

The Java[®] Virtual Machine Specification

Java SE 8 Edition

Tim Lindholm
Frank Yellin
Gilad Bracha
Alex Buckley

2015-02-13

Specification: JSR-337 Java® SE 8 Release Contents ("Specification")

Version: 8

Status: Maintenance Release

Release: March 2015

Copyright © 1997, 2015, Oracle America, Inc. and/or its affiliates.

500 Oracle Parkway, Redwood City, California 94065, U.S.A.

All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

The Specification provided herein is provided to you only under the Limited License Grant included herein as Appendix A. Please see Appendix A, *Limited License Grant*.

To Sophia and Susan, in deepest appreciation.

Table of Contents

Preface to the Java SE 8 Edition xiii

1 Introduction 1

- 1.1 A Bit of History 1
- 1.2 The Java Virtual Machine 2
- 1.3 Organization of the Specification 3
- 1.4 Notation 4
- 1.5 Feedback 4

2 The Structure of the Java Virtual Machine 5

- 2.1 The `class` File Format 5
- 2.2 Data Types 6
- 2.3 Primitive Types and Values 6
 - 2.3.1 Integral Types and Values 7
 - 2.3.2 Floating-Point Types, Value Sets, and Values 8
 - 2.3.3 The `returnAddress` Type and Values 10
 - 2.3.4 The `boolean` Type 10
- 2.4 Reference Types and Values 11
- 2.5 Run-Time Data Areas 11
 - 2.5.1 The `pc` Register 12
 - 2.5.2 Java Virtual Machine Stacks 12
 - 2.5.3 Heap 13
 - 2.5.4 Method Area 13
 - 2.5.5 Run-Time Constant Pool 14
 - 2.5.6 Native Method Stacks 14
- 2.6 Frames 15
 - 2.6.1 Local Variables 16
 - 2.6.2 Operand Stacks 17
 - 2.6.3 Dynamic Linking 18
 - 2.6.4 Normal Method Invocation Completion 18
 - 2.6.5 Abrupt Method Invocation Completion 18
- 2.7 Representation of Objects 19
- 2.8 Floating-Point Arithmetic 19
 - 2.8.1 Java Virtual Machine Floating-Point Arithmetic and IEEE 754 19
 - 2.8.2 Floating-Point Modes 20
 - 2.8.3 Value Set Conversion 20
- 2.9 Special Methods 22
- 2.10 Exceptions 23
- 2.11 Instruction Set Summary 25

- 2.11.1 Types and the Java Virtual Machine 26
- 2.11.2 Load and Store Instructions 29
- 2.11.3 Arithmetic Instructions 30
- 2.11.4 Type Conversion Instructions 32
- 2.11.5 Object Creation and Manipulation 34
- 2.11.6 Operand Stack Management Instructions 34
- 2.11.7 Control Transfer Instructions 34
- 2.11.8 Method Invocation and Return Instructions 35
- 2.11.9 Throwing Exceptions 36
- 2.11.10 Synchronization 36
- 2.12 Class Libraries 37
- 2.13 Public Design, Private Implementation 37

3 Compiling for the Java Virtual Machine 39

- 3.1 Format of Examples 39
- 3.2 Use of Constants, Local Variables, and Control Constructs 40
- 3.3 Arithmetic 45
- 3.4 Accessing the Run-Time Constant Pool 46
- 3.5 More Control Examples 47
- 3.6 Receiving Arguments 50
- 3.7 Invoking Methods 51
- 3.8 Working with Class Instances 53
- 3.9 Arrays 55
- 3.10 Compiling Switches 57
- 3.11 Operations on the Operand Stack 59
- 3.12 Throwing and Handling Exceptions 60
- 3.13 Compiling `finally` 63
- 3.14 Synchronization 66
- 3.15 Annotations 67

4 The `class` File Format 69

- 4.1 The `ClassFile` Structure 70
- 4.2 The Internal Form of Names 74
 - 4.2.1 Binary Class and Interface Names 74
 - 4.2.2 Unqualified Names 75
- 4.3 Descriptors 75
 - 4.3.1 Grammar Notation 75
 - 4.3.2 Field Descriptors 76
 - 4.3.3 Method Descriptors 77
- 4.4 The Constant Pool 78
 - 4.4.1 The `CONSTANT_Class_info` Structure 79
 - 4.4.2 The `CONSTANT_Fieldref_info`, `CONSTANT_Methodref_info`, and `CONSTANT_InterfaceMethodref_info` Structures 80
 - 4.4.3 The `CONSTANT_String_info` Structure 81
 - 4.4.4 The `CONSTANT_Integer_info` and `CONSTANT_Float_info` Structures 82

4.4.5	The <code>CONSTANT_Long_info</code> and <code>CONSTANT_Double_info</code> Structures	83
4.4.6	The <code>CONSTANT_NameAndType_info</code> Structure	85
4.4.7	The <code>CONSTANT_Utf8_info</code> Structure	85
4.4.8	The <code>CONSTANT_MethodHandle_info</code> Structure	87
4.4.9	The <code>CONSTANT_MethodType_info</code> Structure	89
4.4.10	The <code>CONSTANT_InvokeDynamic_info</code> Structure	89
4.5	Fields	90
4.6	Methods	92
4.7	Attributes	95
4.7.1	Defining and Naming New Attributes	101
4.7.2	The <code>ConstantValue</code> Attribute	101
4.7.3	The <code>Code</code> Attribute	102
4.7.4	The <code>StackMapTable</code> Attribute	106
4.7.5	The <code>Exceptions</code> Attribute	113
4.7.6	The <code>InnerClasses</code> Attribute	114
4.7.7	The <code>EnclosingMethod</code> Attribute	117
4.7.8	The <code>Synthetic</code> Attribute	118
4.7.9	The <code>Signature</code> Attribute	119
4.7.9.1	Signatures	120
4.7.10	The <code>SourceFile</code> Attribute	123
4.7.11	The <code>SourceDebugExtension</code> Attribute	124
4.7.12	The <code>LineNumberTable</code> Attribute	125
4.7.13	The <code>LocalVariableTable</code> Attribute	126
4.7.14	The <code>LocalVariableTypeTable</code> Attribute	128
4.7.15	The <code>Deprecated</code> Attribute	130
4.7.16	The <code>RuntimeVisibleAnnotations</code> Attribute	131
4.7.16.1	The <code>element_value</code> structure	132
4.7.17	The <code>RuntimeInvisibleAnnotations</code> Attribute	135
4.7.18	The <code>RuntimeVisibleParameterAnnotations</code> Attribute	136
4.7.19	The <code>RuntimeInvisibleParameterAnnotations</code> Attribute	138
4.7.20	The <code>RuntimeVisibleTypeAnnotations</code> Attribute	139
4.7.20.1	The <code>target_info</code> union	145
4.7.20.2	The <code>type_path</code> structure	149
4.7.21	The <code>RuntimeInvisibleTypeAnnotations</code> Attribute	153
4.7.22	The <code>AnnotationDefault</code> Attribute	154
4.7.23	The <code>BootstrapMethods</code> Attribute	155
4.7.24	The <code>MethodParameters</code> Attribute	157
4.8	Format Checking	159
4.9	Constraints on Java Virtual Machine Code	160
4.9.1	Static Constraints	160
4.9.2	Structural Constraints	164
4.10	Verification of <code>class</code> Files	167
4.10.1	Verification by Type Checking	168
4.10.1.1	Accessors for Java Virtual Machine Artifacts	171
4.10.1.2	Verification Type System	175
4.10.1.3	Instruction Representation	178
4.10.1.4	Stack Map Frame Representation	180