

Java

**9-е
издание**

руководство для начинающих

Создавайте, компилируйте и запускайте программы на языке Java прямо сейчас!

Обновленное и расширенное издание



Герберт Шилдт



**Mc
Graw
Hill
Education**



ДУАЛЕКТИКА

Java:

руководство
для начинающих
9-е издание

Java™

A Beginner's Guide

Ninth Edition

Herbert Schildt



New York Chicago San Francisco
Athens London Madrid Mexico City
Milan New Delhi Singapore Sydney Toronto

Java:

руководство для начинающих 9-е издание

Герберт Шилдт

Київ
Видавництво
“НАУКОВИЙ СВІТ”
2023

УДК 004.432

Ш57

Перевод с английского и редакция Ю.Н. Артеменко

Шилдт, Г.

Ш57 Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ. Ю.Н. Артеменко. — Киев. : “Науковий Світ”, 2023. — 752 с. : ил. — Парал. тит. англ.

ISBN 978-617-550-100-9 (укр.)

ISBN 978-1-260-46355-2 (англ.)

В книге раскрыты основы и кратко описаны расширенные функциональные средства языка Java, в числе которых многопоточное программирование, обобщения, лямбда-выражения и графический интерфейс Swing. Вдобавок приводится четкое объяснение перечислений, модулей и методов интерфейса. В этом руководстве предлагается эффективное сочетание теории и практики написания кода, которое позволит быстро приступить к разработке приложений на языке Java. Книга предназначена для программистов, желающих изучить язык Java, и для разработчиков веб-приложений, которые стремятся повысить уровень знаний и мастерства.

УДК 004.432

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства McGraw Hill.

Copyright © 2022 by McGraw Hill.

All rights reserved.

All trademarks are trademarks of their respective owners.

Oracle Corporation does not make any representations or warranties as to the accuracy, adequacy, or completeness of any information contained in this Work, and is not responsible for any errors or omissions.

Authorized translation from the English language edition of the *Java: A Beginner's Guide, 9th Edition* (ISBN 978-1-260-46355-2), published by McGraw Hill.

Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

ISBN 978-617-550-100-9 (укр.)

ISBN 978-1-260-46355-2 (англ.)

© ООО “Науковий Світ”, перевод, 2023

© 2022 by McGraw Hill

Оглавление

Предисловие	19
Глава 1. Основы языка Java	27
Глава 2. Введение в типы данных и операции	65
Глава 3. Операторы управления программой	97
Глава 4. Введение в классы, объекты и методы	135
Глава 5. Дополнительные сведения о типах данных и операциях	165
Глава 6. Дополнительные сведения о методах и классах	217
Глава 7. Наследование	259
Глава 8. Пакеты и интерфейсы	301
Глава 9. Обработка исключений	335
Глава 10. Использование ввода-вывода	365
Глава 11. Многопоточное программирование	411
Глава 12. Перечисления, автоупаковка, аннотации и многое другое	453
Глава 13. Обобщения	487
Глава 14. Лямбда-выражения и ссылки на методы	523
Глава 15. Модули	557
Глава 16. Выражения <code>switch</code> , записи и прочие недавно добавленные средства	589
Глава 17. Введение в Swing	623
Приложение А. Ответы на вопросы и решения упражнений для самопроверки	661
Приложение Б. Использование документирующих комментариев Java	715
Приложение В. Компиляция и запуск простых однофайловых программ за один шаг	725
Приложение Г. Введение в JShell	729
Приложение Д. Дополнительные ключевые слова Java	741
Предметный указатель	746

Содержание

Об авторе	18
Предисловие	19
Эволюция языка Java	19
Структура книги	25
Ключевые навыки и концепции	25
Вопросы и упражнения для самопроверки	25
Спросим у эксперта	25
Упражнения	25
Предыдущий опыт программирования не требуется	25
Необходимое программное обеспечение	25
Исходный код примеров	26
Благодарности	26
Глава 1. Основы языка Java	27
История и философия языка Java	29
Происхождение Java	29
Родословная Java: C и C++	30
Влияние Java на Интернет	31
Магия Java: байт-код	34
Выход за рамки апплетов	36
Более быстрый график выпуска	37
Характеристики языка Java	38
Объектно-ориентированное программирование	39
Инкапсуляция	40
Полиморфизм	41
Наследование	41
Комплект разработчика на Java	42
Первая простая программа	43
Ввод исходного кода программы	44
Компиляция программы	44
Подробный анализ первого примера программы	45
Обработка синтаксических ошибок	48
Вторая простая программа	49
Другие типы данных	51
Два управляющих оператора	54
Оператор <code>if</code>	54
Цикл <code>for</code>	55
Создание блоков кода	57

