

Содержание

Об авторе	27
О техническом редакторе	27
Книга для всех программистов	29
Введение	29
Что внутри	30
Коды примеров доступны в веб	30
Особые благодарности	30
Для дальнейшего изучения	31
Соглашения, принятые в этой книге	31
От издательства	32
Часть I. Язык Java	33
Глава 1. История и развитие языка Java	35
Происхождение языка Java	35
Зарождение современного программирования: язык C	36
Следующий шаг: язык C++	37
Предпосылки создания языка Java	38
Создание языка Java	38
Связь с языком C#	41
Как язык Java изменил Интернет	41
Апплеты Java	41
Безопасность	42
Переносимость	42
Магия Java: код виртуальной машины	42
Сервлеты: серверные программы Java	44
Терминология, связанная с Java	44
Простота	45
Объектная ориентированность	45
Устойчивость	45
Многопоточность	46
Архитектурная нейтральность	46
Интерпретируемость и высокая производительность	46
Распределенный характер	46
Динамический характер	47
Эволюция языка Java	47

Java SE 7	49
Культура инновации	50
Глава 2. Обзор языка Java	51
Объектно-ориентированное программирование	51
Две концепции	51
Абстракция	52
Три принципа ООП	52
Первый пример простой программы	58
Ввод кода программы	58
Компиляция программы	59
Более подробное рассмотрение первого примера программы	59
Второй пример короткой программы	62
Два управляющих оператора	63
Оператор <code>if</code>	63
Цикл <code>for</code>	65
Использование блоков кода	66
Вопросы лексики	67
Отступ	67
Идентификаторы	68
Литералы	68
Комментарии	68
Разделители	68
Ключевые слова Java	69
Библиотеки классов Java	70
Глава 3. Типы данных, переменные и массивы	71
Java – строго типизированный язык	71
Элементарные типы	71
Целочисленные значения	72
Тип <code>byte</code>	73
Тип <code>short</code>	73
Тип <code>int</code>	73
Тип <code>long</code>	73
Типы с плавающей точкой	74
Тип <code>float</code>	75
Тип <code>double</code>	75
Символы	75
Булевы значения	77
Более подробное рассмотрение литералов	78
Целочисленные литералы	78
Литералы с плавающей точкой	79
Булевы литералы	80
Символьные литералы	80
Строковые литералы	81
Переменные	81
Объявление переменной	81
Динамическая инициализация	82
Область видимости и продолжительность существования переменных	82

8 Содержание

Преобразование и приведение типов	85
Автоматическое преобразование типов в Java	85
Приведение несовместимых типов	85
Автоматическое повышение типа в выражениях	87
Правила повышения типа	88
Массивы	88
Одномерные массивы	88
Многомерные массивы	91
Альтернативный синтаксис объявления массивов	95
Несколько слов о строках	95
Замечание по поводу указателей для программистов на C/C++	96
Глава 4. Операторы	97
Арифметические операторы	97
Основные арифметические операторы	98
Оператор деления по модулю	99
Составные арифметические операторы с присваиванием	99
Инкремент и декремент	100
Побитовые операторы	101
Побитовые логические операторы	103
Сдвиг влево	105
Сдвиг вправо	106
Сдвиг вправо без учета знака	108
Операторы сравнения	110
Логические операторы	111
Сокращенные логические операторы	112
Оператор присваивания	113
Оператор ?	113
Приоритет операторов	114
Использование круглых скобок	115
Глава 5. Управляющие операторы	117
Операторы выбора	117
Оператор <code>if</code>	117
Оператор <code>switch</code>	120
Операторы цикла	124
Цикл <code>while</code>	125
Цикл <code>do-while</code>	126
Цикл <code>for</code>	129
Вложенные циклы	137
Операторы перехода	138
Использование оператора <code>break</code>	138
Использование оператора <code>continue</code>	141
Оператор <code>return</code>	143
Глава 6. Знакомство с классами	145
Основы классов	145
Общая форма класса	145
Простой класс	146
Объявление объектов	149

