



The Java[®] Language Specification

Java SE 7 Edition

James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley



ORACLE[®]

The Java[®] Language Specification

Java SE 7 Edition

James Gosling
Bill Joy
Guy Steele
Gilad Bracha
Alex Buckley

◆ Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Copyright © 1997, 2013, Oracle and/or its affiliates. All rights reserved.
500 Oracle Parkway, Redwood City, California 94065, U.S.A.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

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

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document, except as specified in the Limited License Grant herein at [Appendix A](#). This document is subject to the Limited License Grant included herein as [Appendix A](#), and may otherwise not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact U.S. Corporate and Government Sales, (800) 382-3419, corpsales@pearsontechgroup.com. For sales outside the United States, please contact International Sales, international@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2012953694
ISBN-13: 978-0-13-326022-9
ISBN-10: 0-13-326022-4

Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to (201) 236-3290.

The Specification provided herein is provided to you only under the Limited License Grant included herein as [Appendix A](#). Please see [Appendix A](#).

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana. First printing, January 2013.

Table of Contents

[Preface to the Java SE 7 Edition](#)

[Preface to the Third Edition](#)

[Preface to the Second Edition](#)

[Preface to the First Edition](#)

[1 Introduction](#)

[1.1 Organization of the Specification](#)

[1.2 Example Programs](#)

[1.3 Notation](#)

[1.4 Relationship to Predefined Classes and Interfaces](#)

[1.5 References](#)

[2 Grammars](#)

[2.1 Context-Free Grammars](#)

[2.2 The Lexical Grammar](#)

[2.3 The Syntactic Grammar](#)

[2.4 Grammar Notation](#)

[3 Lexical Structure](#)

[3.1 Unicode](#)

[3.2 Lexical Translations](#)

[3.3 Unicode Escapes](#)

[3.4 Line Terminators](#)

[3.5 Input Elements and Tokens](#)

[3.6 White Space](#)

[3.7 Comments](#)

[3.8 Identifiers](#)

[3.9 Keywords](#)

[3.10 Literals](#)

[3.10.1 Integer Literals](#)

[3.10.2 Floating-Point Literals](#)

[3.10.3 Boolean Literals](#)

[3.10.4 Character Literals](#)

[3.10.5 String Literals](#)

[3.10.6 Escape Sequences for Character and String Literals](#)

[3.10.7 The Null Literal](#)

[3.11 Separators](#)

[3.12 Operators](#)

[4 Types, Values, and Variables](#)

[4.1 The Kinds of Types and Values](#)

[4.2 Primitive Types and Values](#)

[4.2.1 Integral Types and Values](#)

[4.2.2 Integer Operations](#)

[4.2.3 Floating-Point Types, Formats, and Values](#)

[4.2.4 Floating-Point Operations](#)

[4.2.5 The `boolean` Type and boolean Values](#)

[4.3 Reference Types and Values](#)

[4.3.1 Objects](#)

[4.3.2 The Class `Object`](#)

[4.3.3 The Class `String`](#)

[4.3.4 When Reference Types Are the Same](#)

[4.4 Type Variables](#)

[4.5 Parameterized Types](#)

[4.5.1 Type Arguments and Wildcards](#)

[4.5.2 Members and Constructors of Parameterized Types](#)

[4.6 Type Erasure](#)

[4.7 Reifiable Types](#)

[4.8 Raw Types](#)

[4.9 Intersection Types](#)

[4.10 Subtyping](#)

[4.10.1 Subtyping among Primitive Types](#)

[4.10.2 Subtyping among Class and Interface Types](#)

[4.10.3 Subtyping among Array Types](#)

[4.11 Where Types Are Used](#)

[4.12 Variables](#)

[4.12.1 Variables of Primitive Type](#)

[4.12.2 Variables of Reference Type](#)

[4.12.3 Kinds of Variables](#)

[4.12.4 `final` Variables](#)

[4.12.5 Initial Values of Variables](#)

[4.12.6 Types, Classes, and Interfaces](#)

5 Conversions and Promotions

[5.1 Kinds of Conversion](#)

[5.1.1 Identity Conversion](#)

[5.1.2 Widening Primitive Conversion](#)

[5.1.3 Narrowing Primitive Conversion](#)

[5.1.4 Widening and Narrowing Primitive Conversion](#)

[5.1.5 Widening Reference Conversion](#)

[5.1.6 Narrowing Reference Conversion](#)

[5.1.7 Boxing Conversion](#)

[5.1.8 Unboxing Conversion](#)

[5.1.9 Unchecked Conversion](#)

