

# Содержание

|  |    |
|--|----|
| <b>Предисловие</b>                               | 28 |
| <b>Часть I. Язык Java</b>                        | 31 |
| <b>Глава 1. История и развитие языка Java</b>    | 33 |
| Происхождение Java                               | 33 |
| Зарождение современного программирования: язык C | 34 |
| Следующий этап: язык C++                         | 35 |
| Предпосылки к созданию Java                      | 37 |
| Создание языка Java                              | 37 |
| Связь с языком C#                                | 39 |
| Каким образом язык Java изменил Интернет         | 40 |
| Апплеты на Java                                  | 40 |
| Безопасность                                     | 41 |
| Переносимость                                    | 41 |
| Чудо Java: байт-код                              | 41 |
| Сервлеты: серверные программы на Java            | 43 |
| Отличительные особенности Java                   | 43 |
| Простота   | 44 |
| Объектная ориентированность                      | 44 |
| Надежность                                       | 44 |
| Многопоточность                                  | 45 |
| Архитектурная нейтральность                      | 45 |
| Интерпретируемость и высокая производительность  | 46 |
| Распределенность                                 | 46 |
| Динамичность                                     | 46 |
| Эволюция языка Java                              | 46 |
| Версия Java SE 8                                 | 50 |
| Культура нововведений                            | 51 |
| <b>Глава 2. Краткий обзор Java</b>               | 53 |
| Объектно-ориентированное программирование        | 53 |
| Две методики                                     | 53 |
| Абстракция                                       | 54 |
| Три принципа ООП                                 | 54 |

|   |           |
|---|-----------|
| Первый пример простой программы                             | 60        |
| Ввод кода программы   | 61        |
| Компиляция программы  | 61        |
| Подробный анализ первого примера программы                  | 62        |
| Второй пример короткой программы                            | 65        |
| Два управляющих оператора                                   | 66        |
| Условный оператор <code>if</code>                           | 67        |
| Оператор цикла <code>for</code>                             | 68        |
| Использование блоков кода                                   | 69        |
| Вопросы лексики   | 70        |
| Пробелы   | 71        |
| Идентификаторы  | 71        |
| Литералы  | 71        |
| Комментарии   | 72        |
| Разделители   | 72        |
| Ключевые слова Java   | 72        |
| Библиотеки классов Java                                     | 73        |
| <b>Глава 3. Типы данных, переменные и массивы</b>           | <b>75</b> |
| Java – строго типизированный язык                           | 75        |
| Примитивные типы  | 75        |
| Целые числа   | 76        |
| Тип <code>byte</code>                                       | 77        |
| Тип <code>short</code>                                      | 77        |
| Тип <code>int</code>  | 77        |
| Тип <code>long</code>                                       | 78        |
| Числа с плавающей точкой                                    | 78        |
| Тип <code>float</code>                                      | 79        |
| Тип <code>double</code>                                     | 79        |
| Символы   | 80        |
| Логические значения   | 81        |
| Подробнее о литералах                                       | 82        |
| Целочисленные литералы                                      | 82        |
| Литералы с плавающей точкой                                 | 84        |
| Логические литералы   | 85        |
| Символьные литералы   | 85        |
| Строковые литералы  | 86        |
| Переменные  | 86        |
| Объявление переменной                                       | 86        |
| Динамическая инициализация                                  | 87        |
| Область и срок действия переменных                          | 87        |
| Преобразование и приведение типов                           | 90        |
| Автоматическое преобразование типов в Java                  | 90        |
| Приведение несовместимых типов                              | 91        |
| Автоматическое продвижение типов в выражениях               | 92        |
| Правила продвижения типов                                   | 93        |
| Массивы   | 94        |
| Одномерные массивы  | 94        |
| Многомерные массивы   | 97        |
| Альтернативный синтаксис объявления массивов                | 101       |
| Введение в символьные строки                                | 101       |
| Замечание по поводу указателей для программирующих на C/C++ | 102       |