Точки с запятой и размещение операторов	58
Практика отступов	59
Ключевые слова Java	61
Идентификаторы в Java	62
Библиотеки классов Java	62
Глава 2. Введение в типы данных и операции	65
Важность типов данных	66
Примитивные типы Java	67
Целые числа	67
Типы с плавающей точкой	69
Символы	70
Тип boolean	71
Литералы	73
Шестнадцатеричные, восьмеричные и двоичные литералы	74
Управляющие последовательности символов	74
Строковые литералы	75
Подробный анализ переменных	76
Инициализация переменной	77
Динамическая инициализация	77
Область видимости и время жизни переменных	78
Операции	81
Арифметические операции	81
Инкремент и декремент	82
Операции отношения и логические операции	83
Короткозамкнутые логические операции	85
Операция присваивания	86
Сокращенные операции присваивания	88
Преобразование типов при присваивании	88
Приведение несовместимых типов	90
Старшинство операций	91
Выражения	93
Преобразования типов в выражениях	94
Использование пробельных символов и круглых скобок	95
Глава 3. Операторы управления программой	97
Ввод символов с клавиатуры	98
Оператор if	100
Вложенные операторы if	101
Цепочка if-else-if	102
Традиционный оператор switch	103
Вложенные операторы switch	107
Цикл for	109

Некоторые разновидности цикла <code>for</code>	111
Пропуск частей	112
Бесконечный цикл	113
Циклы без тела	113
Объявление переменных управления циклом внутри цикла <code>for</code>	114
Расширенный цикл <code>for</code>	115
Цикл <code>while</code>	115
Цикл <code>do-while</code>	116
Использование оператора <code>break</code> для выхода из цикла	121
Использование оператора <code>break</code> как разновидности перехода в стиле “ <code>goto</code> ”	123
Использование оператора <code>continue</code>	127
Вложенные циклы	132
Глава 4. Введение в классы, объекты и методы	135
Основы классов	136
Общая форма класса	137
Определение класса	138
Создание объектов	141
Ссылочные переменные и присваивание	142
Методы	142
Добавление метода в класс <code>Vehicle</code>	143
Возврат из метода	145
Возврат значения	146
Использование параметров	148
Добавление параметризованного метода в класс <code>Vehicle</code>	150
Конструкторы	157
Параметризованные конструкторы	158
Добавление конструктора в класс <code>Vehicle</code>	159
Еще раз об операции <code>new</code>	160
Сборка мусора	160
Ключевое слово <code>this</code>	161
Глава 5. Дополнительные сведения о типах данных и операциях	165
Массивы	166
Одномерные массивы	167
Многомерные массивы	172
Двумерные массивы	172
Массивы нестандартной формы	173
Массивы с тремя и более измерениями	175
Инициализация многомерных массивов	175
Альтернативный синтаксис объявления массивов	176
Присваивание ссылок на массивы	176

Использование члена массива <code>length</code>	177
Цикл <code>for</code> в стиле “ <code>for-each</code> ”	183
Проход по многомерным массивам	186
Применение расширенного цикла <code>for</code>	188
Строки	188
Конструирование строк	189
Оперирование строками	190
Массивы строк	192
Строки являются неизменяемыми	192
Использование строки для управления оператором <code>switch</code>	193
Использование аргументов командной строки	196
Использование вывода типов для локальных переменных	197
Использование вывода типов локальных переменных для ссылочных типов	199
Использование вывода типов локальных переменных в цикле <code>for</code>	201
Некоторые ограничения <code>var</code>	202
Побитовые операции	203
Побитовые операции И, ИЛИ, исключающее ИЛИ и НЕ	203
Операции сдвига	208
Составные побитовые операции присваивания	210
Операция ?	213
Глава 6. Дополнительные сведения о методах и классах	217
Управление доступом к членам класса	218
Модификаторы доступа Java	219
Передача объектов методам	225
Способы передачи аргументов	226
Возвращение объектов	228
Перегрузка методов	230
Перегрузка конструкторов	235
Рекурсия	239
Ключевое слово <code>static</code>	242
Статические блоки	244
Введение во вложенные и внутренние классы	248
Аргументы переменной длины	251
Основы аргументов переменной длины	252
Перегрузка методов с аргументами переменной длины	254
Аргументы переменной длины и неоднозначность	256
Глава 7. Наследование	259
Основы наследования	260
Доступ к членам и наследование	263
Конструкторы и наследование	265

Использование ключевого слова <code>super</code> для вызова конструкторов суперкласса	267
Использование ключевого слова <code>super</code> для доступа к членам суперкласса	271
Создание многоуровневой иерархии	274
Когда конструкторы выполняются?	277
Ссылки на суперклассы и объекты подклассов	278
Переопределение методов	282
Переопределенные методы поддерживают полиморфизм	285
Зачем нужны переопределенные методы?	287
Применение переопределения методов к классу <code>TwoDShape</code>	287
Использование абстрактных классов	290
Использование ключевого слова <code>final</code>	294
Использование ключевого слова <code>final</code> для предотвращения переопределения	294
Использование ключевого слова <code>final</code> для предотвращения наследования	295
Использование ключевого слова <code>final</code> с членами данных	295
Класс <code>Object</code>	297
Глава 8. Пакеты и интерфейсы	301
Пакеты	302
Определение пакета	303
Поиск пакетов и <code>CLASSPATH</code>	304
Краткий пример пакета	305
Пакеты и доступ к членам классов	306
Пример доступа к пакету	307
Защищенные члены	309
Импортирование пакетов	311
Библиотека классов Java содержится в пакетах	312
Интерфейсы	313
Реализация интерфейсов	315
Использование ссылок на интерфейсы	318
Переменные в интерфейсах	324
Интерфейсы можно расширять	325
Стандартные методы интерфейса	326
Основы стандартных методов	327
Более реалистичный пример стандартного метода	329
Проблемы множественного наследования	330
Использование статических методов в интерфейсе	331
Закрытые методы интерфейса	332
Заключительные соображения по поводу пакетов и интерфейсов	333

