

Содержание

| | |
|--|----|
| Об авторе | 16 |
| О техническом редакторе | 16 |
| Введение | 17 |
| Эволюция Java | 17 |
| Java SE 10 | 21 |
| Структура книги | 22 |
| Вопросы и упражнения для самопроверки | 22 |
| Вопросы к эксперту | 22 |
| Упражнения к главам | 22 |
| Книга для всех программистов | 22 |
| Необходимое программное обеспечение | 22 |
| Исходный код примеров программ | 23 |
| Ждем ваших отзывов! | 23 |
| Глава 1. Основы Java | 25 |
| Истоки Java | 27 |
| Связь Java с языками C и C++ | 28 |
| Вклад Java в развитие Интернета | 29 |
| Облегчение разработки интернет-приложений с помощью Java | 30 |
| Java-апплеты | 30 |
| Безопасность | 31 |
| Переносимость | 31 |
| Волшебный байт-код Java | 32 |
| За пределами апплетов | 34 |
| Основные характеристики Java | 35 |
| Объектно-ориентированное программирование | 35 |
| Инкапсуляция | 37 |
| Полиморфизм | 37 |
| Наследование | 38 |
| Установка Java Development Kit | 39 |
| Первая программа на Java | 40 |
| Ввод исходного кода программ | 41 |
| Компиляция программы | 41 |
| Построчный анализ исходного кода примера | 42 |
| Обработка синтаксических ошибок | 45 |
| Еще одна простая программа | 46 |
| Другие типы данных | 48 |
| Две управляющие инструкции | 51 |
| Инструкция <code>if</code> | 51 |
| Цикл <code>for</code> | 53 |
| Создание блоков кода | 54 |

| | |
|---|-----------|
| Использование точки с запятой в коде программы | 56 |
| Стилевое оформление текста программ с помощью отступов | 57 |
| Ключевые слова Java | 59 |
| Идентификаторы в Java | 60 |
| Библиотеки классов Java | 60 |
| Глава 2. Знакомство с типами данных и операторами | 63 |
| Почему типы данных столь важны | 64 |
| Примитивные типы данных Java | 65 |
| Целочисленные типы данных | 65 |
| Типы данных с плавающей точкой | 67 |
| Символы | 68 |
| Логический тип данных | 70 |
| Литералы | 72 |
| Шестнадцатеричные, восьмеричные и двоичные литералы | 73 |
| Управляющие последовательности символов | 73 |
| Строковые литералы | 74 |
| Подробнее о переменных | 75 |
| Инициализация переменных | 76 |
| Динамическая инициализация | 76 |
| Область действия и время жизни переменных | 77 |
| Операторы | 80 |
| Арифметические операторы | 80 |
| Инкремент и декремент | 82 |
| Операторы сравнения и логические операторы | 83 |
| Укороченные логические операторы | 85 |
| Оператор присваивания | 87 |
| Составные операторы присваивания | 87 |
| Преобразование типов при присваивании | 88 |
| Приведение несовместимых типов | 90 |
| Приоритеты операций | 91 |
| Выражения | 94 |
| Преобразование типов в выражениях | 94 |
| Пробелы и круглые скобки | 96 |
| Глава 3. Управляющие инструкции | 99 |
| Ввод символов с клавиатуры | 100 |
| Условная инструкция <code>if</code> | 102 |
| Вложенные условные инструкции <code>if</code> | 103 |
| Многоступенчатая конструкция <code>if-else-if</code> | 104 |
| Инструкция <code>switch</code> | 106 |
| Вложенные инструкции <code>switch</code> | 109 |
| Цикл <code>for</code> | 112 |
| Некоторые разновидности цикла <code>for</code> | 114 |
| Пропуск отдельных частей в определении цикла <code>for</code> | 115 |
| Бесконечный цикл | 116 |
| Циклы без тела | 117 |
| Объявление управляющих переменных в цикле <code>for</code> | 117 |
| Расширенный цикл <code>for</code> | 118 |