[5.1.10 Capture Conversion](#)

[5.1.11 String Conversion](#)

[5.1.12 Forbidden Conversions](#)

[5.1.13 Value Set Conversion](#)

[5.2 Assignment Conversion](#)

[5.3 Method Invocation Conversion](#)

[5.4 String Conversion](#)

[5.5 Casting Conversion](#)

[5.5.1 Reference Type Casting](#)

[5.5.2 Checked Casts and Unchecked Casts](#)

[5.5.3 Checked Casts at Run Time](#)

[5.6 Numeric Promotions](#)

[5.6.1 Unary Numeric Promotion](#)

[5.6.2 Binary Numeric Promotion](#)

6 Names

[6.1 Declarations](#)

- [6.2 Names and Identifiers](#)
- [6.3 Scope of a Declaration](#)
- [6.4 Shadowing and Obscuring](#)
 - [6.4.1 Shadowing](#)
 - [6.4.2 Obscuring](#)
- [6.5 Determining the Meaning of a Name](#)
 - [6.5.1 Syntactic Classification of a Name According to Context](#)
 - [6.5.2 Reclassification of Contextually Ambiguous Names](#)
 - [6.5.3 Meaning of Package Names](#)
 - [6.5.3.1 Simple Package Names](#)
 - [6.5.3.2 Qualified Package Names](#)
 - [6.5.4 Meaning of *PackageOrTypeNames*](#)
 - [6.5.4.1 Simple *PackageOrTypeNames*](#)
 - [6.5.4.2 Qualified *PackageOrTypeNames*](#)
 - [6.5.5 Meaning of Type Names](#)
 - [6.5.5.1 Simple Type Names](#)
 - [6.5.5.2 Qualified Type Names](#)
 - [6.5.6 Meaning of Expression Names](#)
 - [6.5.6.1 Simple Expression Names](#)
 - [6.5.6.2 Qualified Expression Names](#)
 - [6.5.7 Meaning of Method Names](#)
 - [6.5.7.1 Simple Method Names](#)
 - [6.5.7.2 Qualified Method Names](#)
- [6.6 Access Control](#)
 - [6.6.1 Determining Accessibility](#)
 - [6.6.2 Details on protected Access](#)
 - [6.6.2.1 Access to a `protected` Member](#)
 - [6.6.2.2 Qualified Access to a `protected` Constructor](#)
- [6.7 Fully Qualified Names and Canonical Names](#)

[7 Packages](#)

- [7.1 Package Members](#)
- [7.2 Host Support for Packages](#)
- [7.3 Compilation Units](#)
- [7.4 Package Declarations](#)
 - [7.4.1 Named Packages](#)
 - [7.4.2 Unnamed Packages](#)
 - [7.4.3 Observability of a Package](#)
- [7.5 Import Declarations](#)
 - [7.5.1 Single-Type-Import Declarations](#)
 - [7.5.2 Type-Import-on-Demand Declarations](#)
 - [7.5.3 Single-Static-Import Declarations](#)
 - [7.5.4 Static-Import-on-Demand Declarations](#)
- [7.6 Top Level Type Declarations](#)

[8 Classes](#)

- [8.1 Class Declarations](#)
 - [8.1.1 Class Modifiers](#)
 - [8.1.1.1 `abstract` Classes](#)
 - [8.1.1.2 `final` Classes](#)
 - [8.1.1.3 `strictfp` Classes](#)
 - [8.1.2 Generic Classes and Type Parameters](#)

[8.1.3 Inner Classes and Enclosing Instances](#)

[8.1.4 Superclasses and Subclasses](#)

[8.1.5 Superinterfaces](#)

[8.1.6 Class Body and Member Declarations](#)

[8.2 Class Members](#)

[8.3 Field Declarations](#)

[8.3.1 Field Modifiers](#)

[8.3.1.1 `static` Fields](#)

[8.3.1.2 `final` Fields](#)

[8.3.1.3 `transient` Fields](#)

[8.3.1.4 `volatile` Fields](#)

[8.3.2 Initialization of Fields](#)

[8.3.2.1 Initializers for Class Variables](#)

[8.3.2.2 Initializers for Instance Variables](#)

[8.3.2.3 Restrictions on the use of Fields during Initialization](#)

[8.4 Method Declarations](#)

[8.4.1 Formal Parameters](#)

[8.4.2 Method Signature](#)

[8.4.3 Method Modifiers](#)

[8.4.3.1 `abstract` Methods](#)

[8.4.3.2 `static` Methods](#)

