

Java语言规范



[Java语言规范_下载链接1](#)

著者:詹姆斯·高斯林 (James Gosling)

出版者:机械工业出版社

出版时间:2016-1-1

装帧:平装

isbn:9787111523994

Oracle官方发布，Java编程语言及规范创建人撰写，国内资深Java技术专家翻译，Ama

zon畅销书!

基于最新Java SE

8, 完整且准确地阐述Java编程语言及其规范, 是深度了解Java语言及其技术细节的权威参考书。

本书由Java技术的发明者编写, 是Java编程语言的权威性技术指南。如果你想知道Java语言构造的精确含义以及各种技术细节, 本书是最好的资源。

本书全面、准确而详细地讨论了Java编程语言, 是Java语言最新版本的规范。全书从Java最基础的文法、类型、变量到高级特性lambda表达式、线程与锁等进行了详尽阐述。该规范针对每一项语言特性都给出了代表性的示例程序, 以帮助读者更容易地理解和掌握这些特性。通过阅读本规范, 可以全面系统地了解Java语言的各项特性, 为充分利用这些特性来编写出更加高效简洁的Java程序提供帮助。读者还可以从Oracle的Java官方网站上看到本书的最新进展和修订, 以了解Java语言的最新发展动向。

本书可作为高等院校Java语言程序设计等相关专业的学习手册, 也可作为Java语言爱好者的参考资料。

本书主要内容

全面反映Java语言的整个发展历程。

详细讲述Java SE 8的语法、语义和结构。

彻底覆盖Java SE 8的主要改进, 包括lambda表达式、方法引用、缺省方法等。

充分解决如何提升JAVA语言中最流行的特性之一——注解的效用。

包含宝贵的见解, 区分JAVA的正式规则和真实的编译器行为。

对每一个使用Java语言进行企业级开发的程序员都是不可或缺的资源。

作者介绍:

詹姆斯·高斯林

(James Gosling)

软件专家, Java编程语言的共同创始人之一, 曾先后为Sun公司、Oracle公司和Google公司工作, 是计算机行业最著名的程序员之一。他于1977年获得了加拿大卡尔加里大学计算机科学学士学位, 1983年获得了美国卡内基梅隆大学计算机科学博士学位。2013年, Gosling博士被选为ACM Fellow; 2015年, 获得IEEE 冯诺伊曼奖。

目录: 出版者的话

译者序

前言

第1章 概述 1

1.1 本书结构 1

1.2 样例程序 4

1.3 表示法 4

1.4 与预定义的类和接口的关系 4

1.5 反馈	5
1.6 参考文献	5
第2章 文法	6
2.1 上下文无关文法	6
2.2 词法	6
2.3 句法	6
2.4 文法表示法	6
第3章 词法结构	9
3.1 Unicode	9
3.2 词法翻译	10
3.3 Unicode转义字符	10
3.4 行终止符	11
3.5 输入元素和符号	12
3.6 空白字符	12
3.7 注释	13
3.8 标识符	14
3.9 关键字	15
3.10 字面常量	15
3.10.1 整数字面常量	15
3.10.2 浮点数字面常量	19
3.10.3 布尔字面常量	21
3.10.4 字符字面常量	21
3.10.5 字符串字面常量	22
3.10.6 字符和字符串字面常量的转义序列	23
3.10.7 空字面常量	24
3.11 分隔符	24
3.12 操作符	24
第4章 类型、值和变量	25
4.1 类型和值的种类	25
4.2 简单类型和值	25
4.2.1 整数类型和值	26
4.2.2 整数操作	26
4.2.3 浮点数类型、格式和值	27
4.2.4 浮点数操作	29
4.2.5 boolean类型和布尔值	31
4.3 引用类型和值	32
4.3.1 对象	33
4.3.2 Object类	35
4.3.3 String类	35
4.3.4 当引用类型相同时	35
4.4 类型变量	36
4.5 参数化类型	37
4.5.1 参数化类型的类型引元	38
4.5.2 参数化类型的成员和构造器	40
4.6 类型擦除	40
4.7 可具化类型	41
4.8 原生类型	42
4.9 交集类型	45
4.10 子类型化	45
4.10.1 简单类型之间的子类型化	46
4.10.2 类与接口类型之间的子类型化	46
4.10.3 数组类型之间的子类型化	47
4.10.4 最低上边界	47