| | |
|---|------------|
| Цикл <code>while</code> | 118 |
| Цикл <code>do-while</code> | 120 |
| Применение инструкции <code>break</code> для выхода из цикла | 125 |
| Применение инструкции <code>break</code> в качестве оператора <code>goto</code> | 127 |
| Использование инструкции <code>continue</code> | 131 |
| Вложенные циклы | 136 |
| Глава 4. Знакомство с классами, объектами и методами | 139 |
| Основные сведения о классах | 140 |
| Общая форма определения класса | 141 |
| Определение класса | 142 |
| Порядок создания объектов | 145 |
| Переменные ссылочного типа и присваивание | 146 |
| Методы | 147 |
| Добавление метода в класс <code>Vehicle</code> | 148 |
| Возврат из метода | 150 |
| Возврат значения | 151 |
| Использование параметров | 153 |
| Добавление параметризованного метода в класс <code>Vehicle</code> | 155 |
| Конструкторы | 162 |
| Параметризованные конструкторы | 164 |
| Добавление конструктора в класс <code>Vehicle</code> | 164 |
| Еще раз об операторе <code>new</code> | 166 |
| Сборка мусора | 167 |
| Ключевое слово <code>this</code> | 167 |
| Глава 5. Подробнее о типах данных и операторах | 171 |
| Массивы | 172 |
| Одномерные массивы | 173 |
| Многомерные массивы | 178 |
| Двумерные массивы | 178 |
| Нерегулярные массивы | 179 |
| Трехмерные, четырехмерные и многомерные массивы | 181 |
| Инициализация многомерных массивов | 181 |
| Альтернативный синтаксис объявления массивов | 182 |
| Присваивание ссылок на массивы | 183 |
| Применение переменной экземпляра <code>length</code> | 184 |
| Цикл типа <code>for-each</code> | 191 |
| Циклическое обращение к элементам многомерных массивов | 194 |
| Использование расширенного цикла <code>for</code> | 195 |
| Символьные строки | 196 |
| Создание строк | 196 |
| Операции над символьными строками | 197 |
| Массивы строк | 199 |
| Неизменяемость строк | 200 |
| Использование строк для управления инструкцией <code>switch</code> | 201 |
| Использование аргументов командной строки | 203 |
| Побитовые операторы | 204 |
| Побитовые операции И, ИЛИ, исключающее ИЛИ и НЕ | 205 |

| | |
|---|------------|
| Операции побитового сдвига | 210 |
| Побитовые составные операторы присваивания | 213 |
| Оператор ? | 215 |
| Глава 6. Подробнее о методах и классах | 219 |
| Управление доступом к членам класса | 220 |
| Модификаторы доступа в Java | 221 |
| Передача объектов методам | 227 |
| Способы передачи аргументов методу | 228 |
| Возврат объектов методами | 231 |
| Перегрузка методов | 233 |
| Перегрузка конструкторов | 238 |
| Рекурсия | 244 |
| Применение ключевого слова <code>static</code> | 246 |
| Статические блоки | 249 |
| Вложенные и внутренние классы | 253 |
| Переменное число аргументов | 257 |
| Использование методов с переменным числом аргументов | 257 |
| Перегрузка методов с переменным числом аргументов | 260 |
| Переменное число аргументов и неоднозначность | 262 |
| Глава 7. Наследование | 265 |
| Основы наследования | 266 |
| Наследование и доступ к членам класса | 270 |
| Конструкторы и наследование | 272 |
| Использование ключевого слова <code>super</code> для вызова конструктора суперкласса | 274 |
| Использование ключевого слова <code>super</code> для доступа к членам суперкласса | 278 |
| Создание многоуровневой иерархии классов | 282 |
| Очередность вызова конструкторов | 285 |
| Ссылки на суперкласс и объекты подклассов | 287 |
| Переопределение методов | 291 |
| Поддержка полиморфизма в переопределяемых методах | 294 |
| Для чего нужны переопределяемые методы | 296 |
| Демонстрация механизма переопределения методов на примере класса <code>TwoDShape</code> | 297 |
| Использование абстрактных классов | 301 |
| Использование ключевого слова <code>final</code> | 305 |
| Предотвращение переопределения методов | 305 |
| Предотвращение наследования | 306 |
| Применение ключевого слова <code>final</code> к переменным экземпляра | 306 |
| Класс <code>Object</code> | 308 |
| Глава 8. Пакеты и интерфейсы | 311 |
| Пакеты | 312 |
| Определение пакета | 313 |
| Поиск пакетов и переменная среды <code>CLASSPATH</code> | 314 |
| Простой пример применения пакета | 315 |
| Пакеты и доступ к членам классов | 316 |
| Пример доступа к пакету | 318 |
| Защищенные члены классов | 319 |

