already had a pronounced effect on the administration of government and industry and, consequently, upon the individual. In the decade of the 1980s, computer technology is changing the fabric of modern society. The ability to put enormous computing power on a chip has resulted in the introduction of computers into consumer products such as automobiles, microwave ovens, washing machines, watches, and home entertainment systems. We are witnessing the introduction of the personal computer, not only for the hobbyist, but also for the individual who wishes to manage finances, control heating and air-conditioning systems, and even manage the family's activities.

All of these computers need to be programmed, and the proliferation of computers increases the already burgeoning demand for software. In this decade, software is going to be one of the most critical technologies in our society. We cannot expect to simply write programs. We must be prepared to engineer software for our systems.

In the late sixties and throughout the seventies, a number of computer professionals pioneered the concepts of software engineering. They began to preach the gospel of careful design, structured coding, and software life-cycle management. New techniques, language features, and software tools were investigated in an attempt to improve the quality of software and improve the productivity of those involved in the process of software development.

As early as 1975, a few far-sighted individuals in the United States Department of Defense recognized the impending software challenge and realized that an effective software development system should be founded on a

standard programming language with features that would encourage modern programming practices. They defined the requirements, recognized that no existing languages would support such requirements, and hence commissioned the worldwide language design competition that produced Ada.

Throughout its development, the Ada program has enjoyed the benefit of advice and assistance from an enormous number of the world's premiere computer scientists and software engineers. Attracted by the opportunity to create a language that could support the development of more sophisticated software practices, many people freely gave their advice and experience. Even the critics sought to be helpful in their criticism.

The result is a language good enough to capture the interest of a substantial portion of the computing community, a language that has been called "the language of the eighties." Even so, it is not perfect, nor is it the ultimate in programming languages. It is the result of an engineering effort in which compromises were made of necessity. In making them, however, the principles of software engineering were not forsaken. Designed for software portability and reuse, Ada promises to support the development of a software components industry. The encapsulation facilities and data abstraction capabilities of the Ada package will facilitate the development of applications libraries and other forms of reusable software.

This book captures much of the software engineering aspect of Ada. It offers a consistent approach to design and offers advice for the development of an appropriate style. The author is well versed in the presentation of Ada: He has taught a number of short courses for the Department of Defense and, through extensive experimentation, has develped an understanding of how best to present the material. The result is a well-balanced guide not only of the language but, more importantly, to the proper use of it.

If you approach the study of Ada simply as an exercise in learning the syntax of a new language, you will be terribly disappointed. The language will appear complex, and you will very likely produce poorly constructed programs. However, if you approach your study with the intention of learning the language as a vehicle for gaining an understanding of modern software engineering, you will be pleased to find that it is simpler than its size suggests. You will also develop the ability to express your designs coherently. This book will guide you in that endeavor, providing motivation for the language features and demonstrating their proper use.

Larry E. Druffel
Arlington, Virginia
January 1983

PREFACE

Ada IS A GENERAL-PURPOSE programming language with considerable expressive power. It was developed at the initiative of the United States Department of Defense in response to the crisis in software development. Ada was designed specifically for the domain of large, real-time, embedded computer systems, although it will certainly have an impact on many other application areas.

Unlike most other production high-order languages, such as FORTRAN, COBOL, or even Pascal, Ada not only embodies many modern software development principles but also enforces them. The greatest benefits in this common high-order language effort will thus be gained from the application of good software development methods, which are facilitated by using Ada as the language of expression. As a result, the introduction of Ada represents a tremendous opportunity for improvement in the clarity, reliability, efficiency, and maintainability of software systems.

Ada is more than just another programming language, however. Along with the Ada Programming Support Environment, it represents a very powerful facility that helps us to understand problems and express their solutions in a manner that directly reflects the multidimensional real world.

Goals

This book is not just another introduction to Ada. It has been written to satisfy the following three specific goals:

- To provide an intensive study of Ada's features.
- To motivate and give examples of good Ada design and programming style.

Content Features

Structure

Many texts present the details of a programming language only from a syntactic or semantic perspective. In this book, I have instead chosen to start with a software design approach and then introduce Ada from the top down in the context of good programming methods. This model reflects the recommendation of the Education Panel and the Association for Computing Machinery Special Interest Group for Programming Languages (ACM SIGPLAN) Symposium on Ada, held during December 1980 in Boston, and the philosophy of the AJPO Strategy for Ada Education and Training.

The book is divided into eight packages, each of which contains three chapters that are logically related. The first package begins with a look at the Ada problem domain. It includes an examination of Ada's development history in order to provide a perspective on some of the features of the language. In the second package, a number of modern software development principles are examined and the object-oriented development method is introduced.

In the third through seventh packages, a detailed presentaion of Ada as an embodiment of these principles is provided, built around five complete design examples. Each problem is increasingly more complex, and together they require the application of almost every Ada feature. In addition, these problems provide a vehicle for demonstrating the object-oriented development method, along with a programming style that emphasizes understandability. In the chapters between these five large examples, a detailed discussion of Ada's constructs is presented. The eighth package examines the Ada Programming Support Environment, plus the application of Ada across the software life cycle.

