

# THE Elements OF Java<sup>TM</sup> Style

---

Allan Vermeulen • Scott W. Ambler  
Greg Bumgardner • Eldon Metz • Trevor Misfeldt  
Jim Shur • Patrick Thompson

SIGS REFERENCE LIBRARY

 **Rogue Wave**  
SOFTWARE

# *The Elements* *of* **Java™** *Style*



# The Elements of Java™ Style

Al Vermeulen  
Scott W. Ambler  
Greg Burgardner  
Eldon Metz  
Trevor Mischelt  
Jim Shur  
Patrick Thompson

## SIGS Reference Library

1. Object Methodology Overview CD-ROM • *Doug Rosenberg*
2. Directory of Object Technology • *edited by Dale J. Gammier*
3. Dictionary of Object Technology: The Definitive Desk Reference •  
*Donald G. Firesmith and Edward M. Eykholt*
4. Next Generation Computing: Distributed Objects for Business •  
*edited by Peter Finger, Dennis Read, and Jim Skeelisher*
5. C++ Gems • *edited by Stanley B. Lippman*
6. OMT Insights: Perspectives on Modeling from the Journal of  
Object-Oriented Programming • *James Rumbaugh*
7. Best of Booch: Designing Strategies for Object Technology •  
*Grady Booch (Edited by Ed Eykholt)*
8. Wisdom of the Gurus: A Vision for Object Technology •  
*selected and edited by Charles F. Bouman*
9. Open Modeling Language (OML) Reference Manual •  
*Donald Firesmith, Brian Henderson-Sellers, and Ian Graham*
10. Java™ Gems: Jewels from Java™ Report •  
*collected and introduced by Dwight Dango, Ph.D*
11. The Netscape Programmer's Guide: Using OLE to Build  
Componentware™ Apps • *Richard B. Lam*
12. Advanced Object-Oriented Analysis and Design  
Using UML • *James J. Odell*
13. The Patterns Handbook: Techniques, Strategies, and  
Applications • *edited by Linda Ring*
14. Kent Beck's Guide to Better Smalltalk:  
A Sorted Collection • *Kent Beck*
15. The Elements of Java™ Style • *Al Vermeulen, et al*
16. More Java™ Gems • *edited by Dwight Dango, Ph.D*
17. More C++ Gems • *edited by Robert C. Martin*

Additional Volumes in Preparation



# Introduction

- style*: 1b. the shadow-producing pin of a sundial.  
2c. the custom or plan followed in spelling, capitalization, punctuation, and typographic arrangement and display.

—*Webster's New Collegiate Dictionary*

THE SYNTAX OF A PROGRAMMING LANGUAGE tells you what code it is possible to write—what the machine will understand. *Style* tells you what you ought to write—what the humans reading the code will understand. Code written with a consistent, simple style will be maintainable, robust, and contain fewer bugs. Code written with no regard to style will contain more bugs. It may simply be thrown away and rewritten rather than maintained.

Our two favorite style guides are classics: Strunk and White's *The Elements of Style* and Kernighan and Plauger's *The Elements of Programming Style*. These small books work because they are simple—a list of rules, each containing a brief explanation and examples of correct, and sometimes incorrect, use. We followed the same pattern in this book.

This simple treatment—a series of rules—enabled us to keep this book short and easy to understand. The idea is to provide a clear standard to follow, so programmers can spend their time on solving the problems of their customers, instead of worrying about naming conventions and formatting.