## 8 Содержание

|   |     |
|---|-----|
| <b>Глава 4 . Операции</b>                         | 103 |
| Арифметические операции                           | 103 |
| Основные арифметические операции                  | 104 |
| Операция деления по модулю                        | 105 |
| Составные арифметические операции с присваиванием | 105 |
| Операции инкремента и декремента                  | 106 |
| Поразрядные операции                              | 108 |
| Поразрядные логические операции                   | 109 |
| Сдвиг влево                                       | 111 |
| Сдвиг вправо                                      | 113 |
| Беззнаковый сдвиг вправо                          | 114 |
| Поразрядные составные операции с присваиванием    | 116 |
| Операции отношения                                | 117 |
| Логические операции                               | 118 |
| Укороченные логические операции                   | 120 |
| Операция присваивания                             | 120 |
| Тернарная операция ?                              | 121 |
| Предшествование операций                          | 122 |
| Применение круглых скобок                         | 123 |
| <b>Глава 5. Управляющие операторы</b>             | 125 |
| Операторы выбора                                  | 125 |
| Условный оператор <code>if</code>                 | 125 |
| Оператор <code>switch</code>                      | 128 |
| Операторы цикла                                   | 133 |
| Цикл <code>while</code>                           | 133 |
| Цикл <code>do-while</code>                        | 135 |
| Цикл <code>for</code>                             | 138 |
| Вложенные циклы                                   | 147 |
| Операторы перехода                                | 147 |
| Применение оператора <code>break</code>           | 148 |
| Применение оператора <code>continue</code>        | 151 |
| Оператор <code>return</code>                      | 153 |
| <b>Глава 6. Введение в классы</b>                 | 155 |
| Основы классов                                    | 155 |
| Общая форма класса                                | 155 |
| Простой класс                                     | 156 |
| Объявление объектов                               | 159 |
| Подробное рассмотрение оператора <code>new</code> | 160 |
| Присваивание переменным ссылок на объекты         | 161 |
| Введение в методы                                 | 162 |
| Ввод метода в класс <code>Box</code>              | 163 |
| Возврат значений                                  | 164 |
| Ввод метода, принимающего параметры               | 166 |
| Конструкторы                                      | 168 |
| Параметризованные конструкторы                    | 170 |
| Ключевое слово <code>this</code>                  | 171 |
| Соккрытие переменной экземпляра                   | 172 |
| Сборка “мусора”                                   | 172 |
| Метод <code>finalize()</code>                     | 173 |
| Класс <code>Stack</code>                          | 174 |

|  |     |
|--|-----|
| <b>Глава 7. Подробное рассмотрение классов и методов</b>                     | 177 |
| Перегрузка методов   | 177 |
| Перегрузка конструкторов   | 180 |
| Применение объектов в качестве параметров                                    | 182 |
| Подробное рассмотрение особенностей передачи аргументов                      | 184 |
| Возврат объектов   | 186 |
| Рекурсия   | 187 |
| Введение в управление доступом   | 189 |
| Ключевое слово <code>static</code>   | 193 |
| Ключевое слово <code>final</code>  | 194 |
| Еще раз о массивах   | 195 |
| Вложенные и внутренние классы  | 197 |
| Краткий обзор класса <code>String</code>                                     | 200 |
| Применение аргументов командной строки                                       | 202 |
| Аргументы переменной длины   | 203 |
| Перегрузка методов с аргументами переменной длины                            | 206 |
| Аргументы переменной длины и неоднозначность                                 | 207 |
| <b>Глава 8. Наследование</b>   | 209 |
| Основы наследования  | 209 |
| Доступ к членам класса и наследование  | 211 |
| Более практический пример  | 212 |
| Переменная суперкласса может ссылаться на объект подкласса                   | 213 |
| Ключевое слово <code>super</code>  | 214 |
| Вызов конструкторов суперкласса с помощью ключевого слова <code>super</code> | 215 |
| Другое применение ключевого слова <code>super</code>                         | 218 |
| Создание многоуровневой иерархии   | 219 |
| Порядок вызова конструкторов   | 222 |
| Переопределение методов  | 223 |
| Динамическая диспетчеризация методов   | 225 |
| Назначение переопределенных методов  | 226 |
| Применение переопределения методов   | 227 |
| Применение абстрактных классов   | 228 |
| Ключевое слово <code>final</code> в сочетании с наследованием                | 231 |
| Предотвращение переопределения с помощью ключевого слова <code>final</code>  | 231 |
| Предотвращения наследования с помощью ключевого слова <code>final</code>     | 232 |
| Класс <code>Object</code>  | 233 |
| <b>Глава 9. Пакеты и интерфейсы</b>  | 235 |
| Пакеты   | 235 |
| Определение пакета   | 236 |
| Поиск пакетов и переменная окружения <code>CLASSPATH</code>                  | 236 |
| Краткий пример пакета  | 237 |
| Защита доступа   | 238 |
| Пример защиты доступа  | 239 |
| Импорт пакетов   | 242 |
| Интерфейсы   | 244 |