Resources

At the end of most of the chapters, I have provided a set of exercises for the student. Difficult problems are marked with a star (★). In addition, the book concludes with six appendices that provide further technical details of Ada: The first two contain complete syntax charts and a summary style guide; the next three describe the predefined elements of the language; the last one provides the solutions of each design problem. I also provide a glossary of Ada terms and an extensive list of additional references.

Course Organization

This is a "generic" book in the sense that it can be instantiated at a number of levels. I have taught this material in a one-semester course (40 one-hour lessons) and as a five-day seminar using the following outline:

Lesson 1	Chapter 1:	Introduction
Lesson 2	Chapter 2:	The Software Crisis
Lessons 3–4	Chapter 3:	The History of Ada's Development
Lessons 5–8	Chapter 4:	Software Engineering
Lessons 9–10	Chapter 5:	Object-Oriented Development
Lessons 11–12	Chapter 6:	An Overview of the Language
Lesson 13	Chapter 7:	The First Problem: Document Concordance
Lessons 14–15	Chapter 8:	Data Abstraction and Ada's Types
Lesson 16	Chapter 9:	The Second Design Problem: Data Base System
Lessons 17–18	Chapter 10:	Subprograms
Lessons 19–20	Chapter 11:	Expressions and Statements
Lesson 21	Chapter 12:	The Second Design Problem: Continued
Lessons 22–23	Chapter 13:	Packages
Lessons 24–25	Chapter 14:	Generic Program Units
Lesson 26	Chapter 15:	The Third Design Problem: Generic Tree Package
Lessons 27–28	Chapter 16:	Tasks
Lessons 29–30	Chapter 17:	Exception Handling and Low-Level Features
Lesson 31	Chapter 18:	The Fourth Design Problem: Environment Monitoring
Lesson 32	Chapter 19:	Input/Output
Lessons 33–34	Chapter 20:	Programming in the Large
Lesson 35	Chapter 21:	The Fifth Design Problem: Heads-Up Display
Lessons 36–37	Chapter 22:	The Ada Programming Support Environment
Lessons 38–39	Chapter 23:	The Software Life Cycle with Ada
Lesson 40	Chapter 24:	Trends and Conclusion

In addition, the following structure is appropriate as a brief introduction to the application of Ada for program managers:

Block 1	Chapter 2:	The Software Crisis
	Chapter 3:	The History of Ada's Development
Block 2	Chapter 4:	Software Engineering
	Chapter 23:	The Software Life Cycle With Ada
Block 3	Chapter 6:	An Overview of the Language
Block 4	Chapter 21:	The Fifth Design Problem: Heads-Up Display
	Chapter 22:	The Ada Programming Support Environment
	Chapter 24:	Trends and Conclusion

Acknowledgments

I wish to thank a number of people who helped me during the preparation of this manuscript. In particular, Dick Bolz and Larry Schwartz both reviewed this work during its development and provided countless helpful suggestions. My friends and Air Force Academy classmates Mike Devlin and Paul Levy, both of Rational, gave me many hours of stimulating discussion on the technical and managerial issues of Ada; Mike also contributed to the design of my first Ada course.

Larry Druffel and Vance Mall of the Ada Joint Program Office provided an environment in which I could examine the issues of Ada education, along with the chance to teach the language across the country. For this opportunity to be a part of the process, I am deeply indebted.

Many other people provided comments that influenced the thoughts in this book, especially Russ Abbott, Lucie Bennett, Ken Bowles, Doug Bryan, Bill Carlson, Mark Feldman, John Goodenough, Hal Hart, Richard Kaufmann, Nico Lomuto, Bob Mathis, Mark Sadler, Keith Schillington, Tim Standish, Peter Wegner, and Bill Whitaker, plus the other reviewers listed at the end of this preface. In addition, I received very helpful criticism from the students in my Ada courses—criticism that enabled me to refine the organization and presentation of this material.

A very special thanks goes to Sam Harbrough, who provided many useful improvements for the second edition. Additionally, I wish to thank Dick Bolz for his feedback; I have grown much from my contact with him.

I also wish to thank my editor, Alan Apt, for his constant support and guidance. Most of all, I thank my wife for her patience and encouragement during the life of this project.

Reviewers

Russell Abbott
California State University
and The Aerospace Corporation

Cheryl Allen
Control Data Corporation

Christine Ausnett
SofTech, Inc.

Richard Bolz
United States Air Force Academy

Ben Brosgol
Alsys

此为试读,需要完整PDF请访问: www.ertongbook.com

Kenneth Bowles
TeleSoft, Inc.

Doug Bryan
Stanford University

Larry Druffel
Ada Joint Program Office

Michael Feldman
George Washington University

Gerry Fisher
IBM Research

John Foreman
Texas Instruments

John Goodenough
SofTech, Inc.

Samuel S. Harbaugh
Consultant

Hal Hart
TRW

Charles McKay
University of Houston

Elliott Organick
University of Utah

Carol Righini
GTE Sylvania

Bryan Scharr
Consultant

James Schnelker
General Dynamics

Larry Schwartz
International Business Machines, Inc.

Sally Shepherd
Texas A&M University

John Warner
Consultant

CONTENTS

THE 2nd PACKAGE

THE 3rd PACKAGE

THE 5th PACKAGE

此为试读,需要完整PDF请访问: www.ertongbook.com