| | |
|---|------------|
| Импорт пакетов | 322 |
| Библиотечные классы Java, содержащиеся в пакетах | 323 |
| Интерфейсы | 323 |
| Реализация интерфейсов | 325 |
| Применение интерфейсных ссылок | 329 |
| Переменные в интерфейсах | 336 |
| Наследование интерфейсов | 337 |
| Методы интерфейсов, используемые по умолчанию | 338 |
| Основные сведения о методах по умолчанию | 340 |
| Практический пример использования метода по умолчанию | 342 |
| Множественное наследование | 343 |
| Использование статических методов интерфейса | 344 |
| Закрытые методы интерфейса | 345 |
| Итоговые замечания относительно пакетов и интерфейсов | 346 |
| Глава 9. Обработка исключений | 349 |
| Иерархия исключений | 351 |
| Общие сведения об обработке исключений | 351 |
| Использование инструкций <code>try</code> и <code>catch</code> | 352 |
| Простой пример обработки исключений | 353 |
| Необработанные исключения | 355 |
| Обработка исключений — изящный способ устранения программных ошибок | 357 |
| Множественные блоки <code>catch</code> | 358 |
| Перехват исключений, генерируемых подклассами | 359 |
| Вложенные блоки <code>try</code> | 360 |
| Генерирование исключений | 362 |
| Повторное генерирование исключений | 362 |
| Подробнее о классе <code>Throwable</code> | 364 |
| Использование ключевого слова <code>finally</code> | 365 |
| Использование ключевого слова <code>throws</code> | 367 |
| Три дополнительных средства обработки исключений | 369 |
| Встроенные классы исключений Java | 371 |
| Создание подклассов, производных от класса <code>Exception</code> | 372 |
| Глава 10. Ввод-вывод данных | 381 |
| Потоковая организация ввода-вывода в Java | 383 |
| Байтовые и символьные потоки | 383 |
| Классы байтовых потоков | 383 |
| Классы символьных потоков | 384 |
| Встроенные потоки | 385 |
| Использование байтовых потоков | 386 |
| Консольный ввод | 388 |
| Вывод на консоль | 389 |
| Чтение и запись файлов с использованием байтовых потоков | 390 |
| Чтение данных из файла | 391 |
| Запись в файл | 395 |
| Автоматическое закрытие файлов | 397 |
| Чтение и запись двоичных данных | 401 |
| Файлы с произвольным доступом | 405 |

| | |
|---|------------|
| Использование символьных потоков Java | 408 |
| Консольный ввод с использованием символьных потоков | 411 |
| Вывод на консоль с использованием символьных потоков | 413 |
| Файловый ввод-вывод с использованием символьных потоков | 415 |
| Класс <code>FileWriter</code> | 415 |
| Класс <code>FileReader</code> | 416 |
| Использование классов-оболочек для преобразования числовых строк | 418 |
| Глава 11. Многопоточное программирование | 429 |
| Основы многопоточной обработки | 430 |
| Класс <code>Thread</code> и интерфейс <code>Runnable</code> | 432 |
| Создание потока | 432 |
| Несложные усовершенствования многопоточной программы | 436 |
| Создание нескольких потоков | 444 |
| Определяем момент завершения потока | 446 |
| Приоритеты потоков | 449 |
| Синхронизация | 452 |
| Использование синхронизированных методов | 453 |
| Синхронизированные блоки кода | 456 |
| Организация взаимодействия потоков с помощью методов <code>notify()</code> , <code>wait()</code> и <code>notifyAll()</code> | 459 |
| Пример использования методов <code>wait()</code> и <code>notify()</code> | 461 |
| Приостановка, возобновление и остановка потоков | 467 |
| Глава 12. Перечисления, автоупаковка, статический импорт и аннотации | 475 |
| Перечисления | 476 |
| Основные сведения о перечислениях | 477 |
| Перечисления Java являются типами классов | 479 |
| Методы <code>values()</code> и <code>valueOf()</code> | 480 |
| Конструкторы, методы, переменные экземпляра и перечисления | 481 |
| Два важных ограничения | 483 |
| Перечисления наследуются от класса <code>Enum</code> | 484 |
| Автоупаковка | 491 |
| Оболочки типов | 491 |
| Основные сведения об автоупаковке | 493 |
| Автоупаковка и методы | 494 |
| Автоупаковка и автораспаковка в выражениях | 496 |
| Предупреждение относительно использования автоупаковки и автораспаковки | 497 |
| Статический импорт | 498 |
| Аннотации (метаданные) | 501 |
| Глава 13. Обобщения | 507 |
| Основные сведения об обобщениях | 508 |
| Простой пример обобщений | 510 |
| Обобщения работают только с объектами | 514 |
| Различение обобщений по аргументам типа | 514 |
| Обобщенный класс с двумя параметрами типа | 514 |
| Общая форма обобщенного класса | 516 |