- 4.11 使用类型之处 49
- 4.12 变量 52
 - 4.12.1 简单类型的变量 52
 - 4.12.2 引用类型的变量 52
 - 4.12.3 变量的种类 54
 - 4.12.4 final变量 55
 - 4.12.5 变量的初始值 56
 - 4.12.6 类型、类和接口 57
- 第5章 类型转换与上下文 59
 - 5.1 转换的种类 61
 - 5.1.1 标识转换 61
 - 5.1.2 拓宽简单类型转换 61
 - 5.1.3 窄化简单类型转换 62
 - 5.1.4 拓宽和窄化简单类型转换 64
 - 5.1.5 拓宽引用类型转换 64
 - 5.1.6 窄化引用类型转换 64
 - 5.1.7 装箱转换 65
 - 5.1.8 拆箱转换 66
 - 5.1.9 非受检转换 67
 - 5.1.10 捕获转换 67
 - 5.1.11 字符串转换 69
 - 5.1.12 被禁止的转换 69
 - 5.1.13 值集转换 70
 - 5.2 赋值上下文 70
 - 5.3 方法调用上下文 74
 - 5.4 字符串上下文 75
 - 5.5 强制类型转换上下文 75
 - 5.5.1 引用类型强制类型转换 77
 - 5.5.2 受检强制类型转换和非受检强制类型转换 79
 - 5.5.3 运行时的受检强制类型转换 80
 - 5.6 数字上下文 81
 - 5.6.1 一元数字提升 82
 - 5.6.2 二元数字提升 83
- 第6章 名字 84
 - 6.1 声明 84
 - 6.2 名字与标识符 89
 - 6.3 声明的作用域 91
 - 6.4 遮蔽和遮掩 93
 - 6.4.1 遮蔽 94
 - 6.4.2 遮掩 96
 - 6.5 确定名字的含义 97
 - 6.5.1 根据上下文的名字的句法分类 98
 - 6.5.2 对上下文歧义名字的重分类 100
 - 6.5.3 包名的含义 101
 - 6.5.4 PackageOrTypeNames的含义 102
 - 6.5.5 类型名的含义 102
 - 6.5.6 表达式名的含义 103
 - 6.5.7 方法名的含义 105
 - 6.6 访问控制 106
 - 6.6.1 确定可访问性 107
 - 6.6.2 受保护访问权限的细节 110
 - 6.7 完全限定名和规范名 111
- 第7章 包 113
 - 7.1 包成员 113

- 7.2 主机对包的支持 114
- 7.3 编译单元 115
- 7.4 包声明 116
 - 7.4.1 具名包 116
 - 7.4.2 不具名包 116
 - 7.4.3 包的可观察性 117
- 7.5 导入声明 117
 - 7.5.1 单类型导入声明 118
 - 7.5.2 按需类型导入声明 119
 - 7.5.3 单静态导入声明 120
 - 7.5.4 按需静态导入声明 120
- 7.6 顶层类型声明 121
- 第8章 类 123
 - 8.1 类声明 124
 - 8.1.1 类修饰符 124
 - 8.1.2 泛化类和类型参数 126
 - 8.1.3 内部类和包围实例 128
 - 8.1.4 超类和子类 130
 - 8.1.5 超接口 132
 - 8.1.6 类体和成员声明 134
 - 8.2 类成员 135
 - 8.3 域声明 138
 - 8.3.1 域修饰符 141
 - 8.3.2 域的初始化 145
 - 8.3.3 在域初始化过程中的向前引用 146
 - 8.4 方法声明 148
 - 8.4.1 形参 149
 - 8.4.2 方法签名 152
 - 8.4.3 方法修饰符 152
 - 8.4.4 泛化方法 156
 - 8.4.5 方法的结果 157
 - 8.4.6 方法抛出异常 157
 - 8.4.7 方法体 158
 - 8.4.8 继承、覆盖和隐藏 159
 - 8.4.9 重载 166
 - 8.5 成员类型声明 168
 - 8.5.1 静态成员类型声明 169
 - 8.6 实例初始化器 169
 - 8.7 静态初始化器 169
 - 8.8 构造器声明 170
 - 8.8.1 形参 170
 - 8.8.2 构造器签名 171
 - 8.8.3 构造器修饰符 171
 - 8.8.4 泛化构造器 172
 - 8.8.5 构造器抛出异常 172
 - 8.8.6 构造器的类型 172
 - 8.8.7 构造器体 172
 - 8.8.8 构造器重载 176
 - 8.8.9 缺省构造器 176
 - 8.8.10 阻止类的实例化 177
 - 8.9 枚举类型 177
 - 8.9.1 枚举常量 178
 - 8.9.2 枚举体声明 178
 - 8.9.3 枚举成员 180