Подробное рассмотрение оператора <code>new</code>	150
Присваивание переменных объектных ссылок	151
Знакомство с методами	151
Добавление метода к классу <code>Box</code>	152
Возвращение значения	154
Добавление метода, принимающего параметры	155
Конструкторы	157
Конструкторы с параметрами	159
Ключевое слово <code>this</code>	160
Соккрытие переменной экземпляра	160
Сбор “мусора”	161
Метод <code>finalize()</code>	161
Класс <code>Stack</code>	162
Глава 7. Более пристальный взгляд на методы и классы	165
Перегрузка методов	165
Перегрузка конструкторов	168
Использование объектов в качестве параметров	170
Более пристальный взгляд на передачу аргументов	172
Возврат объектов	173
Рекурсия	174
Введение в управление доступом	176
Что такое <code>static</code>	180
Знакомство с ключевым словом <code>final</code>	182
Повторное рассмотрение массивов	182
Представление вложенных и внутренних классов	184
Описание класса <code>String</code>	186
Использование аргументов командной строки	188
Список аргументов переменной длины	189
Перегрузка методов с переменным количеством аргументов	192
Переменное количество аргументов и неопределенность	193
Глава 8. Наследование	195
Основы наследования	195
Доступ к членам и наследование	197
Более реальный пример	197
Переменная суперкласса может ссылаться на объект подкласса	199
Использование ключевого слова <code>super</code>	200
Использование ключевого слова <code>super</code> для вызова конструкторов суперкласса	200
Второе применение ключевого слова <code>super</code>	203
Создание многоуровневой иерархии	204
Порядок вызова конструкторов	207
Переопределение методов	208
Динамическая диспетчеризация методов	210
Для чего нужны переопределенные методы	212
Использование переопределения методов	212
Использование абстрактных классов	214
Использование ключевого слова <code>final</code> в сочетании с наследованием	216

10 Содержание

Использование ключевого слова <code>final</code> для предотвращения переопределения	216
Использование ключевого слова <code>final</code> для предотвращения наследования	217
Класс <code>Object</code>	218
Глава 9. Пакеты и интерфейсы	219
Пакеты	219
Определение пакета	220
Поиск пакетов и переменная среды <code>CLASSPATH</code>	220
Краткий пример пакета	221
Защита доступа	222
Пример защиты доступа	223
Импорт пакетов	225
Интерфейсы	227
Определение интерфейса	228
Реализация интерфейсов	229
Доступ к реализациям через ссылки на интерфейсы	229
Вложенные интерфейсы	231
Использование интерфейсов	232
Переменные в интерфейсах	235
Возможность расширения интерфейсов	237
Глава 10. Обработка исключений	239
Основы обработки исключений	239
Типы исключений	240
Необработанные исключения	240
Использование блоков <code>try</code> и <code>catch</code>	242
Отображение описания исключения	243
Множественные операторы <code>catch</code>	244
Вложенные операторы <code>try</code>	245
Встроенные исключения Java	251
Создание собственных подклассов исключений	252
Сцепленные исключения	254
Три новых средства исключений JDK 7	256
Использование исключений	257
Глава 11. Многопоточное программирование	259
Модель потоков Java	260
Приоритеты потоков	261
Синхронизация	262
Обмен сообщениями	262
Класс <code>Thread</code> и интерфейс <code>Runnable</code>	262
Главный поток	263
Создание потока	265
Реализация интерфейса <code>Runnable</code>	265
Расширение класса <code>Thread</code>	267
Выбор подхода	268
Создание множества потоков	268
Использование методов <code>isAlive()</code> и <code>join()</code>	269

Приоритеты потоков	271
Синхронизация	272
Использование синхронизированных методов	273
Оператор <code>synchronized</code>	275
Межпоточковые коммуникации	276
Взаимная блокировка	280
Приостановка, возобновление и останов потоков	282
Приостановка, возобновление и останов потоков в Java 1.1 и более ранних версиях	283
Современный способ приостановки, возобновления и остановки потоков	285
Получение состояния потока	287
Использование многопоточности	288
Глава 12. Перечисления, автоупаковка и аннотации (метаданные)	289
Перечисления	289
Основные понятия о перечислениях	289
Методы <code>values()</code> и <code>valueOf()</code>	291
Перечисления в Java являются типами классов	292
Перечисления наследуются от класса <code>Enum</code>	294
Еще один пример перечисления	296
Оболочки типов	297
Оболочки числовых типов	298
Автоупаковка	300
Автоупаковка и методы	300
Автоупаковка и распаковка в выражениях	301
Автоупаковка и распаковка значений классов <code>Boolean</code> и <code>Character</code>	303
Автоупаковка и распаковка помогают предотвратить ошибки	304
Предостережения	304
Аннотации (метаданные)	305
Основы аннотирования	305
Политика удержания аннотации	306
Получение аннотаций во время выполнения с использованием рефлексии	306
Второй пример применения рефлексии	309
Получение всех аннотаций	310
Интерфейс <code>AnnotatedElement</code>	311
Использование значений по умолчанию	311
Аннотация-маркер	313
Одночленные аннотации	313
Встроенные аннотации	315
Некоторые ограничения	316
Глава 13. Ввод-вывод, апплеты и другие темы	317
Основы ввода-вывода	317
Потоки	317
Байтовые и символьные потоки	318
Классы байтовых потоков	318