Глава 9. Обработка исключений	335
Иерархия исключений	337
Основы обработки исключений	337
Использование <code>try</code> и <code>catch</code>	338
Простой пример обработки исключений	339
Последствия от перехваченных исключений	341
Исключения позволяют изящно обрабатывать ошибки	342
Использование нескольких операторов <code>catch</code>	343
Перехват подклассов исключений	344
Блоки <code>try</code> могут быть вложенными	346
Генерация исключений	347
Повторная генерация исключений	348
Подробный анализ класса <code>Throwable</code>	349
Использование <code>finally</code>	350
Использование <code>throws</code>	352
Три дополнительных средства в системе исключений	354
Встроенные исключения Java	355
Создание подклассов <code>Exception</code>	358
 Глава 10. Использование ввода-вывода	 365
Ввод-вывод в Java основан на потоках	367
Потоки байтовых и символьных данных	367
Классы потоков байтовых данных	367
Классы потоков символьных данных	369
Предопределенные потоки	370
Использование потоков байтовых данных	370
Чтение консольного ввода	372
Запись консольного вывода	373
Чтение и запись в файлы с использованием потоков байтовых данных	374
Ввод из файла	375
Вывод в файл	379
Автоматическое закрытие файла	380
Чтение и запись двоичных данных	384
Файлы с произвольным доступом	388
Использование потоков Java, основанных на символах	390
Консольный ввод с использованием потоков символьных данных	392
Консольный вывод с использованием потоков символьных данных	396
Файловый ввод-вывод с использованием потоков символьных данных	398
Использование <code>FileWriter</code>	398
Использование <code>FileReader</code>	399
Использование оболочек типов Java для преобразования числовых строк	400
Вопросы и упражнения для самопроверки	410

Глава 11. Многопоточное программирование	411
Основы многопоточности	412
Класс Thread и интерфейс Runnable	414
Создание потока	414
Одно улучшение и два простых изменения	418
Создание нескольких потоков	425
Выяснение, завершен ли поток	427
Приоритеты потоков	430
Синхронизация	433
Использование синхронизированных методов	434
Оператор synchronized	437
Взаимодействие между потоками с использованием notify(), wait() и notifyAll()	439
Пример использования wait() и notify()	441
Приостановка, возобновление и останов потоков	446
Глава 12. Перечисления, автоупаковка, аннотации и многое другое	453
Перечисления	455
Основы перечислений	455
Перечисления Java являются типами классов	458
Методы values() и valueOf()	458
Конструкторы, методы, переменные экземпляра и перечисления	459
Два важных ограничения	461
Перечисления унаследованы от Enum	461
Автоупаковка	468
Оболочки типов	469
Основы автоупаковки	471
Автоупаковка и методы	472
Автоупаковка/автораспаковка и выражения	473
Предостережение	475
Статическое импортирование	475
Аннотации (метаданные)	479
Введение в операцию instanceof	482
Глава 13. Обобщения	487
Основы обобщений	489
Простой пример обобщения	490
Обобщения работают только со ссылочными типами	493
Обобщенные типы различаются на основе их аргументов типов	494
Обобщенный класс с двумя параметрами типов	494
Общая форма обобщенного класса	495

Ограниченные типы	495
Использование аргументов с подстановочными знаками	499
Ограниченные аргументы с подстановочными знаками	501
Обобщенные методы	503
Обобщенные конструкторы	506
Обобщенные интерфейсы	507
Низкоуровневые типы и унаследованный код	513
Выведение типов с помощью ромбовидной операции	516
Выведение типов локальных переменных и обобщения	518
Стирание	518
Ошибки неоднозначности	519
Некоторые ограничения обобщений	520
Невозможность создать экземпляры параметров типов	520
Ограничения, касающиеся статических членов	520
Ограничения, касающиеся обобщенных массивов	520
Ограничения, касающиеся обобщенных исключений	521
Продолжение изучения обобщений	521
Глава 14. Лямбда-выражения и ссылки на методы	523
Введение в лямбда-выражения	525
Основы лямбда-выражений	525
Функциональные интерфейсы	527
Лямбда-выражения в действии	529
Блочные лямбда-выражения	533
Обобщенные функциональные интерфейсы	535
Лямбда-выражения и захват переменных	541
Генерация исключений в лямбда-выражениях	542
Ссылки на методы	544
Ссылки на статические методы	544
Ссылки на методы экземпляра	546
Ссылки на конструкторы	550
Предопределенные функциональные интерфейсы	552
Глава 15. Модули	557
Основы модулей	559
Простой пример модуля	560
Компиляция и запуск первого примера модуля	565
Более подробный анализ операторов <code>requires</code> и <code>exports</code>	566
Модуль <code>java.base</code> и модули платформы	567
Унаследованный код и неименованные модули	569
Экспортирование пакета для указанного модуля	570
Использование <code>requires transitive</code>	571