- 第9章 接口 184
 - 9.1 接口声明 184
 - 9.1.1 接口修饰符 185
 - 9.1.2 泛化接口和类型参数 185
 - 9.1.3 超接口和子接口 186
 - 9.1.4 接口体和成员声明 187
 - 9.2 接口成员 187
 - 9.3 域（常量）声明 187
 - 9.3.1 接口中域的初始化 189
 - 9.4 方法声明 189
 - 9.4.1 继承和覆盖 190
 - 9.4.2 重载 193
 - 9.4.3 接口方法体 193
 - 9.5 成员类型声明 193
 - 9.6 注解类型 194
 - 9.6.1 注解类型元素 195
 - 9.6.2 注解类型元素的缺省值 197
 - 9.6.3 可重复的注解类型 198
 - 9.6.4 预定义的注解类型 200
 - 9.7 注解 204
 - 9.7.1 普通注解 205
 - 9.7.2 标记注解 207
 - 9.7.3 单元素注解 207
 - 9.7.4 注解可以出现在何处 208
 - 9.7.5 同种类型的多重注解 211
 - 9.8 函数型接口 212
 - 9.9 函数类型 214
- 第10章 数组 218
 - 10.1 数组类型 218
 - 10.2 数组变量 219
 - 10.3 数组创建 220
 - 10.4 数组访问 220
 - 10.5 数组存储异常 220
 - 10.6 数组初始化器 221
 - 10.7 数组成员 222
 - 10.8 数组的Class对象 223
 - 10.9 字符数组不是String 224
- 第11章 异常 225
 - 11.1 异常的种类和成因 225
 - 11.1.1 异常的种类 225
 - 11.1.2 异常的成因 226
 - 11.1.3 异步异常 226
 - 11.2 异常的编译时检查 227
 - 11.2.1 表达式异常分析 228
 - 11.2.2 语句异常分析 228
 - 11.2.3 异常检查 229
 - 11.3 异常的运行时处理 230
- 第12章 执行 233
 - 12.1 Java虚拟机启动 233
 - 12.1.1 加载Test类 233
 - 12.1.2 链接Test：校验、准备、（可选的）解析 233
 - 12.1.3 初始化Test：执行初始化器 234
 - 12.1.4 调用Test.main 234
 - 12.2 加载类和接口 235