12 Содержание

Классы символьных потоков	319
Предопределенные потоки	320
Чтение консольного ввода	321
Чтение символов	321
Чтение строк	322
Запись консольного вывода	323
Класс <code>PrintWriter</code>	324
Чтение и запись файлов	325
Автоматическое закрытие файла	331
Основы организации апплетов	334
Модификаторы <code>transient</code> и <code>volatile</code>	337
Использование оператора <code>instanceof</code>	337
Машинно-зависимые методы	340
Проблемы, связанные с машинно-зависимыми методами	343
Использование ключевого слова <code>assert</code>	343
Параметры включения и отключения утверждений	346
Статический импорт	346
Вызов перегруженных конструкторов через <code>this()</code>	348
Глава 14. Обобщения	351
Что такое обобщения	351
Простой пример обобщения	352
Обобщения работают только с объектами	355
Отличие обобщенных типов в зависимости от аргументов типа	356
Обобщения повышают безопасность типов	356
Обобщенный класс с двумя параметрами типа	358
Общая форма обобщенного класса	359
Ограниченные типы	360
Использование шаблонов аргументов	362
Ограниченные шаблоны	364
Создание обобщенного метода	369
Обобщенные конструкторы	371
Обобщенные интерфейсы	371
Базовые типы и унаследованный код	373
Иерархии обобщенных классов	376
Использование обобщенного суперкласса	376
Обобщенный подкласс	378
Сравнение типов обобщенной иерархии во время выполнения	379
Приведение	381
Переопределение методов в обобщенном классе	381
Выведение типов и обобщения	382
Очистка	384
Методы-мосты	386
Ошибки неоднозначности	387
Некоторые ограничения обобщений	388
Нельзя создавать экземпляр типа параметра	388
Ограничения на статические члены	389
Ограничения обобщенных массивов	389
Ограничения обобщенных исключений	390

Часть II. Библиотека Java	391
Глава 15. Обработка строк	393
Конструкторы строк	394
Длина строки	395
Специальные строковые операции	396
Строковые литералы	396
Конкатенация строк	396
Конкатенация строк с другими типами данных	397
Преобразование строк и метод <code>toString()</code>	397
Извлечение символов	398
Метод <code>charAt()</code>	399
Метод <code>getChars()</code>	399
Метод <code>getBytes()</code>	399
Метод <code>toCharArray()</code>	400
Сравнение строк	400
Методы <code>equals()</code> и <code>equalsIgnoreCase()</code>	400
Метод <code>regionMatches()</code>	401
Методы <code>startsWith()</code> и <code>endsWith()</code>	401
Сравнение метода <code>equals()</code> и оператора <code>==</code>	402
Метод <code>compareTo()</code>	402
Поиск строк	404
Модификация строк	405
Метод <code>substring()</code>	405
Метод <code>concat()</code>	406
Метод <code>replace()</code>	406
Метод <code>trim()</code>	406
Преобразование данных с помощью метода <code>valueOf()</code>	407
Изменение регистра символов в строке	408
Дополнительные методы класса <code>String</code>	408
Класс <code>StringBuffer</code>	410
Конструкторы класса <code>StringBuffer</code>	410
Методы <code>length()</code> и <code>capacity()</code>	410
Метод <code>ensureCapacity()</code>	411
Метод <code>setLength()</code>	411
Методы <code>charAt()</code> и <code>setCharAt()</code>	411
Метод <code>getChars()</code>	412
Метод <code>append()</code>	412
Метод <code>insert()</code>	413
Метод <code>reverse()</code>	413
Методы <code>delete()</code> и <code>deleteCharAt()</code>	414
Метод <code>replace()</code>	414
Метод <code>substring()</code>	415
Дополнительные методы класса <code>StringBuffer</code>	415
Класс <code>StringBuilder</code>	416
Глава 16. Пакет <code>java.lang</code>	417
Оболочки элементарных типов	418
Класс <code>Number</code>	418
Классы <code>Double</code> и <code>Float</code>	418