| | |
|--|------------|
| Ограниченные типы | 516 |
| Использование шаблонов аргументов | 520 |
| Ограниченные шаблоны | 523 |
| Обобщенные методы | 525 |
| Обобщенные конструкторы | 528 |
| Обобщенные интерфейсы | 529 |
| Базовые типы и унаследованный код | 536 |
| Выведение типов с помощью ромбовидного оператора | 539 |
| Очистка | 541 |
| Ошибки неоднозначности | 542 |
| Ограничения на использование обобщений | 543 |
| Невозможность создания экземпляров параметров типа | 543 |
| Ограничения статических членов класса | 543 |
| Ограничения обобщенных массивов | 543 |
| Ограничения обобщенных исключений | 545 |
| Дальнейшее изучение обобщений | 545 |
| Глава 14. Лямбда-выражения и ссылки на методы | 547 |
| Знакомство с лямбда-выражениями | 549 |
| Основные сведения о лямбда-выражениях | 549 |
| Функциональные интерфейсы | 551 |
| Применение лямбда-выражений | 553 |
| Блочные лямбда-выражения | 558 |
| Обобщенные функциональные интерфейсы | 560 |
| Лямбда-выражения и захват переменных | 567 |
| Генерация исключений в лямбда-выражениях | 568 |
| Ссылки на методы | 570 |
| Ссылки на статические методы | 570 |
| Ссылки на методы экземпляров | 572 |
| Ссылки на конструкторы | 577 |
| Предопределенные функциональные интерфейсы | 579 |
| Глава 15. Модули | 583 |
| Знакомство с модулями | 585 |
| Простой пример модуля | 586 |
| Компиляция и выполнение первого примера модуля | 591 |
| Подробное знакомство с инструкциями <code>requires</code> и <code>exports</code> | 592 |
| Платформенные модули и пакет <code>java.base</code> | 593 |
| Унаследованный код и безымянный модуль | 595 |
| Выполнение экспорта для определенного модуля | 596 |
| Использование инструкции <code>requires transitive</code> | 597 |
| Использование служб | 602 |
| Общие сведения о службах и провайдерах служб | 603 |
| Ключевые слова, используемые при работе со службами | 603 |
| Пример использования модульной службы | 604 |
| Дополнительные возможности модулей | 611 |
| Открытые модули | 611 |
| Инструкция <code>opens</code> | 612 |

| | |
|--|------------|
| Инструкция <code>requires static</code> | 612 |
| Дополнительные сведения о модулях | 612 |
| Глава 16. Введение в Swing | 615 |
| Происхождение и философия Swing | 617 |
| Компоненты и контейнеры | 620 |
| Компоненты | 620 |
| Контейнеры | 621 |
| Панели контейнеров верхнего уровня | 621 |
| Менеджеры компоновки | 622 |
| Первая простая Swing-программа | 623 |
| Построчный анализ первой Swing-программы | 625 |
| Обработка событий Swing | 629 |
| События | 629 |
| Источники событий | 629 |
| Слушатели событий | 630 |
| Классы событий и интерфейсы слушателей | 630 |
| Использование компонента <code>JButton</code> | 631 |
| Работа с компонентом <code>JTextField</code> | 635 |
| Создание флажков с помощью компонента <code>JCheckBox</code> | 639 |
| Класс <code>JList</code> | 643 |
| Применение анонимных внутренних классов или лямбда-выражений для обработки событий | 653 |
| Глава 17. Введение в JavaFX | 657 |
| Базовые понятия JavaFX | 659 |
| Пакеты JavaFX | 659 |
| Классы <code>Stage</code> и <code>Scene</code> | 660 |
| Узлы и графы сцены | 660 |
| Панели компоновки | 661 |
| Класс <code>Application</code> и жизненный цикл приложения | 661 |
| Запуск приложения JavaFX | 661 |
| Каркас приложения JavaFX | 662 |
| Компиляция и выполнение программы JavaFX | 666 |
| Поток выполнения приложения | 666 |
| Простой компонент JavaFX: <code>Label</code> | 667 |
| Использование кнопок и событий | 669 |
| Основные сведения о событиях | 669 |
| Компонент <code>Button</code> | 670 |
| Обработка событий кнопки | 671 |
| Три других компонента JavaFX | 674 |
| Компонент <code>CheckBox</code> | 674 |
| Компонент <code>ListView</code> | 679 |
| Компонент <code>TextField</code> | 684 |
| Знакомство с эффектами и преобразованиями | 687 |
| Эффекты | 688 |
| Преобразования | 690 |
| Демонстрация эффектов и преобразований | 691 |
| Что дальше | 694 |