- 12.2.1 加载过程 235
- 12.3 链接类和接口 236
 - 12.3.1 二进制表示的校验 236
 - 12.3.2 类或接口类型的准备 236
 - 12.3.3 符号引用的解析 237
- 12.4 初始化类和接口 237
 - 12.4.1 当初始化发生时 238
 - 12.4.2 详细的初始化过程 239
- 12.5 创建新的类实例 241
- 12.6 类实例的终结 243
 - 12.6.1 实现终结 244
 - 12.6.2 与内存模型的交互 245
- 12.7 卸载类和接口 246
- 12.8 程序退出 247
- 第13章 二进制兼容性 248
 - 13.1 二进制形式 249
 - 13.2 二进制兼容性到底是什么 252
 - 13.3 包的演化 252
 - 13.4 类的演化 253
 - 13.4.1 abstract类 253
 - 13.4.2 final类 253
 - 13.4.3 public类 253
 - 13.4.4 超类和超接口 253
 - 13.4.5 类的类型参数 254
 - 13.4.6 类体和成员声明 255
 - 13.4.7 对成员和构造器的访问权限 256
 - 13.4.8 域声明 257
 - 13.4.9 final域和static
常量变量 258
 - 13.4.10 static域 260
 - 13.4.11 transient域 260
 - 13.4.12 方法和构造器声明 260
 - 13.4.13 方法和构造器的类型参数 261
 - 13.4.14 方法和构造器的形式参数 261
 - 13.4.15 方法返回类型 262
 - 13.4.16 abstract方法 262
 - 13.4.17 final方法 262
 - 13.4.18 native方法 263
 - 13.4.19 static方法 263
 - 13.4.20 synchronized方法 263
 - 13.4.21 方法和构造器的抛出物 263
 - 13.4.22 方法和构造器体 263
 - 13.4.23 方法和构造器的重载 264
 - 13.4.24 方法覆盖 264
 - 13.4.25 静态初始化器 264
 - 13.4.26 枚举的演化 265
 - 13.5 接口的演化 265
 - 13.5.1 public接口 265
 - 13.5.2 超接口 265
 - 13.5.3 接口成员 265
 - 13.5.4 接口的类型参数 266
 - 13.5.5 域声明 266
 - 13.5.6 接口方法声明 266
 - 13.5.7 注解类型的演化 267

- 第14章 块和语句 268
 - 14.1 语句的正常结束和猝然结束 268
 - 14.2 块 269
 - 14.3 局部类声明 269
 - 14.4 局部变量声明语句 270
 - 14.4.1 局部变量声明符和类型 271
 - 14.4.2 局部变量声明的执行 271
 - 14.5 语句 271
 - 14.6 空语句 272
 - 14.7 标号语句 273
 - 14.8 表达式语句 274
 - 14.9 if语句 274
 - 14.9.1 if-then语句 274
 - 14.9.2 if-then-else语句 275
 - 14.10 assert语句 275
 - 14.11 switch语句 277
 - 14.12 while语句 280
 - 14.12.1 while语句的猝然结束 280
 - 14.13 do语句 281
 - 14.13.1 do语句的猝然结束 281
 - 14.14 for语句 282
 - 14.14.1 基本for语句 282
 - 14.14.2 增强for语句 284
 - 14.15 break语句 285
 - 14.16 continue语句 287
 - 14.17 return语句 288
 - 14.18 throw语句 289
 - 14.19 synchronized语句 290
 - 14.20 try语句 291
 - 14.20.1 try-catch的执行 293
 - 14.20.2 try-finally和try-catch-finally的执行 294
 - 14.20.3 带资源的try 296
 - 14.21 不可达语句 299
- 第15章 表达式 303
 - 15.1 计算、表示和结果 303
 - 15.2 表达式的形式 303
 - 15.3 表达式的类型 304
 - 15.4 FP-严格的表达式 304
 - 15.5 表达式和运行时检查 305
 - 15.6 计算的正常和猝然结束 306
 - 15.7 计算顺序 307
 - 15.7.1 首先计算左操作数 307
 - 15.7.2 在操作之前计算操作数 308
 - 15.7.3 计算遵循括号和优先级 309
 - 15.7.4 引元列表是自左向右计算的 310
 - 15.7.5 其他表达式的计算顺序 310
 - 15.8 基本表达式 311
 - 15.8.1 词法上的字面常量 312
 - 15.8.2 类字面常量 312
 - 15.8.3 this 313
 - 15.8.4 限定的this 313
 - 15.8.5 带括号的表达式 314
 - 15.9 类实例创建表达式 314
 - 15.9.1 确定要实例化的类 315