## 10 Содержание

|   |            |
|---|------------|
| Объявление интерфейса   | 245        |
| Реализация интерфейсов  | 246        |
| Вложенные интерфейсы  | 249        |
| Применение интерфейсов  | 250        |
| Переменные в интерфейсах  | 253        |
| Расширение интерфейсов  | 255        |
| Методы по умолчанию   | 255        |
| Основы применения методов по умолчанию                                | 257        |
| Более практический пример   | 258        |
| Вопросы множественного наследования                                   | 259        |
| Применение статических методов в интерфейсе                           | 260        |
| Заключительные соображения по поводу пакетов и интерфейсов            | 261        |
| <b>Глава 10. Обработка исключений</b>                                 | <b>263</b> |
| Основы обработки исключений   | 263        |
| Типы исключений   | 264        |
| Необрабатываемые исключения   | 265        |
| Использование блоков операторов <code>try</code> и <code>catch</code> | 266        |
| Вывод описания исключения   | 268        |
| Применение нескольких операторов <code>catch</code>                   | 268        |
| Вложенные операторы <code>try</code>                                  | 270        |
| Оператор <code>throw</code>   | 272        |
| Оператор <code>throws</code>  | 273        |
| Оператор <code>finally</code>   | 275        |
| Встроенные в Java исключения  | 276        |
| Создание собственных подклассов исключений                            | 278        |
| Цепочки исключений  | 280        |
| Недавно внедренные средства для обработки исключений                  | 282        |
| Применение исключений   | 284        |
| <b>Глава 11. Многопоточное программирование</b>                       | <b>285</b> |
| Модель потоков исполнения в Java                                      | 286        |
| Приоритеты потоков  | 287        |
| Синхронизация   | 288        |
| Обмен сообщениями   | 289        |
| Класс <code>Thread</code> и интерфейс <code>Runnable</code>           | 289        |
| Главный поток исполнения  | 290        |
| Создание потока исполнения  | 291        |
| Реализация интерфейса <code>Runnable</code>                           | 292        |
| Расширение класса <code>Thread</code>                                 | 294        |
| Выбор способа создания потоков исполнения                             | 295        |
| Создание многих потоков исполнения                                    | 295        |
| Применение методов <code>isAlive()</code> и <code>join()</code>       | 297        |
| Приоритеты потоков исполнения   | 299        |
| Синхронизация   | 300        |
| Применение синхронизированных методов                                 | 301        |
| Оператор <code>synchronized</code>                                    | 303        |
| Взаимодействие потоков исполнения                                     | 304        |
| Взаимная блокировка   | 309        |