[8.4.3.3 `final` Methods](#)

[8.4.3.4 `native` Methods](#)

[8.4.3.5 `strictfp` Methods](#)

[8.4.3.6 `synchronized` Methods](#)

[8.4.4 Generic Methods](#)

[8.4.5 Method Return Type](#)

[8.4.6 Method Throws](#)

[8.4.7 Method Body](#)

[8.4.8 Inheritance, Overriding, and Hiding](#)

[8.4.8.1 Overriding \(by Instance Methods\)](#)

[8.4.8.2 Hiding \(by Class Methods\)](#)

[8.4.8.3 Requirements in Overriding and Hiding](#)

[8.4.8.4 Inheriting Methods with Override-Equivalent Signatures](#)

[8.4.9 Overloading](#)

[8.5 Member Type Declarations](#)

[8.5.1 Static Member Type Declarations](#)

[8.6 Instance Initializers](#)

[8.7 Static Initializers](#)

[8.8 Constructor Declarations](#)

[8.8.1 Formal Parameters and Type Parameters](#)

[8.8.2 Constructor Signature](#)

[8.8.3 Constructor Modifiers](#)

[8.8.4 Generic Constructors](#)

[8.8.5 Constructor Throws](#)

[8.8.6 The Type of a Constructor](#)

[8.8.7 Constructor Body](#)

[8.8.7.1 Explicit Constructor Invocations](#)

[8.8.8 Constructor Overloading](#)

[8.8.9 Default Constructor](#)

[8.8.10 Preventing Instantiation of a Class](#)

[8.9 Enums](#)

[8.9.1 Enum Constants](#)

[8.9.2 Enum Body Declarations](#)

[**9 Interfaces**](#)

[9.1 Interface Declarations](#)

[9.1.1 Interface Modifiers](#)

[9.1.1.1 `abstract` Interfaces](#)

[9.1.1.2 `strictfp` Interfaces](#)

[9.1.2 Generic Interfaces and Type Parameters](#)

[9.1.3 Superinterfaces and Subinterfaces](#)

[9.1.4 Interface Body and Member Declarations](#)

[9.2 Interface Members](#)

[9.3 Field \(Constant\) Declarations](#)

[9.3.1 Initialization of Fields in Interfaces](#)

[9.4 Abstract Method Declarations](#)

[9.4.1 Inheritance and Overriding](#)

[9.4.1.1 Overriding \(by Instance Methods\)](#)

[9.4.1.2 Requirements in Overriding](#)

[9.4.1.3 Inheriting Methods with Override-Equivalent Signatures](#)

[9.4.2 Overloading](#)

[9.5 Member Type Declarations](#)

[9.6 Annotation Types](#)

[9.6.1 Annotation Type Elements](#)

[9.6.2 Defaults for Annotation Type Elements](#)

[9.6.3 Predefined Annotation Types](#)

[9.6.3.1 `@Target`](#)

[9.6.3.2 `@Retention`](#)

[9.6.3.3 `@Inherited`](#)

[9.6.3.4 `@Override`](#)

[9.6.3.5 `@SuppressWarnings`](#)

[9.6.3.6 `@Deprecated`](#)

[9.6.3.7 `@SafeVarargs`](#)

[9.7 Annotations](#)

[9.7.1 Normal Annotations](#)

[9.7.2 Marker Annotations](#)

[9.7.3 Single-Element Annotations](#)

[**10 Arrays**](#)

[10.1 Array Types](#)

[10.2 Array Variables](#)

[10.3 Array Creation](#)

[10.4 Array Access](#)

[10.5 Array Store Exception](#)

[10.6 Array Initializers](#)

[10.7 Array Members](#)

[10.8 `Class` Objects for Arrays](#)

[10.9 An Array of Characters is Not a `String`](#)

[**11 Exceptions**](#)

[11.1 The Kinds and Causes of Exceptions](#)

[11.1.1 The Kinds of Exceptions](#)

[11.1.2 The Causes of Exceptions](#)

[11.1.3 Asynchronous Exceptions](#)

[11.2 Compile-Time Checking of Exceptions](#)

[11.2.1 Exception Analysis of Expressions](#)

[11.2.2 Exception Analysis of Statements](#)

[11.2.3 Exception Checking](#)

[11.3 Run-Time Handling of an Exception](#)

12 Execution

[12.1 Java Virtual Machine Startup](#)

[12.1.1 Load the Class `Test`](#)

[12.1.2 Link `Test`: Verify, Prepare, \(Optionally\) Resolve](#)

[12.1.3 Initialize `Test`: Execute Initializers](#)