Использование служб	576
Основы служб и поставщиков служб	576
Ключевые слова, связанные со службами	577
Пример службы, основанной на модулях	578
Дополнительные характеристики модулей	584
Открытые модули	584
Оператор <code>opens</code>	585
Оператор <code>requires static</code>	585
Продолжение изучения модулей	585
Глава 16. Выражения <code>switch</code>, записи и прочие недавно добавленные средства	589
Расширения оператора <code>switch</code>	591
Использование списка констант <code>case</code>	593
Появление выражения <code>switch</code> и оператора <code>yield</code>	594
Появление стрелки в операторе <code>case</code>	596
Подробный анализ оператора <code>case</code> со стрелкой	598
Записи	605
Основы записей	605
Создание конструкторов записи	608
Подробный анализ методов получения для записи	613
Сопоставление с образцом в операции <code>instanceof</code>	613
Запечатанные классы и интерфейсы	616
Запечатанные классы	616
Запечатанные интерфейсы	618
Будущие направления развития	620
Глава 17. Введение в <code>Swing</code>	623
Происхождение и философия, положенная в основу проектного решения <code>Swing</code>	625
Компоненты и контейнеры	627
Компоненты	628
Контейнеры	628
Панели контейнеров верхнего уровня	629
Диспетчеры компоновки	629
Первая простая программа <code>Swing</code>	630
Анализ первой простой программы <code>Swing</code>	632
Обработка событий в <code>Swing</code>	636
События	636
Источники событий	636
Прослушиватели событий	637
Классы событий и интерфейсы прослушивателей	637

Использование JButton	638
Работа с JTextField	642
Создание JCheckBox	645
Работа с JList	648
Использование внутренних анонимных классов или лямбда-выражений для обработки событий	657
Приложение А. Ответы на вопросы и решения упражнений для самопроверки	661
Глава 1. Основы языка Java	662
Глава 2. Введение в типы данных и операции	664
Глава 3. Операторы управления программой	665
Глава 4. Введение в классы, объекты и методы	668
Глава 5. Дополнительные сведения о типах данных и операциях	669
Глава 6. Дополнительные сведения о методах и классах	674
Глава 7. Наследование	678
Глава 8. Пакеты и интерфейсы	681
Глава 9. Обработка исключений	683
Глава 10. Использование ввода-вывода	686
Глава 11. Многопоточное программирование	689
Глава 12. Перечисления, автоупаковка, аннотации и многое другое	691
Глава 13. Обобщения	695
Глава 14. Лямбда-выражения и ссылки на методы	699
Глава 15. Модули	702
Глава 16. Выражения switch, записи и прочие недавно добавленные средства	704
Глава 17. Введение в Swing	708
Приложение Б. Использование документирующих комментариев Java	715
Дескрипторы javadoc	716
@author	717
{@code}	718
@deprecated	718
{@docRoot}	718
@exception	718
@hidden	718
{@index}	718
{@inheritDoc}	719
{@link}	719
{@linkplain}	719
{@literal}	719
@param	720
@provides	720

@return	720
@see	720
@since	721
{@summary}	721
@throws	721
@uses	721
{@value}	721
@version	722
Общая форма документирующего комментария	722
Вывод утилиты javadoc	722
Пример использования документирующих комментариев	722
Приложение В. Компиляция и запуск простых однофайловых программ за один шаг	725
Приложение Г. Введение в JShell	729
Основы JShell	730
Просмотр, редактирование и повторное выполнение кода	733
Добавление метода	734
Создание класса	735
Использование интерфейса	736
Вычисление выражений и использование встроенных переменных	737
Импортирование пакетов	738
Исключения	739
Другие команды JShell	739
Дальнейшее исследование JShell	740
Приложение Д. Дополнительные ключевые слова Java	741
Модификаторы transient и volatile	742
strictfp	742
assert	743
Собственные методы	744
Еще одна форма this	744
Предметный указатель	746

Об авторе

Автор многочисленных бестселлеров **Герберт Шилдт** занимался написанием книг по программированию на протяжении более трех десятилетий и считается признанным авторитетом по языку Java. Журнал *International Developer* назвал его одним из ведущих авторов книг по программированию в мире. Книги Герберта Шилдта продавались миллионными тиражами по всему миру и переведены на многие языки. Он является автором многочисленных книг по Java, в том числе *Java. Полное руководство*; *C++: методики программирования Шилдта*; *Искусство программирования на Java и SWING: руководство для начинающих*. Герберт Шилдт также много писал о языках C, C++ и C#. В книге Эда Бернса *Secrets of the Rock Star Programmers: Riding the IT Crest* указано, что как один из звездных программистов Шилдт интересуется всеми аспектами вычислений, но его основное внимание сосредоточено на языках программирования. Герберт Шилдт получил степени бакалавра и магистра в Иллинойском университете. Его сайт доступен по адресу www.HerbSchildt.com.