|   |            |
|---|------------|
| Приостановка, возобновление и остановка потоков исполнения                                  | 311        |
| Получение состояния потока исполнения   | 313        |
| Применение многопоточности  | 315        |
| <b>Глава 12. Перечисления, автоупаковка и аннотации (метаданные)</b>                        | <b>317</b> |
| Перечисления  | 317        |
| Основные положения о перечислениях  | 317        |
| Методы <code>values()</code> и <code>valueOf()</code>                                       | 319        |
| Перечисления в Java относятся к типам классов   | 321        |
| Перечисления наследуются от класса <code>Enum</code>  | 323        |
| Еще один пример перечисления  | 325        |
| Оболочки типов  | 326        |
| Класс <code>Character</code>  | 327        |
| Класс <code>Boolean</code>  | 327        |
| Оболочки числовых типов   | 327        |
| Автоупаковка  | 328        |
| Автоупаковка и методы   | 329        |
| Автоупаковка и автораспаковка в выражениях  | 330        |
| Автоупаковка и распаковка значений из классов <code>Boolean</code> и <code>Character</code> | 332        |
| Автоупаковка и автораспаковка помогают предотвратить ошибки                                 | 333        |
| Предупреждение  | 333        |
| Аннотации (метаданные)  | 334        |
| Основы аннотирования программ   | 334        |
| Правила удержания аннотаций   | 335        |
| Получение аннотаций во время выполнения с помощью рефлексии                                 | 336        |
| Второй пример применения рефлексии  | 338        |
| Получение всех аннотаций  | 339        |
| Интерфейс <code>AnnotatedElement</code>   | 341        |
| Использование значений по умолчанию   | 341        |
| Аннотации-маркеры   | 343        |
| Одночленные аннотации   | 343        |
| Встроенные аннотации  | 345        |
| Типовые аннотации   | 347        |
| Повторяющиеся аннотации   | 352        |
| Некоторые ограничения на аннотации  | 353        |
| <b>Глава 13. Ввод-вывод, апплеты и прочие вопросы</b>                                       | <b>355</b> |
| Основы ввода-вывода   | 355        |
| Потоки ввода-вывода   | 356        |
| Потоки ввода-вывода байтов и символов   | 356        |
| Предопределенные потоки ввода-вывода  | 359        |
| Чтение данных, вводимых с консоли   | 359        |
| Чтение символов   | 360        |
| Чтение символьных строк   | 361        |
| Запись данных, выводимых на консоль   | 362        |
| Класс <code>PrintWriter</code>  | 363        |
| Чтение и запись данных в файлы  | 364        |
| Автоматическое закрытие файла   | 370        |
| Основы создания апплетов  | 374        |

## 12 Содержание

|   |            |
|---|------------|
| Модификаторы доступа <code>transient</code> и <code>volatile</code> | 377        |
| Применение оператора <code>instanceof</code>                        | 378        |
| Модификатор доступа <code>strictfp</code>                           | 380        |
| Платформенно-ориентированные методы                                 | 381        |
| Трудности, связанные с платформенно-ориентированными методами       | 384        |
| Применение ключевого слова <code>assert</code>                      | 385        |
| Параметры включения и отключения режима проверки утверждений        | 388        |
| Статический импорт  | 388        |
| Вызов перегружаемых конструкторов по ссылке <code>this()</code>     | 391        |
| Компактные профили Java API   | 393        |
| <b>Глава 14. Обобщения</b>  | <b>395</b> |
| Что такое обобщения   | 396        |
| Простой пример обобщения  | 396        |
| Обобщения действуют только со ссылочными типами                     | 400        |
| Обобщенные типы различаются по аргументам типа                      | 400        |
| Каким образом обобщения повышают типовую безопасность               | 401        |
| Обобщенный класс с двумя параметрами типа                           | 403        |
| Общая форма обобщенного класса                                      | 404        |
| Ограниченные типы   | 405        |
| Применение метасимвольных аргументов                                | 407        |
| Ограниченные метасимвольные аргументы                               | 410        |
| Создание обобщенного метода   | 414        |
| Обобщенные конструкторы   | 417        |
| Обобщенные интерфейсы   | 417        |
| Базовые типы и унаследованный код                                   | 420        |
| Иерархии обобщенных классов   | 422        |
| Применение обобщенного суперкласса                                  | 422        |
| Обобщенный подкласс   | 424        |
| Сравнение типов в обобщенной иерархии во время выполнения           | 425        |
| Приведение типов  | 428        |
| Переопределение методов в обобщенном классе                         | 428        |
| Выведение типов и обобщения   | 429        |
| Стирание  | 431        |
| Мостовые методы   | 431        |
| Ошибки неоднозначности  | 433        |
| Некоторые ограничения, присущие обобщениям                          | 434        |
| Получить экземпляр по параметру типа нельзя                         | 434        |
| Ограничения на статические члены                                    | 435        |
| Ограничения на обобщенные массивы                                   | 435        |
| Ограничения на обобщенные исключения                                | 436        |
| <b>Глава 15. Лямбда-выражения</b>                                   | <b>437</b> |
| Введение в лямбда-выражения   | 438        |
| Основные положения о лямбда-выражениях                              | 438        |
| Функциональные интерфейсы   | 439        |
| Некоторые примеры лямбда-выражений                                  | 440        |
| Блочные лямбда-выражения  | 444        |
| Обобщенные функциональные интерфейсы                                | 446        |
| Передача лямбда-выражений в качестве аргументов                     | 447        |
| Лямбда-выражения и исключения                                       | 450        |