Методы <code>isInfinite()</code> и <code>isNaN()</code>	422
Классы <code>Byte</code> , <code>Short</code> , <code>Integer</code> и <code>Long</code>	423
Преобразование чисел в строки и обратно	430
Класс <code>Character</code>	431
Дополнения к классу <code>Character</code> для поддержки кодовых точек <code>Unicode</code>	434
Класс <code>Boolean</code>	436
Класс <code>Void</code>	437
Класс <code>Process</code>	437
Класс <code>Runtime</code>	438
Управление памятью	439
Выполнение других программ	440
Класс <code>ProcessBuilder</code>	441
Класс <code>System</code>	444
Использование метода <code>currentTimeMills()</code> для измерения времени выполнения программы	446
Использование метода <code>arraycopy()</code>	446
Свойства окружения	447
Класс <code>Object</code>	447
Использование метода <code>clone()</code> и интерфейса <code>Cloneable</code>	448
Класс <code>Class</code>	450
Класс <code>ClassLoader</code>	453
Класс <code>Math</code>	453
Тригонометрические функции	454
Экспоненциальные функции	454
Функции округления	455
Прочие методы класса <code>Math</code>	456
Класс <code>StrictMath</code>	457
Класс <code>Compiler</code>	457
Классы <code>Thread</code> , <code>ThreadGroup</code> и интерфейс <code>Runnable</code>	457
Интерфейс <code>Runnable</code>	458
Класс <code>Thread</code>	458
Класс <code>ThreadGroup</code>	460
Классы <code>ThreadLocal</code> и <code>InheritableThreadLocal</code>	464
Класс <code>Package</code>	465
Класс <code>RuntimePermission</code>	466
Класс <code>Throwable</code>	466
Класс <code>SecurityManager</code>	466
Класс <code>StackTraceElement</code>	467
Класс <code>Enum</code>	468
Класс <code>ClassValue</code>	468
Интерфейс <code>CharSequence</code>	469
Интерфейс <code>Comparable</code>	469
Интерфейс <code>Appendable</code>	469
Интерфейс <code>Iterable</code>	470
Интерфейс <code>Readable</code>	470
Интерфейс <code>AutoCloseable</code>	470
Интерфейс <code>Thread.UncaughtExceptionHandler</code>	471
Вложенные пакеты <code>java.lang</code>	471
Пакет <code>java.lang.annotation</code>	471

Пакет <code>java.lang.instrument</code>	471
Пакет <code>java.lang.invoke</code>	471
Пакет <code>java.lang.management</code>	472
Пакет <code>java.lang.ref</code>	472
Пакет <code>java.lang.reflect</code>	472
Глава 17. Пакет <code>java.util</code>: инфраструктура	
Collections Framework	473
Обзор коллекций	474
Комплект JDK 5 изменил инфраструктуру Collections Framework	475
Обобщенные определения фундаментально изменили инфраструктуру коллекций	475
Средства автоматической упаковки используют элементарные типы	476
Стиль цикла “for-each”	476
Интерфейсы коллекций	476
Интерфейс <code>Collection</code>	477
Интерфейс <code>List</code>	479
Интерфейс <code>Set</code>	481
Интерфейс <code>SortedSet</code>	481
Интерфейс <code>NavigableSet</code>	482
Интерфейс <code>Queue</code>	483
Интерфейс <code>Deque</code>	484
Классы коллекций	486
Класс <code>ArrayList</code>	487
Класс <code>LinkedList</code>	490
Класс <code>HashSet</code>	491
Класс <code>LinkedHashSet</code>	492
Класс <code>TreeSet</code>	493
Класс <code>PriorityQueue</code>	494
Класс <code>ArrayDeque</code>	495
Класс <code>EnumSet</code>	496
Доступ к коллекциям через итератор	497
Использование интерфейса <code>Iterator</code>	498
Версия “for-each” цикла <code>for</code> как альтернатива итераторам	499
Использование пользовательских классов в коллекциях	500
Интерфейс <code>RandomAccess</code>	501
Работа с картами	502
Интерфейсы карт	502
Классы карт	507
Компараторы	511
Использование компараторов	512
Алгоритмы коллекций	514
Класс <code>Arrays</code>	519
Зачем нужны обобщенные коллекции	523
Унаследованные классы и интерфейсы	526
Интерфейс <code>Enumeration</code>	526
Класс <code>Vector</code>	527
Класс <code>Stack</code>	530
Класс <code>Dictionary</code>	532

16 Содержание

Класс Hashtable	533
Класс Properties	536
Использование методов store() и load()	539
Заключительные соображения по поводу коллекций	541