О научном редакторе

Доктор Дэнни Ковард работал над всеми версиями платформы Java. Он руководил определением сервлетов Java в первой версии платформы Java EE и в более поздних версиях, внедрением веб-служб в Java ME, а также стратегией и планированием для Java SE 7. Он основал технологию JavaFX и совсем недавно создал крупнейшее дополнение к стандарту Java EE 7, Java WebSocket API. Доктор Дэнни Ковард обладает уникально широким взглядом на многие аспекты технологии Java. Его опыт простирается от написания кода на Java до разработки API с отраслевыми экспертами и работы в течение нескольких лет в качестве исполнительного директора в Java Community Process. Кроме того, он является автором двух книг по программированию на Java: *Java WebSocket Programming* и *Java EE 7: The Big Picture*. Совсем недавно он применил свои знания Java, помогая масштабировать крупные службы на основе Java для одной из самых успешных в мире компаний-разработчиков программного обеспечения. Доктор Дэнни Ковард получил степень бакалавра, магистра и доктора математики в Оксфордском университете.

Предисловие

Цель настоящей книги — научить вас основам программирования на языке Java. В ней используется пошаговый подход с многочисленными примерами, самопроверками и проектами. В книге предполагается отсутствие у читателя предыдущего опыта программирования. Книга начинается с описания основ, таких как компиляция и запуск программы на Java. Затем обсуждаются ключевые слова, функциональные средства и конструкции, формирующие ядро языка Java. Вы также найдете описание ряда наиболее сложных средств Java, включая многопоточное программирование, обобщения, лямбда-выражения, записи и модули. Завершает книгу введение в основы Swing. Закончив изучение книги, вы будете хорошо разбираться в основах программирования на Java.

Прежде всего, важно отметить, что данная книга является лишь отправной точкой. Java — нечто большее, чем просто элементы, определяющие язык. Java также включает обширные библиотеки и инструменты, помогающие разрабатывать программы. Чтобы быть первоклассным программистом на Java, необходимо освоить эти области. После прочтения книги у вас будут знания, необходимые для изучения остальных аспектов Java.

Эволюция языка Java

Коренным образом изменить саму суть программирования смогли только некоторые языки. Но и в этой группе избранных один язык выделяется среди остальных, потому что его влияние было быстрым и широко распространенным. Конечно же, речь идет о Java. Не будет преувеличением сказать, что первоначальный выпуск Java 1.0 в 1995 году компанией Sun Microsystems, Inc. произвел революцию в программировании, которая радикально трансформировала Интернет в по-настоящему интерактивную среду. Вдобавок язык Java установил новый стандарт в проектировании языков программирования.

На протяжении многих лет язык Java продолжал расти, развиваться и иным образом переопределять себя. В отличие от многих других языков, которые медленно внедряют новые функциональные средства, Java часто находится в авангарде разработки языков программирования. Одной из причин является культура инноваций и изменений, которая окружала Java. В результате язык Java претерпел несколько обновлений, из которых одни были относительно небольшими, а другие — более значительными.

Первым крупным обновлением Java стала версия 1.1. Возможности, добавленные в Java 1.1, оказались более существенными, чем можно было подумать из-за увеличения номера только младшей версии. Например, в Java 1.1 добавлено много новых библиотечных элементов, переопределен способ обработки событий и перенастроены многие функции исходной библиотеки версии 1.0.

Следующим крупным выпуском Java был Java 2, где “2” означает “второе поколение”. Создание Java 2 стало переломным событием, положившим начало “современной эпохи” Java. Первый выпуск Java 2 имел номер версии 1.2. Может показаться странным, что для первого выпуска Java 2 использовался номер версии 1.2. Причина в том, что первоначально он относился к внутреннему номеру версии библиотек Java, но потом был обобщен для ссылки на целый выпуск. С выходом Java 2 в компании Sun переупаковали продукт Java как J2SE (Java 2 Platform Standard Edition — платформа Java 2, стандартная редакция) и номера версий стали применяться к данному продукту.

Первым крупным обновлением первоначального выпуска Java 2 стала версия J2SE 1.3. По большей части она расширяла существующую функциональность и “усиливала” среду разработки. Выпуск J2SE 1.4 дополнительно улучшил Java. Он содержал несколько важных обновлений, улучшений и дополнений, в том числе цепочки исключений, подсистему ввода-вывода на основе каналов и ключевое слово `assert`.