|   |            |
|---|------------|
| Лямбда-выражения и захват переменных                                | 451        |
| Ссылки на методы  | 453        |
| Ссылки на статические методы  | 453        |
| Ссылки на методы экземпляра   | 454        |
| Ссылки на обобщенные методы   | 458        |
| Ссылки на конструкторы  | 460        |
| Предопределенные функциональные интерфейсы                          | 465        |
| <b>Часть II. Библиотека Java</b>                                    | <b>467</b> |
| <b>Глава 16. Обработка символьных строк</b>                         | <b>469</b> |
| Конструкторы символьных строк                                       | 470        |
| Длина символьной строки   | 472        |
| Специальные строковые операции                                      | 472        |
| Строковые литералы  | 472        |
| Сцепление строк   | 473        |
| Сцепление символьных строк с другими типами данных                  | 473        |
| Преобразование символьных строк и метод <code>toString()</code>     | 474        |
| Извлечение символов   | 475        |
| Метод <code>charAt()</code>   | 475        |
| Метод <code>getChars()</code>                                       | 476        |
| Метод <code>getBytes()</code>                                       | 476        |
| Метод <code>toCharArray()</code>                                    | 477        |
| Сравнение символьных строк  | 477        |
| Методы <code>equals()</code> и <code>equalsIgnoreCase()</code>      | 477        |
| Метод <code>regionMatches()</code>                                  | 478        |
| Методы <code>startsWith()</code> и <code>endsWith()</code>          | 478        |
| Метод <code>equals()</code> в сравнении с операцией <code>==</code> | 479        |
| Метод <code>compareTo()</code>                                      | 480        |
| Поиск в символьных строках  | 481        |
| Видоизменение символьных строк                                      | 482        |
| Метод <code>substring()</code>                                      | 483        |
| Метод <code>concat()</code>   | 484        |
| Метод <code>replace()</code>  | 484        |
| Метод <code>trim()</code>   | 484        |
| Преобразование данных методом <code>valueOf()</code>                | 485        |
| Изменение регистра символов в строке                                | 486        |
| Соединение символьных строк   | 487        |
| Дополнительные методы из класса <code>String</code>                 | 487        |
| Класс <code>StringBuffer</code>                                     | 489        |
| Конструкторы класса <code>StringBuffer</code>                       | 489        |
| Методы <code>length()</code> и <code>capacity()</code>              | 490        |
| Метод <code>ensureCapacity()</code>                                 | 490        |
| Метод <code>setLength()</code>                                      | 491        |
| Методы <code>charAt()</code> и <code>setCharAt()</code>             | 491        |
| Метод <code>getChars()</code>                                       | 492        |
| Метод <code>append()</code>   | 492        |
| Метод <code>insert()</code>   | 493        |
| Метод <code>reverse()</code>  | 493        |
| Методы <code>delete()</code> и <code>deleteCharAt()</code>          | 494        |



## 14 Содержание

|   |     |
|---|-----|
| Метод <code>replace()</code>                              | 494 |
| Метод <code>substring()</code>                            | 495 |
| Дополнительные методы из класса <code>StringBuffer</code> | 495 |
| Класс <code>StringBuilder</code>                          | 496 |

