The Java™ Virtual Machine Specification

The Java™ Series

Lisa Friendly, Series Editor
Bill Joy, Technical Advisor

The Java™ Programming Language
Ken Arnold and James Gosling
ISBN 0-201-63455-4

The Java™ Language Specification
James Gosling, Bill Joy, and Guy Steele
ISBN 0-201-63451-1

The Java™ Virtual Machine Specification
Tim Lindholm and Frank Yellin
ISBN 0-201-63452-X

The Java™ Application Programming Interface,
Volume 1: Core Packages
James Gosling, Frank Yellin, and the Java Team
ISBN 0-201-63453-8

The Java™ Application Programming Interface,
Volume 2: Window Toolkit and Applets
James Gosling, Frank Yellin, and the Java Team
ISBN 0-201-63459-7

The Java™ Tutorial: Object-Oriented Programming for the Internet
Mary Campione and Kathy Walrath
ISBN 0-201-63454-6

The Java™ Class Libraries: An Annotated Reference
Patrick Chan and Rosanna Lee
ISBN 0-201-63458-9

The Java™ FAQ: Frequently Asked Questions
Jonni Kanerva
ISBN 0-201-63456-2

The Java™ Virtual Machine Specification

Tim Lindholm
Frank Yellin

ADDISON-WESLEY

An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney
Bonn • Amsterdam • Tokyo • Mexico City

Copyright © 1997 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.
All rights reserved.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The release described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Sun Microsystems, Inc. (SUN) hereby grants to you a fully-paid, nonexclusive, nontransferable, perpetual, worldwide limited license (without the right to sublicense) under SUN's intellectual property rights that are essential to practice this specification. This license allows and is limited to the creation and distribution of clean room implementations of this specification that (i) include a complete implementation of the current version of this specification without subsetting or supersetting, (ii) implement all the interfaces and functionality of the standard `java.*` packages as defined by SUN, without subsetting or supersetting, (iii) do not add any additional packages, classes or methods to the `java.*` packages (iv) pass all test suites relating to the most recent published version of this specification that are available from SUN six (6) months prior to any beta release of the clean room implementation or upgrade thereto, (v) do not derive from SUN source code or binary materials, and (vi) do not include any SUN binary materials without an appropriate and separate license from SUN.

Sun, Sun Microsystems, Sun Microsystems Computer Corporation, the Sun logo, the Sun Microsystems Computer Corporation logo, Java, JavaSoft, JavaScript and HotJava are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX® is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. All other product names mentioned herein are the trademarks of their respective owners.

THIS PUBLICATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Text printed on recycled and acid-free paper

ISBN 0-201-63452-X
1 2 3 4 5 6 7 8 9-MA-00999897
First printing, September1996

To Lucy, Beatrice, and Arnold —*TL*

To Mark —*FY*

Table of Contents

Preface xv

1 Introduction 1

2 Java Concepts 5

2.1 Unicode 5
2.2 Identifiers 6
2.3 Literals 6
2.4 Types and Values 6
2.4.1 Primitive Types and Values 7
2.4.2 Operators on Integral Values 8
2.4.3 Operators on Floating-Point Values 8
2.4.4 Operators on `boolean` Values 9
2.4.5 Reference Types, Objects, and Reference Values 9
2.4.6 The Class `Object` 10
2.4.7 The Class `String` 10
2.4.8 Operators on Objects 10
2.5 Variables 10
2.5.1 Initial Values of Variables 12
2.5.2 Variables Have Types, Objects Have Classes 13
2.6 Conversions and Promotions 13
2.6.1 Identity Conversions 14
2.6.2 Widening Primitive Conversions 15
2.6.3 Narrowing Primitive Conversions 15
2.6.4 Widening Reference Conversions 16
2.6.5 Narrowing Reference Conversions 16
2.6.6 Assignment Conversion 17
2.6.7 Method Invocation Conversion 18
2.6.8 Casting Conversions 19
2.6.9 Numeric Promotion 19
2.7 Names and Packages 20
2.7.1 Names 20
2.7.2 Packages 20
2.7.3 Members 21
2.7.4 Package Members 21
2.7.5 The Members of a Class Type 21
2.7.6 The Members of an Interface Type 22
2.7.7 The Members of an Array Type 22