[12.1.4 Invoke `Test.main`](#)

[12.2 Loading of Classes and Interfaces](#)

[12.2.1 The Loading Process](#)

[12.3 Linking of Classes and Interfaces](#)

[12.3.1 Verification of the Binary Representation](#)

[12.3.2 Preparation of a Class or Interface Type](#)

[12.3.3 Resolution of Symbolic References](#)

[12.4 Initialization of Classes and Interfaces](#)

[12.4.1 When Initialization Occurs](#)

[12.4.2 Detailed Initialization Procedure](#)

[12.5 Creation of New Class Instances](#)

[12.6 Finalization of Class Instances](#)

[12.6.1 Implementing Finalization](#)

[12.6.2 Interaction with the Memory Model](#)

[12.7 Unloading of Classes and Interfaces](#)

[12.8 Program Exit](#)

13 Binary Compatibility

[13.1 The Form of a Binary](#)

[13.2 What Binary Compatibility Is and Is Not](#)

[13.3 Evolution of Packages](#)

[13.4 Evolution of Classes](#)

[13.4.1 `abstract` Classes](#)

[13.4.2 `final` Classes](#)

[13.4.3 `public` Classes](#)

[13.4.4 Superclasses and Superinterfaces](#)

[13.4.5 Class Type Parameters](#)

[13.4.6 Class Body and Member Declarations](#)

[13.4.7 Access to Members and Constructors](#)

[13.4.8 Field Declarations](#)

[13.4.9 `final` Fields and Constants](#)

[13.4.10 `static` Fields](#)

[13.4.11 `transient` Fields](#)

[13.4.12 Method and Constructor Declarations](#)

[13.4.13 Method and Constructor Type Parameters](#)

[13.4.14 Method and Constructor Formal Parameters](#)

[13.4.15 Method Result Type](#)

[13.4.16 `abstract` Methods](#)

[13.4.17 `final` Methods](#)

[13.4.18 `native` Methods](#)

- [13.4.19 `static` Methods](#)
- [13.4.20 `synchronized` Methods](#)
- [13.4.21 Method and Constructor Throws](#)
- [13.4.22 Method and Constructor Body](#)
- [13.4.23 Method and Constructor Overloading](#)
- [13.4.24 Method Overriding](#)
- [13.4.25 Static Initializers](#)
- [13.4.26 Evolution of Enums](#)

[13.5 Evolution of Interfaces](#)

- [13.5.1 `public` Interfaces](#)
- [13.5.2 Superinterfaces](#)
- [13.5.3 Interface Members](#)
- [13.5.4 Interface Type Parameters](#)
- [13.5.5 Field Declarations](#)
- [13.5.6 `abstract` Methods](#)
- [13.5.7 Evolution of Annotation Types](#)

[14 Blocks and Statements](#)

- [14.1 Normal and Abrupt Completion of Statements](#)
- [14.2 Blocks](#)
- [14.3 Local Class Declarations](#)
- [14.4 Local Variable Declaration Statements](#)
 - [14.4.1 Local Variable Declarators and Types](#)
 - [14.4.2 Execution of Local Variable Declarations](#)
- [14.5 Statements](#)
- [14.6 The Empty Statement](#)
- [14.7 Labeled Statements](#)
- [14.8 Expression Statements](#)
- [14.9 The `if` Statement](#)
 - [14.9.1 The `if-then` Statement](#)
 - [14.9.2 The `if-then-else` Statement](#)
- [14.10 The `assert` Statement](#)
- [14.11 The `switch` Statement](#)
- [14.12 The `while` Statement](#)
 - [14.12.1 Abrupt Completion of `while` Statement](#)
- [14.13 The `do` Statement](#)
 - [14.13.1 Abrupt Completion of `do` Statement](#)
- [14.14 The `for` Statement](#)
 - [14.14.1 The basic `for` Statement](#)
 - [14.14.1.1 Initialization of `for` Statement](#)
 - [14.14.1.2 Iteration of `for` Statement](#)
 - [14.14.1.3 Abrupt Completion of `for` Statement](#)
 - [14.14.2 The enhanced `for` statement](#)
- [14.15 The `break` Statement](#)
- [14.16 The `continue` Statement](#)
- [14.17 The `return` Statement](#)
- [14.18 The `throw` Statement](#)
- [14.19 The `synchronized` Statement](#)
- [14.20 The `try` statement](#)
 - [14.20.1 Execution of `try-catch`](#)
 - [14.20.2 Execution of `try-finally` and `try-catch-finally`](#)