### Глава 17. Пакет `java.lang` 497

|  |     |
|--|-----|
| Оболочки примитивных типов   | 498 |
| Класс <code>Number</code>  | 498 |
| Классы <code>Double</code> и <code>Float</code>  | 498 |
| Методы <code>isInfinite()</code> и <code>isNaN()</code>                                  | 503 |
| Классы <code>Byte</code> , <code>Short</code> , <code>Integer</code> и <code>Long</code> | 503 |
| Класс <code>Character</code>   | 514 |
| Дополнения класса <code>Character</code> для поддержки кодовых точек в Юникоде           | 517 |
| Класс <code>Boolean</code>   | 519 |
| Класс <code>Void</code>  | 520 |
| Класс <code>Process</code>   | 520 |
| Класс <code>Runtime</code>   | 521 |
| Управление памятью   | 523 |
| Выполнение других программ   | 524 |
| Класс <code>ProcessBuilder</code>  | 525 |
| Класс <code>System</code>  | 528 |
| Измерение времени выполнения программы методом <code>currentTimeMills()</code>           | 530 |
| Применение метода <code>arraycopy()</code>   | 531 |
| Свойства окружения   | 532 |
| Класс <code>Object</code>  | 532 |
| Применение метода <code>clone()</code> и интерфейса <code>Cloneable</code>               | 533 |
| Класс <code>Class</code>   | 535 |
| Класс <code>ClassLoader</code>   | 538 |
| Класс <code>Math</code>  | 539 |
| Тригонометрические функции   | 539 |
| Экспоненциальные функции   | 540 |
| Функции округления   | 540 |
| Прочие методы из класса <code>Math</code>  | 542 |
| Класс <code>StrictMath</code>  | 544 |
| Класс <code>Compiler</code>  | 545 |
| Классы <code>Thread</code> , <code>ThreadGroup</code> и интерфейс <code>Runnable</code>  | 545 |
| Интерфейс <code>Runnable</code>  | 545 |
| Класс <code>Thread</code>  | 545 |
| Класс <code>ThreadGroup</code>   | 548 |
| Классы <code>ThreadLocal</code> и <code>InheritableThreadLocal</code>                    | 553 |
| Класс <code>Package</code>   | 553 |
| Класс <code>RuntimePermission</code>   | 555 |
| Класс <code>Throwable</code>   | 555 |
| Класс <code>SecurityManager</code>   | 555 |
| Класс <code>StackTraceElement</code>   | 555 |
| Класс <code>Enum</code>  | 556 |
| Класс <code>ClassValue</code>  | 557 |
| Интерфейс <code>CharSequence</code>  | 558 |
| Интерфейс <code>Comparable</code>  | 558 |
| Интерфейс <code>Appendable</code>  | 559 |

|  |            |
|--|------------|
| Интерфейс Iterable   | 559        |
| Интерфейс Readable   | 560        |
| Интерфейс AutoCloseable  | 560        |
| Интерфейс Thread.UncaughtExceptionHandler                        | 560        |
| Подпакеты из пакета в java.lang                                  | 561        |
| Пакет java.lang.annotation                                       | 561        |
| Пакет java.lang.instrument                                       | 561        |
| Пакет java.lang.invoke   | 561        |
| Пакет java.lang.management                                       | 561        |
| Пакет java.lang.ref  | 562        |
| Пакет java.lang.reflect  | 562        |
| <b>Глава 18. Пакет java.util, часть I. Collections Framework</b> | <b>563</b> |
| Краткий обзор коллекций  | 564        |
| Изменения каркаса коллекций в версии JDK 5                       | 566        |
| Обобщения коренным образом изменили каркас коллекций             | 566        |
| В средствах автоматической упаковки используются                 |            |
| примитивные типы данных  | 567        |
| Цикл for в стиле for each  | 567        |
| Интерфейсы коллекций   | 567        |
| Интерфейс Collection   | 568        |
| Интерфейс List   | 571        |
| Интерфейс Set  | 573        |
| Интерфейс SortedSet  | 573        |
| Интерфейс NavigableSet   | 574        |
| Интерфейс Queue  | 576        |
| Интерфейс Dequeue  | 577        |
| Классы коллекций   | 579        |
| Класс ArrayList  | 580        |
| Класс LinkedList   | 584        |
| Класс HashSet  | 585        |
| Класс LinkedHashSet  | 587        |
| Класс TreeSet  | 587        |
| Класс PriorityQueue  | 588        |
| Класс ArrayDeque   | 589        |
| Класс EnumSet  | 590        |
| Доступ к коллекциям через итератор                               | 591        |
| Применение интерфейса Iterator                                   | 593        |
| Цикл for в стиле for each как альтернатива итераторам            | 595        |
| Итераторы-разделители  | 596        |
| Сохранение объектов пользовательских классов в коллекциях        | 599        |
| Интерфейс RandomAccess   | 600        |
| Обращение с отображениями  | 601        |
| Интерфейсы отображений   | 601        |
| Классы отображений   | 609        |
| Компараторы  | 614        |
| Применение компараторов  | 617        |
| Алгоритмы коллекций  | 622        |
| Массивы  | 629        |
| Унаследованные классы и интерфейсы                               | 634        |