Глава 18. Пакет java.util: прочие служебные классы 543

Класс StringTokenizer	543
Класс BitSet	545
Класс Date	547
Класс Calendar	549
Класс GregorianCalendar	552
Класс TimeZone	553
Класс SimpleTimeZone	554
Класс Locale	555
Класс Random	556
Класс Observable	558
Интерфейс Observer	559
Пример использования интерфейса Observer	559
Классы Timer и TimerTask	562
Класс Currency	564
Класс Formatter	565
Конструкторы класса Formatter	565
Методы класса Formatter	566
Основы форматирования	567
Форматирование строк и символов	568
Форматирование чисел	569
Форматирование времени и даты	569
Спецификаторы %n и %%	571
Указание минимальной ширины поля	572
Указание точности	573
Использование флагов формата	574
Выравнивание вывода	574
Флаги пробела, +, 0 и (575
Флаг “запятая”	576
Флаг #	576
Параметры верхнего регистра	576
Использование индекса аргументов	577
Закрытие объекта класса Formatter	578
Подключение функции Java printf()	578
Класс Scanner	579
Конструкторы класса Scanner	579
Основы сканирования	580
Некоторые примеры применения класса Scanner	583
Установка разделителей	587
Прочие возможности класса Scanner	588
Классы ResourceBundle, ListResourceBundle и PropertyResourceBundle	589
Прочие служебные классы и интерфейсы	593
Вложенные пакеты java.util	593

Пакеты <code>java.util.concurrent</code> , <code>java.util.concurrent.atomic</code> , <code>java.util.concurrent.locks</code>	594
Пакет <code>java.util.jar</code>	594
Пакет <code>java.util.logging</code>	594
Пакет <code>java.util.prefs</code>	594
Пакет <code>java.util.regex</code>	594
Пакет <code>java.util.spi</code>	594
Пакет <code>java.util.zip</code>	594
Глава 19. Ввод-вывод: пакет <code>java.io</code>	595
Классы и интерфейсы ввода-вывода Java	595
Класс <code>File</code>	596
Каталоги	599
Использование интерфейса <code>FilenameFilter</code>	600
Альтернатива – метод <code>listFiles()</code>	601
Создание каталогов	601
Интерфейсы <code>AutoCloseable</code> , <code>Closeable</code> и <code>Flushable</code>	602
Исключения ввода-вывода	602
Два способа закрытия потока	603
Классы потоков	604
Байтовые потоки	604
Класс <code>InputStream</code>	605
Класс <code>OutputStream</code>	605
Класс <code>FileInputStream</code>	606
Класс <code>FileOutputStream</code>	608
Класс <code>ByteArrayInputStream</code>	610
Класс <code>ByteArrayOutputStream</code>	611
Фильтруемые потоки байтов	613
Буферизуемые потоки байтов	613
Символьные потоки	622
Класс <code>Reader</code>	622
Класс <code>Writer</code>	623
Класс <code>FileReader</code>	624
Класс <code>FileWriter</code>	625
Класс <code>CharArrayReader</code>	626
Класс <code>CharArrayWriter</code>	627
Класс <code>BufferedReader</code>	628
Класс <code>BufferedWriter</code>	629
Класс <code>PushbackReader</code>	629
Класс <code>PrintWriter</code>	631
Класс <code>Console</code>	631
Сериализация	633
Интерфейс <code>Serializable</code>	634
Интерфейс <code>Externalizable</code>	634
Интерфейс <code>ObjectOutput</code>	634
Класс <code>ObjectOutputStream</code>	635
Интерфейс <code>ObjectInput</code>	636
Класс <code>ObjectInputStream</code>	637
Пример сериализации	638
Преимущества потоков	639

Глава 20. Исследование NIO	641
Классы NIO	641
Основы NIO	642
Наборы символов и селекторы	645
Дополнения, внесенные в NIO (комплект JDK 7)	646
Интерфейс Path	646
Класс Files	647
Класс Paths	650
Интерфейсы атрибутов файла	650
Классы FileSystem, FileSystems и FileStore	652
Использование системы NIO	653
Использование системы NIO для канального ввода-вывода	653
Использование системы NIO для потокового ввода-вывода	662
Использование системы NIO для операций файловой системы	664
Примеры использования каналов до JDK 7	671
Чтение из файла до JDK 7	672
Запись в файл до JDK 7	674
Глава 21. Сеть	677
Основы работы с сетью	677
Сетевые классы и интерфейсы	678
Класс InetAddress	679
Методы-фабрики	679
Методы экземпляра	680
Классы Inet4Address и Inet6Address	681
Клиентские сокеты TCP/IP	681
Класс URL	684
Класс URLConnection	686
Класс HttpURLConnection	688
Класс URI	690
Файлы cookie	690
Серверные сокеты TCP/IP	691
Дейтаграммы	691
Класс DatagramSocket	692
Класс DatagramPacket	693
Пример работы с дейтаграммами	693
Глава 22. Класс Applet	695
Два типа апплетов	695
Основы апплетов	695
Класс Applet	697
Архитектура апплетов	699
Шаблон апплета	699
Инициализация и прекращение работы апплета	700
Переопределение метода update ()	702
Простые методы отображения апплетов	702
Запрос перерисовки	704
Простой апплет с баннером	705
Использование строки состояния	707