| | |
|--|-----|
| Приложение А. Ответы на вопросы и решения упражнений для самопроверки | 697 |
| Глава 1 | 698 |
| Глава 2 | 700 |
| Глава 3 | 701 |
| Глава 4 | 704 |
| Глава 5 | 705 |
| Глава 6 | 709 |
| Глава 7 | 714 |
| Глава 8 | 716 |
| Глава 9 | 718 |
| Глава 10 | 721 |
| Глава 11 | 724 |
| Глава 12 | 727 |
| Глава 13 | 730 |
| Глава 14 | 735 |
| Глава 15 | 738 |
| Глава 16 | 740 |
| Глава 17 | 745 |
| Приложение Б. Применение документирующих комментариев в Java | 751 |
| Дескрипторы javadoc | 752 |
| @author | 754 |
| {@code} | 754 |
| @deprecated | 754 |
| {@docRoot} | 754 |
| @exception | 754 |
| @hidden | 754 |
| {@index} | 755 |
| {@inheritDoc} | 755 |
| {@link} | 755 |
| {@linkplain} | 755 |
| {@literal} | 755 |
| @param | 756 |
| @provides | 756 |
| @return | 756 |
| @see | 756 |
| @serial | 756 |
| @serialData | 757 |
| @serialField | 757 |
| @since | 757 |
| @throws | 757 |
| @uses | 757 |
| {@value} | 757 |
| @version | 758 |
| Общая форма документирующих комментариев | 758 |
| Результат, выводимый утилитой javadoc | 758 |
| Пример использования документирующих комментариев | 758 |

| | |
|--|-----|
| Приложение В. Обзор технологии Java Web Start | 761 |
| Знакомство с Java Web Start | 762 |
| Развертывание Java Web Start | 763 |
| Для приложений JavaWS требуется JAR-файл | 763 |
| Подписанные приложения JavaWS | 764 |
| Использование файлов JNLP при работе с приложениями JavaWS | 765 |
| Связывание с файлом JNLP | 766 |
| Эксперименты с Java Web Start в локальной файловой системе | 767 |
| Создание JAR-файла для приложения ButtonDemo | 768 |
| Создание хранилища ключей и подписание файла ButtonDemo.jar | 768 |
| Создание файла JNLP для приложения ButtonDemo | 770 |
| Создание HTML-файла StartBD.html | 770 |
| Добавление файла ButtonDemo.jnlp в список исключений на панели управления Java | 771 |
| Выполнение приложения ButtonDemo в браузере | 771 |
| Запуск приложения JavaWS с помощью утилиты javaws | 772 |
| Использование Java Web Start для запуска апплетов | 772 |
| Приложение Г. Введение в JShell | 773 |
| Основы JShell | 774 |
| Вывод, редактирование и повторное выполнение кода | 776 |
| Добавление метода | 777 |
| Создание класса | 778 |
| Использование интерфейса | 779 |
| Оценка выражений и использование встроенных переменных | 780 |
| Импорт пакетов | 781 |
| Исключения | 782 |
| Другие команды JShell | 782 |
| Дальнейшее изучение JShell | 783 |
| Приложение Д. Дополнительные сведения о ключевых словах Java | 785 |
| Модификаторы transient и volatile | 786 |
| Оператор instanceof | 786 |
| Ключевое слово strictfp | 787 |
| Инструкция assert | 787 |
| Собственные методы | 788 |
| Другая форма ключевого слова this | 789 |
| Приложение Е. Знакомство с JDK 10 | 791 |
| Выведение типов локальных переменных | 793 |
| Выведение типов локальных переменных со ссылочными типами | 795 |
| Выведение типов локальных переменных и наследование | 796 |
| Выведение типов локальных переменных и обобщенные типы | 797 |
| Выведение типов локальных переменных в циклах for и блоках try | 798 |
| Ограничения ключевого слова var | 799 |
| Обновление схемы нумерации версий JDK и класс Runtime.Version | 799 |
| Предметный указатель | 803 |