## 16 Содержание

|   |            |
|---|------------|
| Интерфейс Enumeration   | 635        |
| Класс Vector  | 635        |
| Класс Stack   | 639        |
| Класс Dictionary  | 641        |
| Класс Properties  | 645        |
| Применение методов store () и load ()                                 | 649        |
| Заключительные соображения по поводу коллекций                        | 650        |
| <b>Глава 19. Пакет java.util, часть II. Прочие служебные классы</b>   | <b>651</b> |
| Класс StringTokenizer   | 651        |
| Класс BitSet  | 653        |
| Классы Optional, OptionalDouble, OptionalInt и OptionalLong           | 656        |
| Класс Date  | 659        |
| Класс Calendar  | 661        |
| Класс GregorianCalendar   | 664        |
| Класс TimeZone  | 666        |
| Класс SimpleTimeZone  | 667        |
| Класс Locale  | 668        |
| Класс Random  | 670        |
| Класс Observable  | 672        |
| Интерфейс Observer  | 673        |
| Пример наблюдения за объектами  | 674        |
| Классы Timer и TimerTask  | 676        |
| Класс Currency  | 678        |
| Класс Formatter   | 680        |
| Конструкторы класса Formatter   | 680        |
| Методы из класса Formatter  | 681        |
| Основы форматирования   | 682        |
| Форматирование строк и символов                                       | 684        |
| Форматирование чисел  | 684        |
| Форматирование времени и даты   | 685        |
| Спецификаторы формата %n и %%   | 687        |
| Указание минимальной ширины поля                                      | 687        |
| Указание точности   | 689        |
| Применение признаков формата  | 689        |
| Выравнивание выводимых данных   | 690        |
| Признаки пробела, +, 0 и (  | 691        |
| Признак запятой   | 692        |
| Признак #   | 692        |
| Прописные формы спецификаторов формата                                | 692        |
| Применение индекса аргумента  | 693        |
| Закрытие объекта типа Formatter                                       | 694        |
| Аналог функции printf () в Java                                       | 695        |
| Класс Scanner   | 695        |
| Конструкторы класса Scanner   | 695        |
| Некоторые примеры применения класса Scanner                           | 700        |
| Установка разделителей  | 704        |
| Прочие средства класса Scanner  | 705        |
| Классы ResourceBundle, ListResourceBundle<br>и PropertyResourceBundle | 706        |