- 15.9.2 确定包围实例 316
- 15.9.3 选择构造器及其引元 317
- 15.9.4 类实例创建表达式的运行时计算 319
- 15.9.5 匿名类声明 320
- 15.10 数组创建和访问表达式 321
 - 15.10.1 数组创建表达式 321
 - 15.10.2 数组创建表达式的运行时执行 322
 - 15.10.3 数组访问表达式 324
 - 15.10.4 数组访问表达式的运行时计算 324
- 15.11 域访问表达式 326
 - 15.11.1 使用基本表达式访问域 326
 - 15.11.2 使用super访问超类成员 328
- 15.12 方法调用表达式 329
 - 15.12.1 编译时的步骤1: 确定要搜索的类或接口 330
 - 15.12.2 编译时的步骤2: 确定方法签名 332
 - 15.12.3 编译时的步骤3: 选中的方法是否合适 342
 - 15.12.4 方法调用的运行时计算 343
- 15.13 方法引用表达式 350
 - 15.13.1 方法引用的编译时声明 352
 - 15.13.2 方法引用的类型 355
 - 15.13.3 方法引用的运行时计算 356
- 15.14 后缀表达式 359
 - 15.14.1 表达式名字 359
 - 15.14.2 后缀递增操作符++ 359
 - 15.14.3 后缀递减操作符-- 359
- 15.15 一元操作符 360
 - 15.15.1 前缀递增操作符++ 361
 - 15.15.2 前缀递减操作符-- 361
 - 15.15.3 一元加号操作符+ 362
 - 15.15.4 一元减号操作符- 362
 - 15.15.5 按位取反操作符~ 362
 - 15.15.6 逻辑取反操作符! 362
- 15.16 强制类型转换表达式 363
- 15.17 乘除操作符 364
 - 15.17.1 乘法操作符* 364
 - 15.17.2 除法操作符/ 365
 - 15.17.3 取余操作符% 366
- 15.18 加减操作符 367
 - 15.18.1 字符串连接操作符+ 368
 - 15.18.2 用于数字类型的加减操作符 (+和-) 369
- 15.19 移位操作符 371
- 15.20 关系操作符 371
 - 15.20.1 数字比较操作符<、<=、>和>= 372
 - 15.20.2 类型比较操作符instanceof 372
- 15.21 判等操作符 373
 - 15.21.1 数字判等操作符==和!= 373
 - 15.21.2 布尔判等操作符==和!= 374
 - 15.21.3 引用判等操作符==和!= 374
- 15.22 位操作符与逻辑操作符 375
 - 15.22.1 整数位操作符&、^和| 375
 - 15.22.2 布尔逻辑操作符&、^和| 376
- 15.23 条件与操作符&& 376
- 15.24 条件或操作符|| 376
- 15.25 条件操作符?: 377

- 15.25.1 布尔条件表达式 381
- 15.25.2 数字型条件表达式 381
- 15.25.3 引用条件表达式 382
- 15.26 赋值操作符 383
 - 15.26.1 简单赋值操作符= 383
 - 15.26.2 复合赋值操作符 387
- 15.27 lambda表达式 391
 - 15.27.1 lambda参数 393
 - 15.27.2 lambda体 394
 - 15.27.3 lambda表达式的类型 397
 - 15.27.4 lambda表达式的运行时计算 398
- 15.28 常量表达式 399
- 第16章 明确赋值 400
 - 16.1 明确赋值和表达式 404
 - 16.1.1 布尔常量表达式 404
 - 16.1.2 条件与操作符&& 404
 - 16.1.3 条件或操作符|| 404
 - 16.1.4 逻辑取反操作符! 405
 - 16.1.5 条件操作符?: 405
 - 16.1.6 其他boolean类型的表达式 405
 - 16.1.7 赋值表达式 406
 - 16.1.8 操作符++和-- 406
 - 16.1.9 其他表达式 406
 - 16.2 明确赋值与语句 407
 - 16.2.1 空语句 407
 - 16.2.2 块 407
 - 16.2.3 局部类声明语句 408
 - 16.2.4 局部变量声明语句 408
 - 16.2.5 标号语句 409
 - 16.2.6 表达式语句 409
 - 16.2.7 if语句 409
 - 16.2.8 assert语句 409
 - 16.2.9 switch语句 410
 - 16.2.10 while语句 410
 - 16.2.11 do语句 410
 - 16.2.12 for语句 411
 - 16.2.13 break、continue、return和throw语句 412
 - 16.2.14 synchronized语句 412
 - 16.2.15 try语句 412
 - 16.3 明确赋值与参数 413
 - 16.4 明确赋值与数组初始化器 413
 - 16.5 明确赋值与枚举常量 413
 - 16.6 明确赋值与匿名类 414
 - 16.7 明确赋值与成员类型 414
 - 16.8 明确赋值与静态初始化器 414
 - 16.9 明确赋值、构造器和实例初始化器 415
- 第17章 线程与锁 416
 - 17.1 同步 416
 - 17.2 等待集和通知 417
 - 17.2.1 等待 417
 - 17.2.2 通知 418
 - 17.2.3 中断 418
 - 17.2.4 等待、通知和中断的交互 418
 - 17.3 睡眠和让步 419