2.7.8 Qualified Names and Access Control . . . 22
2.7.9 Fully Qualified Names . . . 23
2.8 Classes . . . 24
2.8.1 Class Names . . . 24
2.8.2 Class Modifiers . . . 24
2.8.3 Superclasses and Subclasses . . . 25
2.8.4 The Class Members . . . 25
2.9 Fields . . . 26
2.9.1 Field Modifiers . . . 26
2.9.2 Initialization of Fields . . . 27
2.10 Methods . . . 27
2.10.1 Formal Parameters . . . 28
2.10.2 Signature . . . 28
2.10.3 Method Modifiers . . . 28
2.11 Static Initializers . . . 29
2.12 Constructors . . . 29
2.13 Interfaces . . . 30
2.13.1 Interface Modifiers . . . 31
2.13.2 Superinterfaces . . . 31
2.13.3 Interface Members . . . 31
2.13.4 Interface (Constant) Fields . . . 31
2.13.5 Interface (Abstract) Methods . . . 32
2.13.6 Overriding, Inheritance, and Overloading in Interfaces . . . 32
2.14 Arrays . . . 32
2.14.1 Array Types . . . 33
2.14.2 Array Variables . . . 33
2.14.3 Array Creation . . . 34
2.14.4 Array Access . . . 34
2.15 Exceptions . . . 34
2.15.1 The Causes of Exceptions . . . 35
2.15.2 Handling an Exception . . . 36
2.15.3 The Exception Hierarchy . . . 38
2.15.4 The Classes `Exception` and `RuntimeException` . . . 38
2.16 Execution . . . 40
2.16.1 Virtual Machine Start-up . . . 40
2.16.2 Loading . . . 43
2.16.3 Linking: Verification, Preparation, and Resolution . . . 43
2.16.4 Initialization . . . 46
2.16.5 Detailed Initialization Procedure . . . 47
2.16.6 Creation of New Class Instances . . . 49
2.16.7 Finalization of Class Instances . . . 51
2.16.8 Finalization and Unloading of Classes and Interfaces . . . 52
2.16.9 Virtual Machine Exit . . . 52
2.17 Threads . . . 53

3 Structure of the Java Virtual Machine . . . 57

3.1 Data Types . . . 57
3.2 Primitive Types and Values . . . 58
3.2.1 Integral Types and Values . . . 58
3.2.2 Floating-Point Types and Values . . . 59
3.2.3 The `returnAddress` Type and Values . . . 60
3.2.4 There Is No `boolean` Type . . . 60

3.3 Reference Types and Values 61
3.4 Words 61
3.5 Runtime Data Areas 61
3.5.1 The pc Register 61
3.5.2 Java Stack 62
3.5.3 Heap 63
3.5.4 Method Area 63
3.5.5 Constant Pool 64
3.5.6 Native Method Stacks 65
3.6 Frames 66
3.6.1 Local Variables 66
3.6.2 Operand Stacks 67
3.6.3 Dynamic Linking 67
3.6.4 Normal Method Completion 68
3.6.5 Abnormal Method Completion 68
3.6.6 Additional Information 68
3.7 Representation of Objects 69
3.8 Special Initialization Methods 69
3.9 Exceptions 69
3.10 The `class` File Format 70
3.11 Instruction Set Summary 71
3.11.1 Types and the Java Virtual Machine 72
3.11.2 Load and Store Instructions 74
3.11.3 Arithmetic Instructions 75
3.11.4 Type Conversion Instructions 77
3.11.5 Object Creation and Manipulation 79
3.11.6 Operand Stack Management Instructions 79
3.11.7 Control Transfer Instructions 80
3.11.8 Method Invocation and Return Instructions 80
3.11.9 Throwing and Handling Exceptions 81
3.11.10 Implementing `finally` 81
3.11.11 Synchronization 81
3.12 Public Design, Private Implementation 81

4 The `class` File Format 83

4.1 `ClassFile` 84
4.2 Internal Form of Fully Qualified Class Names 89
4.3 Descriptors 89
4.3.1 Grammar Notation 90
4.3.2 Field Descriptors 90
4.3.3 Method Descriptors 91
4.4 Constant Pool 92
4.4.1 `CONSTANT_Class` 93
4.4.2 `CONSTANT_Fieldref`, `CONSTANT_Methodref`, and `CONSTANT_InterfaceMethodref` 94
4.4.3 `CONSTANT_String` 96
4.4.4 `CONSTANT_Integer` and `CONSTANT_Float` 96
4.4.5 `CONSTANT_Long` and `CONSTANT_Double` 97
4.4.6 `CONSTANT_NameAndType` 99
4.4.7 `CONSTANT_Utf8` 100
4.5 Fields 101
4.6 Methods 104