Дескриптор HTML APPLET	708
Передача параметров аплетам	709
Усовершенствование аплета баннера	711
Методы <code>getDocumentBase()</code> и <code>getCodeBase()</code>	712
Интерфейс <code>AppletContext</code> и метод <code>showDocument()</code>	713
Интерфейс <code>AudioClip</code>	715
Интерфейс <code>AppletStub</code>	715
Консольный вывод	715
Глава 23. Обработка событий	717
Два механизма обработки событий	717
Модель делегирования событий	718
События	718
Источники событий	718
Слушатели событий	719
Классы событий	719
Класс <code>ActionEvent</code>	721
Класс <code>AdjustmentEvent</code>	721
Класс <code>ComponentEvent</code>	722
Класс <code>ContainerEvent</code>	722
Класс <code>FocusEvent</code>	723
Класс <code>InputEvent</code>	724
Класс <code>ItemEvent</code>	724
Класс <code>KeyEvent</code>	725
Класс <code>MouseEvent</code>	726
Класс <code>MouseWheelEvent</code>	727
Класс <code>TextEvent</code>	728
Класс <code>WindowEvent</code>	728
Источники событий	729
Интерфейсы слушателей событий	730
Интерфейс <code>ActionListener</code>	731
Интерфейс <code>AdjustmentListener</code>	731
Интерфейс <code>ComponentListener</code>	731
Интерфейс <code>ContainerListener</code>	731
Интерфейс <code>FocusListener</code>	732
Интерфейс <code>ItemListener</code>	732
Интерфейс <code>KeyListener</code>	732
Интерфейс <code>MouseListener</code>	732
Интерфейс <code>MouseMotionListener</code>	732
Интерфейс <code>MouseWheelListener</code>	733
Интерфейс <code>TextListener</code>	733
Интерфейс <code>WindowFocusListener</code>	733
Интерфейс <code>WindowListener</code>	733
Использование модели делегирования событий	733
Обработка событий мыши	734
Обработка событий клавиатуры	736
Классы адаптеров	739
Вложенные классы	741
Анонимные вложенные классы	742

Глава 24. Введение в библиотеку AWT: работа с окнами, графикой и текстом	745
Классы библиотеки AWT	746
Основы окон	748
Класс Component	748
Класс Container	749
Класс Panel	749
Класс Window	749
Класс Frame	749
Класс Canvas	749
Работа с рамочными окнами	750
Установка размеров окна	750
Скрытие и отображение окна	750
Установка заголовка окна	750
Закрытие рамочного окна	751
Создание рамочного окна в апплете	751
Обработка событий в рамочном окне	753
Создание оконной программы	757
Отображение информации внутри окна	758
Работа с графикой	758
Рисование линий	759
Рисование прямоугольников	759
Рисование эллипсов и окружностей	761
Рисование дуг	762
Рисование многоугольников	763
Установка размеров графики	764
Работа с цветом	765
Методы класса Color	766
Установка режима рисования	767
Работа со шрифтами	769
Определение доступных шрифтов	770
Создание и выбор шрифта	771
Получение информации о шрифте	773
Управление выводом текста с использованием класса FontMetrics	774
Отображение множества строк текста	775
Центрирование текста	777
Выравнивание многострочного текста	778
Глава 25. Использование элементов управления, диспетчеров компоновки и меню библиотеки AWT	783
Основы элементов управления	783
Добавление и удаление элементов управления	784
Реакция на действия над элементами управления	784
Исключение HeadlessException	784
Метки	784
Использование кнопок	785
Обработка кнопок	786
Использование флажков	789

Обработка флажков	789
Класс <code>CheckboxGroup</code>	790
Элементы управления выбором	792
Обработка списков выбора	793
Использование списков	794
Обработка списков	796
Управление полосами прокрутки	797
Обработка полос прокрутки	799
Использование класса <code>TextField</code>	801
Обработка текстовых полей	802
Использование класса <code>TextArea</code>	803
Диспетчеры компоновки	805
Класс <code>FlowLayout</code>	805
Класс <code>BorderLayout</code>	808
Использование класса <code>Insets</code>	809
Класс <code>GridLayout</code>	811
Класс <code>CardLayout</code>	811
Класс <code>GridBagLayout</code>	814
Полосы меню и меню	819
Диалоговые окна	824
Класс <code>FileDialog</code>	828
Обработка событий при расширении компонентов библиотеки AWT	830
Расширение класса <code>Button</code>	831
Расширение класса <code>Checkbox</code>	832
Расширение группы флажков	833
Расширение класса <code>Choice</code>	834
Расширение класса <code>List</code>	834
Расширение класса <code>Scrollbar</code>	835
Несколько слов о переопределении метода <code>paint()</code>	836
Форматы файлов	837
Глава 26. Изображения	837
Основы работы с изображениями: создание, загрузка и отображение	838
Создание объекта класса <code>Image</code>	838
Загрузка изображения	838
Отображение изображения	839
Интерфейс <code>ImageObserver</code>	840
Двойная буферизация	842
Класс <code>MediaTracker</code>	844
Интерфейс <code>ImageProducer</code>	848
Класс <code>MemoryImageSource</code>	848
Интерфейс <code>ImageConsumer</code>	849
Класс <code>PixelGrabber</code>	850
Класс <code>ImageFilter</code>	851
Фильтр класса <code>CropImageFilter</code>	852
Фильтр класса <code>RGBImageFilter</code>	854
Аппликационная анимация	865
Дополнительные классы обработки изображений	868