- 17.4 内存模型 419
 - 17.4.1 共享变量 421
 - 17.4.2 动作 421
 - 17.4.3 程序和程序顺序 422
 - 17.4.4 同步顺序 422
 - 17.4.5 “之前发生”顺序 423
 - 17.4.6 执行 425
 - 17.4.7 良构执行 425
 - 17.4.8 执行和因果关系要求 426
 - 17.4.9 可观察的行为和不终止的执行 428
- 17.5 final域的语义 429
 - 17.5.1 final域的语义 430
 - 17.5.2 在构造阶段读final域 430
 - 17.5.3 对final域的后续修改 431
 - 17.5.4 写受保护的域 432
- 17.6 字撕裂 432
- 17.7 double和long的非原子化处理 433
- 第18章 类型推断 434
 - 18.1 概念与表示法 435
 - 18.1.1 推断变量 435
 - 18.1.2 约束公式 435
 - 18.1.3 边界 435
 - 18.2 归纳 436
 - 18.2.1 表达式可兼容性约束 437
 - 18.2.2 类型可兼容性约束 440
 - 18.2.3 子类型化约束 440
 - 18.2.4 类型相等性约束 442
 - 18.2.5 受检异常约束 442
 - 18.3 合并 443
 - 18.3.1 互补的边界对 444
 - 18.3.2 涉及捕获转换的边界 444
 - 18.4 解析 445
 - 18.5 推断的使用 446
 - 18.5.1 调用可应用性的推断 447
 - 18.5.2 调用类型的推断 448
 - 18.5.3 函数型接口的参数化版本推断 451
 - 18.5.4 更具体方法的推断 452
- 第19章 语法 454
- 索引 470
 - • • • • (收起)

[Java语言规范_下载链接1_](#)

标签

Java

计算机

软件开发

经典

java

AA-Java

计算机科学

最近想读

评论

一周目未完成, 看过 线程与锁, 感觉对理解语言挺有帮助的; 本来想给 4 分, 翻译扣一分;

没怎么读，如果要设计编程语言的建议看看。

ZWWJ 2016\TP312JA\gsl，太乏味了，也就翻了翻。

感觉翻译的有点不通顺，可能是这本书比较枯燥，大概读了一下，以后遇到对应的问题再来读或许比较好理解。

太难了，后面的部分完全看不懂。

[Java语言规范_下载链接1](#)

书评

为何不买《Think in java》,不是不买而是时候未到。小孩识字“apple”,老师没有开篇大论“苹果”是什么颜色,产地是哪里,也没有说与乔布斯的那个“apple”有什么不同,统统的没有。而是拿着苹果的挂图,说这个就是“苹果”,“苹果”两字这样写。如果以前学习程序的,知道Str...

这本书实际上非常不错,因为是语言规范,对于java语言的任何细节都涉及到了,关于语法层面的任何问题都可以在这本书上找到答案(这样在遇到语法上的问题时,有时直接查这本书比在网上到处搜的效率要高)。翻译的其实还不错,当然了,正如书中的前言所说,该书不适合入门学习, ...

[Java语言规范 下载链接1](#)