Выпуск J2SE 5 произвел по существу вторую революцию Java. В отличие от большинства предшествующих обновлений Java, которые предлагали важные, но умеренные улучшения, выпуск J2SE 5 фундаментальным образом расширил границы, возможности и область использования языка. Чтобы оценить значимость изменений, которые были внесены в Java в выпуске J2SE 5, взгляните на следующий список, где перечислены только основные из всех новых функциональных средств:

- обобщения;
- автоупаковка и автораспаковка;
- перечисления;
- расширенный цикл `for` в стиле “`for-each`”;
- аргументы переменной длины (`vararg`);
- статическое импортирование;
- аннотации.

Мелкие подстройки или поэтапные обновления в списке не указаны. Каждый элемент списка представляет значительное дополнение языка Java. Некоторые из них, такие как обобщения, расширенный цикл `for` и аргументы переменной длины, вводили новые синтаксические элементы. Другие, в числе которых автоупаковка и автораспаковка, изменяли семантику языка. Аннотации привнесли в программирование совершенно новое измерение.

Важность упомянутых выше новых средств отражена в назначенном номере версии — 5. В других условиях следующим номером версии Java был бы 1.5. Однако новые средства были настолько значительными, что переход от версии 1.4 к 1.5, казалось, просто не смог бы выразить масштаб изменений. Взамен в Sun предпочли увеличить номер версии до 5, чтобы подчеркнуть тот факт, что произошло важное событие. Таким образом, выпуск получил название J2SE 5, а комплект разработчика на Java (Java Development Kit) — JDK 5. Тем не менее, ради соблюдения согласованности в Sun решили использовать 1.5 в качестве внутреннего номера версии, который также называют номером версии разработчиков. Цифра 5 в J2SE 5 называется номером версии продукта.

Следующий выпуск Java получил имя Java SE 6. В Sun снова решили изменить название платформы Java. Прежде всего, обратите внимание, что цифра 2 была отброшена. Соответственно, платформа стала называться Java SE, а официальный продукт — Java Platform, Standard Edition 6 (платформа Java, стандартная редакция). Комплекту разработчика на Java было дано название JDK 6. По аналогии с J2SE 5 цифра 6 в Java SE 6 — это номер версии продукта. Внутренним номером версии разработчиков являлся 1.6.

Выпуск Java SE 6 построен на основе J2SE 5 с добавлением поэтапных усовершенствований. Он не дополнял какими-либо важными средствами сам язык Java, но расширил библиотеки API, добавил ряд новых пакетов и предоставил усовершенствования исполняющей среды. Кроме того, в течение своего длинного (в понятиях Java) жизненного цикла он прошел через несколько модернизаций, попутно добавив некоторое количество обновлений. В целом выпуск Java SE 6 послужил дальнейшему укреплению достижений выпуска J2SE 5.

Очередным выпуском Java стал Java SE 7 с названием комплекта разработчика JDK 7 и внутренним номером версии 1.7. Выпуск Java SE 7 был первым крупным выпуском Java после того, как компанию Sun Microsystems приобрела компания Oracle. Выпуск Java SE 7 содержал много новых функциональных средств, включая существенные дополнения языка и библиотек API. Новые языковые средства были разработаны в рамках проекта Project Coin. Целью проекта Project Coin была идентификация ряда небольших изменений языка Java, подлежащих включению в JDK 7. Ниже приведен список языковых средств, добавленных комплектом JDK 7.

- Наделение типа `String` способностью управлять оператором `switch`.
- Двоичные целочисленные литералы.
- Символы подчеркивания в числовых литералах.
- Расширение оператора `try` под названием оператор `try с ресурсами`, который поддерживает автоматическое управление ресурсами.
- Выведение типов (через ромбовидную операцию) при конструировании экземпляра обобщенного типа.
- Усовершенствованная обработка исключений, при которой два или большее количество исключений можно перехватывать одним оператором `catch` (многократный перехват), и улучшенная проверка типов для исключений, генерируемых повторно.

Как видите, несмотря на то, что средства Project Coin считались небольшими изменениями, внесенными в язык, их преимущества были гораздо значительнее, чем можно было бы предположить по характеристике “небольшие”. В частности, оператор `try` с ресурсами основательно повлиял на способ написания значительного объема кода.

Следующим выпуском Java был Java SE 8 с комплектом разработчика JDK 8. Он имел внутренний номер версии 1.8. Комплект JDK 8 стал значительным обновлением языка Java из-за включения многообещающего нового языкового средства: лямбда-выражений. Влияние лямбда-выражений было и продолжило быть весьма существенным, изменяя как способ концептуализации программных решений, так и способ написания кода Java. Помимо прочего лямбда-выражения содействуют упрощению и сокращению объема исходного кода, необходимого для создания определенных конструкций, таких как некоторые виды анонимных классов. Добавление лямбда-выражений также привело к появлению в языке новой операции (`->`) и нового элемента синтаксиса. Кроме лямбда-выражений в JDK 8 было добавлено много других важных новых средств. Например, начиная с JDK 8, появилась возможность определять стандартную реализацию для метода, объявленного в интерфейсе. В конечном итоге Java SE 8 стал крупным выпуском, значительно расширившим возможности языка и изменившим способ написания кода Java.