Глава 27. Параллельные утилиты	869
Пакеты параллельного API	870
Пакет <code>java.util.concurrent</code>	870
Пакет <code>java.util.concurrent.atomic</code>	871
Пакет <code>java.util.concurrent.locks</code>	871
Использование объектов синхронизации	872
Класс <code>Semaphore</code>	872
Класс <code>CountDownLatch</code>	877
Класс <code>CyclicBarrier</code>	879
Класс <code>Exchanger</code>	881
Класс <code>Phaser</code>	883
Использование исполнителя	890
Простой пример исполнителя	891
Использование интерфейсов <code>Callable</code> и <code>Future</code>	892
Перечисление <code>TimeUnit</code>	895
Параллельные коллекции	896
Блокировки	896
Атомарные операции	899
Параллельное программирование при помощи инфраструктуры <code>Fork/Join Framework</code>	900
Основные классы инфраструктуры <code>Fork/Join Framework</code>	901
Стратегия “разделяй и властвуй”	904
Первый простой пример ветвления/объединения	905
Влияние уровня параллелизма	907
Пример применения класса <code>RecursiveTask<V></code>	910
Асинхронное выполнение задач	912
Отмена задачи	913
Определение состояния завершения задачи	913
Перезапуск задачи	913
Что исследовать	913
Некоторые советы относительно ветвления/объединения	915
Параллельные утилиты в сравнении с традиционным подходом в Java	916
Пакеты API ядра	917
Глава 28. Регулярные выражения и другие пакеты	917
Обработка регулярных выражений	919
Класс <code>Pattern</code>	919
Класс <code>Matcher</code>	920
Синтаксис регулярного выражения	921
Пример совпадения с шаблоном	921
Два варианта сопоставления с шаблоном	926
Изучение регулярных выражений	927
Рефлексия	927
Дистанционный вызов методов	931
Клиент-серверное приложение, использующее RMI	931
Форматирование текста	934
Класс <code>DateFormat</code>	934
Класс <code>SimpleDateFormat</code>	936

Часть III. Разработка программного обеспечения с использованием Java	939
Глава 29. Компоненты Java Bean	941
Что такое Java Bean	941
Преимущества компонентов Java Bean	942
Самодиагностика	942
Проектные шаблоны для свойств	942
Проектные шаблоны для событий	944
Методы и проектные шаблоны	944
Использование интерфейса BeanInfo	944
Связанные и ограниченные свойства	945
Постоянство	945
Конфигураторы	946
API Java Beans	946
Класс Introspector	948
Класс PropertyDescriptor	949
Класс EventSetDescriptor	949
Класс MethodDescriptor	949
Пример компонента Java Bean	949
Глава 30. Введение в библиотеку Swing	953
Истоки библиотеки Swing	953
Классы библиотеки Swing построены на основе библиотеки AWT	954
Две ключевые особенности библиотеки Swing	954
Компоненты библиотеки Swing являются облегченными	954
Библиотека Swing поддерживает подключаемый внешний вид	955
Архитектура MVC	955
Компоненты и контейнеры	956
Компоненты	957
Контейнеры	957
Панели контейнеров верхнего уровня	958
Пакеты библиотеки Swing	958
Простое приложение Swing	959
Обработка событий	963
Создание апплета Swing	966
Рисование с использованием библиотеки Swing	968
Основы рисования	968
Вычисление области рисования	969
Пример рисования	970
Глава 31. Дополнительные сведения о библиотеке Swing	973
Классы JLabel и ImageIcon	973
Класс JTextField	975
Кнопки библиотеки Swing	977
Класс JButton	977
Класс JToggleButton	980
Флажки	982
Переключатели	984

24 Содержание

Класс JTabbedPane	986
Класс JScrollPane	988
Класс JList	990
Класс JComboBox	993
Деревья	995
Класс JTable	999
Продолжайте изучать библиотеку Swing	1001
Глава 32. Сервлеты	1003
Предварительные сведения	1003
Жизненный цикл сервлета	1004
Возможности разработки сервлетов	1004
Использование контейнера Tomcat	1005
Простой сервлет	1007
Создание и компиляция исходного кода сервлета	1007
Запуск контейнера Tomcat	1008
Запуск веб-браузера и запрос сервлета	1008
Интерфейс Servlet API	1008
Пакет javax.servlet	1008
Интерфейс Servlet	1009
Интерфейс ServletConfig	1010
Интерфейс ServletContext	1010
Интерфейс ServletRequest	1011
Интерфейс ServletResponse	1011
Класс GenericServlet	1012
Класс ServletInputStream	1012
Класс ServletOutputStream	1012
Класс ServletException	1013
Чтение параметров сервлета	1013
Пакет javax.servlet.http	1014
Интерфейс HttpServletRequest	1015
Интерфейс HttpServletResponse	1016
Интерфейс HttpSession	1017
Интерфейс HttpSessionBindingListener	1018
Класс Cookie	1018
Класс HttpServlet	1019
Класс HttpSessionEvent	1020
Класс HttpSessionBindingEvent	1020
Обработка запросов и ответов HTTP	1021
Обработка запросов HTTP GET	1021
Обработка запросов HTTP POST	1022
Использование файлов cookie	1023
Отслеживание сеансов	1025
Часть IV. Применение Java	1027
Глава 33. Финансовые апплеты и сервлеты	1029
Расчет платежей по ссуде	1029
Поля апплета RegPay	1033
Метод init()	1034

Метод <code>makeGUI()</code>	1034
Метод <code>actionPerformed()</code>	1036
Метод <code>compute()</code>	1037
Расчет будущей стоимости вклада	1038
Расчет первоначальной суммы вклада, необходимой для достижения будущей суммы	1041
Расчет первоначальной суммы вклада, необходимой для получения желаемого годового дохода	1046
Нахождение максимального годового дохода для данной суммы вклада	1049
Нахождение остатка баланса по ссуде	1053
Создание финансовых сервлетов	1057
Преобразование апплета <code>RegPay</code> в сервлет	1057
Сервлет <code>RegPayS</code>	1058
Самостоятельная работа	1060
Загрузка данных из Интернета	1061
Глава 34. Создание утилиты загрузки на языке Java	1061
Обзор утилиты <code>Download Manager</code>	1062
Класс <code>Download</code>	1063
Переменные класса <code>Download</code>	1066
Конструктор класса <code>Download</code>	1066
Метод <code>download()</code>	1067
Метод <code>run()</code>	1067
Метод <code>stateChanged()</code>	1070
Методы действия и средства доступа	1071
Класс <code>ProgressRenderer</code>	1071
Класс <code>DownloadsTableModel</code>	1072
Метод <code>addDownload()</code>	1074
Метод <code>clearDownload()</code>	1074
Метод <code>getColumnClass()</code>	1074
Метод <code>getValueAt()</code>	1075
Метод <code>update()</code>	1075
Класс <code>DownloadManager</code>	1075
Переменные класса <code>DownloadManager</code>	1080
Конструктор класса <code>DownloadManager</code>	1081
Метод <code>verifyUrl()</code>	1081
Метод <code>tableSelectionChanged()</code>	1082
Метод <code>updateButtons()</code>	1082
Обработка событий действий	1083
Компиляция и запуск утилиты <code>Download Manager</code>	1083
Расширение утилиты <code>Download Manager</code>	1084
Приложение. Использование комментариев документации	1085
Дескрипторы утилиты <code>javadoc</code>	1085
Дескриптор <code>\$author</code>	1086
Дескриптор <code>{@code}</code>	1086
Дескриптор <code>@deprecated</code>	1086
Дескриптор <code>{@docRoot}</code>	1087
Дескриптор <code>@exception</code>	1087

Дескриптор {@inheritDoc}	1087
Дескриптор {@link}	1087
Дескриптор {@linkplain}	1087
Дескриптор {@literal}	1087
Дескриптор @param	1087
Дескриптор @return	1088
Дескриптор @see	1088
Дескриптор @serial	1088
Дескриптор @serialData	1088
Дескриптор @serialField	1088
Дескриптор @since	1089
Дескриптор @throws	1089
Дескриптор {@value}	1089
Дескриптор @version	1089
Общая форма комментариев документации	1089
Вывод утилиты javadoc	1090
Пример использования комментариев документации	1090
Предметный указатель	1093