Следующим выпуском Java стал Java SE 9. Комплект разработчика назывался JDK 9, а внутренним номером версии также был 9. Комплект JDK 9 представлял крупный выпуск Java, включая значительные улучшения как языка Java, так и его библиотек. Главным новым средством JDK 9 были *модули*, которые позволили указывать взаимосвязи и зависимости кода, составляющего приложение. Модули также добавляют еще одно измерение к средствам контроля доступа Java. Включение модулей привело к добавлению в Java нового элемента синтаксиса, нескольких ключевых слов и различных усовершенствований инструментов. Модули оказали основательное влияние и на библиотеку API, т.к. начиная с JDK 9, библиотечные пакеты организованы в виде модулей.

Помимо модулей в JDK 9 включен ряд других новых средств. Особый интерес представляет JShell — инструмент, который поддерживает интерактивное экспериментирование с программами и изучение Java. (Введение в JShell приведено в приложении Б.) Еще одним интересным обновлением является поддержка закрытых методов интерфейсов. Их добавление в дальнейшем расширяет поддержку JDK 8 стандартных методов в интерфейсах. В JDK 9 инструмент `javadoc` был снабжен возможностью поиска, для поддержки которой предусмотрен новый дескриптор под названием `@index`. Как и предшествующие выпуски, JDK 9 вносит несколько усовершенствований в библиотеки Java API.

Обычно в любом выпуске Java наибольший интерес вызывают новые средства. Тем не менее, выпуск JDK 9 примечателен тем, что в нем объявлен устаревшим один заметный аспект Java: апплеты. Начиная с JDK 9, применять апплеты в новых проектах больше не рекомендуется. Как объяснялось ранее в главе,

из-за ослабления поддержки апплетов со стороны браузеров (и других факторов) в выпуске JDK 9 объявлен устаревшим весь API для апплетов.

Следующим выпуском Java был Java SE 10 (JDK 10). Тем не менее, до этого выпуска в графике выпусков Java произошло значительное изменение. В прошлом между крупными выпусками часто проходило два или более лет. Однако, начиная с JDK 10, время между выпусками значительно сократилось. Ожидается, что в соответствие со строгим графиком выхода расчетное время между крупными выпусками (которые теперь называются выпусками функциональных средств) составляет всего лишь полгода. В результате выпуск JDK 10 появился в марте 2018 года, т.е. спустя полгода после выпуска JDK 9. Такая увеличенная частота выпусков позволит программистам на Java получать своевременный доступ к новым средствам и улучшениям. Вместо того чтобы ждать два года или дольше, когда новое функциональное средство будет готово, оно становится частью следующего запланированного выпуска.

Еще одним аспектом изменений в графике выпусков Java является выпуск с долгосрочной поддержкой (long-term support — LTS). Ожидается, что выпуск LTS будет производиться каждые три года. Выпуск LTS будет поддерживаться (и, следовательно, оставаться жизнеспособным) в течение периода времени, превышающего полгода. Первым выпуском LTS был JDK 11. Вторым выпуском LTS стал JDK 17, с учетом которого была обновлена эта книга. Из-за стабильности, предлагаемой выпуском LTS, вполне вероятно, что его набор средств будет определять базовый уровень функциональности для промежутка в несколько лет. Последние сведения о долгосрочной поддержке и графике выхода выпусков LTS ищите на веб-сайте Oracle.

Главным новым языковым средством, добавленным JDK 10, стала поддержка выведения типов локальных переменных, благодаря которому теперь можно позволить выводить тип локальной переменной из типа ее инициализатора, а не указывать его явно. Для обеспечения такой возможности в Java было добавлено контекстно-чувствительное ключевое слово `var`. Выведение типов способно упростить код за счет устранения необходимости избыточно указывать тип переменной, когда его можно вывести из инициализатора переменной. Выведение типов также может упростить объявления в ситуациях, при которых тип трудно понять или невозможно указать явно. Выведение типов локальных переменных стало обычной частью современной программной среды. Его включение в Java помогло сохранять Java в актуальном состоянии с учетом меняющихся тенденций в проектировании языков. Наряду с другими изменениями в JDK 10 переопределена строка версии Java с изменением смысла номеров версий, чтобы они лучше соотносились с новым графиком выпуска.

Следующим выпуском Java был Java SE 11 (JDK 11). Он вышел в сентябре 2018 года, т.е. через полгода после JDK 10, и был выпуском LTS. Главным новым языковым средством в JDK 11 являлась поддержка использования ключевого слова `var` в лямбда-выражениях. Кроме того, загрузчик приложений Java получил еще один режим выполнения, позволяющий напрямую запускать простые однофайловые программы. В JDK 11 также были удалены некоторые средства.

Возможно, наибольший интерес представляло удаление поддержки апплетов ввиду их исторической значимости. Вспомните, что апплеты впервые были объявлены nereкомендуемыми в JDK 9. В выпуске JDK 11 поддержка апплетов была удалена. Поддержка другой технологии развертывания, называемой Java Web Start, тоже была удалена в JDK 11. Еще одним ключевым изменением в JDK 11 стало то, что JavaFX больше не входит в состав JDK. Взамен эта инфраструктура для построения графических пользовательских интерфейсов превратилась в проект с открытым кодом. Из-за того, что упомянутые средства перестали быть частью JDK, в книге они не обсуждаются.

Между LTS-выпуском JDK 11 и следующим LTS-выпуском (JDK 17) вышли пять выпусков функциональных средств: с JDK 12 до JDK 16. В выпусках JDK 12 и JDK 13 новые языковые средства не добавлялись. В выпуске JDK 14 была добавлена поддержка выражения `switch`, которое представляет собой конструкцию `switch`, производящую значение. Также были проведены другие усовершенствования `switch`. В выпуске JDK 15 появились текстовые блоки, по существу являющиеся строковыми литералами, которые могут занимать более одной строчки. В выпуске JDK 16 операция `instanceof` была расширена сопоставлением с образцом и добавлен новый тип класса под названием запись вместе с новым контекстно-чувствительным ключевым словом `record`. Запись предлагает удобный способ объединения данных. В выпуске JDK 16 также предоставляется новый инструмент пакетирования приложений, называемый `jpackage`.

На момент написания книги самой последней версией Java была Java SE 17 (JDK 17). Как упоминалось ранее, это второй LTS-выпуск Java, что имеет особое значение. Его главное новое средство — возможность запечатывать классы и интерфейсы. Запечатывание дает вам контроль над наследованием класса, а также над наследованием и реализацией интерфейса. Для такой цели были добавлены контекстно-чувствительные ключевые слова `sealed`, `permits` и `non-sealed` (первое ключевое слово Java с дефисом). Кроме того, в JDK 17 произошла пометка API для апплетов как устаревшего и подлежащего удалению. Вы уже знаете, что поддержку апплетов убрали несколько лет назад. Однако API для апплетов просто был помечен как nereкомендуемый, что позволяло компилировать практически исчезающий код, который полагался на этот API. После выхода JDK 17 теперь API для апплетов подлежит удалению в будущем выпуске.

И еще один момент относительно эволюции Java: в 2006 году начался процесс открытия исходного кода Java. В наши дни доступны реализации JDK с открытым кодом. Открытие исходного кода дополнительно способствует динамичной природе процесса разработки Java. В конечном итоге наследием инноваций Java является безопасность. Java остается живым и гибким языком, который привыкли ожидать в мире программирования.

Материал настоящей книги обновлен с учетом JDK 17. Тем не менее, как подчеркивалось в предыдущем обсуждении, история программирования на Java характеризуется энергичными изменениями. Рекомендуется отслеживать новые возможности в каждом последующем выпуске Java. Проще говоря: эволюция Java продолжается!

Структура книги

Эта книга представляет собой учебное пособие, каждый раздел которого опирается на материал предыдущего раздела. Книга содержит 17 глав, в каждой из которых обсуждается тот или иной аспект языка Java. Уникальность книги в том, что она включает специальные элементы, которые закрепляют полученные знания.

Ключевые навыки и концепции

Каждая глава начинается с перечня важнейших навыков, которые вы получите в результате изучения главы.

Вопросы и упражнения для самопроверки

Каждая глава завершается вопросами и упражнениями, которые позволяют вам проверить свои знания. Ответы находятся в приложении А.

Спросим у эксперта

В книге повсеместно встречаются специальные врезки “Спросим у эксперта”, содержащие дополнительную информацию либо интересные комментарии по теме. В них используется формат “вопрос-ответ”.

Упражнения

Каждая глава содержит одно или несколько упражнений, которые представляют собой проекты, демонстрирующие применение на практике того, что вы изучаете. Во многих случаях это реальные примеры, которые вы можете использовать в качестве отправной точки при написании собственных программ.

Предыдущий опыт программирования не требуется

В книге предполагается отсутствие у читателя предыдущего опыта программирования. Таким образом, если вы никогда раньше не программировали, то все равно можете воспользоваться этой книгой. При наличии опыта программирования вы сможете продвигаться немного быстрее. Однако имейте в виду, что Java обладает рядом ключевых отличий от других популярных языков программирования. Важно не делать поспешных выводов. Таким образом, даже для опытного программиста рекомендуется читать внимательно.

Необходимое программное обеспечение

Для компиляции и запуска всех программ, рассмотренных в книге, вам понадобится последняя версия JDK, которой на момент написания книги является JDK 17. Это JDK для Java SE 17. Инструкции по получению Java JDK приведены в главе 1.

Если вы имеете дело с более ранней версией Java, то по-прежнему сможете пользоваться книгой, но утратите возможность компиляции и запуска программ, в которых задействованы новые функциональные средства Java.

Исходный код примеров

Исходный код для всех примеров и проектов, рассмотренных в книге, доступен на веб-сайте издательства.

Благодарности

Я хочу выразить особую благодарность Дэнни Коварду, научному редактору этого издания книги. Дэнни работал над несколькими моими книгами, и его советы, идеи и предложения всегда имели большую ценность и получали высокую оценку.