|   |     |
|---|-----|
| Прочие служебные классы и интерфейсы  | 710 |
| Пакеты <code>java.util.concurrent</code> , <code>java.util.concurrent.atomic</code> , <code>java.util.concurrent.locks</code> | 712 |
| Пакет <code>java.util.function</code>   | 712 |
| Пакет <code>java.util.jar</code>  | 716 |
| Пакет <code>java.util.logging</code>  | 716 |
| Пакет <code>java.util.prefs</code>  | 716 |
| Пакет <code>java.util.regex</code>  | 716 |
| Пакет <code>java.util.spi</code>  | 716 |
| Пакет <code>java.util.stream</code>   | 716 |
| Пакет <code>java.util.zip</code>  | 716 |
| <b>Глава 20. Пакет <code>java.io</code> для ввода-вывода</b>  | 717 |
| Классы и интерфейсы ввода-вывода  | 718 |
| Класс <code>File</code>   | 718 |
| Каталоги  | 722 |
| Применение интерфейса <code>FilenameFilter</code>   | 723 |
| Альтернативный метод <code>listFiles()</code>   | 724 |
| Создание каталогов  | 724 |
| Интерфейсы <code>AutoCloseable</code> , <code>Closeable</code> и <code>Flushable</code>                                       | 724 |
| Исключения ввода-вывода   | 725 |
| Два способа закрытия потока ввода-вывода  | 726 |
| Классы потоков ввода-вывода   | 727 |
| Потоки ввода-вывода байтов  | 728 |
| Класс <code>InputStream</code>  | 728 |
| Класс <code>OutputStream</code>   | 729 |
| Класс <code>FileInputStream</code>  | 730 |
| Класс <code>FileOutputStream</code>   | 732 |
| Класс <code>ByteArrayInputStream</code>   | 734 |
| Класс <code>ByteArrayOutputStream</code>  | 736 |
| Фильтруемые потоки ввода-вывода байтов  | 737 |
| Буферизованные потоки ввода-вывода байтов   | 737 |
| Потоки ввода-вывода символов  | 748 |
| Класс <code>Reader</code>   | 749 |
| Класс <code>Writer</code>   | 750 |
| Класс <code>FileReader</code>   | 751 |
| Класс <code>FileWriter</code>   | 751 |
| Класс <code>CharArrayReader</code>  | 752 |
| Класс <code>CharArrayWriter</code>  | 754 |
| Класс <code>BufferedReader</code>   | 755 |
| Класс <code>BufferedWriter</code>   | 756 |
| Класс <code>PushbackReader</code>   | 757 |
| Класс <code>PrintWriter</code>  | 758 |
| Класс <code>Console</code>  | 759 |
| Сериализация  | 761 |
| Интерфейс <code>Serializable</code>   | 762 |
| Интерфейс <code>Externalizable</code>   | 762 |
| Интерфейс <code>ObjectOutput</code>   | 763 |
| Класс <code>ObjectOutputStream</code>   | 763 |
| Интерфейс <code>ObjectInput</code>  | 764 |

## 18 Содержание

|  |            |
|--|------------|
| Класс <code>ObjectInputStream</code>   | 765        |
| Пример сериализации  | 767        |
| Преимущества потоков ввода-вывода  | 768        |
| <b>Глава 21. Система ввода-вывода NIO</b>  | <b>769</b> |
| Классы системы ввода-вывода NIO  | 769        |
| Основные положения о системе ввода-вывода NIO                                      | 770        |
| Буфера   | 770        |
| Каналы   | 772        |
| Наборы символов и селекторы  | 774        |
| Усовершенствования в системе NIO, начиная с версии JDK 7                           | 774        |
| Интерфейс <code>Path</code>  | 774        |
| Класс <code>Files</code>   | 776        |
| Класс <code>Paths</code>   | 779        |
| Интерфейсы атрибутов файлов  | 780        |
| Классы <code>FileSystem</code> , <code>FileSystems</code> и <code>FileStore</code> | 782        |
| Применение системы ввода-вывода NIO  | 782        |
| Применение системы NIO для канального ввода-вывода                                 | 783        |
| Применение системы NIO для потокового ввода-вывода                                 | 793        |
| Применение системы ввода-вывода NIO для операций в файловой системе                | 796        |
| Примеры организации канального ввода-вывода до версии JDK 7                        | 804        |
| Чтение из файла до версии JDK 7  | 804        |
| Запись в файл версии до JDK 7  | 807        |
| <b>Глава 22. Работа в сети</b>   | <b>811</b> |
| Основы работы в сети   | 811        |
| Сетевые классы и интерфейсы  | 813        |
| Класс <code>InetAddress</code>   | 813        |
| Фабричные методы   | 814        |
| Методы экземпляра  | 815        |
| Классы <code>Inet4Address</code> и <code>Inet6Address</code>                       | 815        |
| Клиентские сокеты по протоколу TCP/IP  | 816        |
| Класс <code>URL</code>   | 819        |
| Класс <code>URLConnection</code>   | 821        |
| Класс <code>HttpURLConnection</code>   | 824        |
| Класс <code>URI</code>   | 826        |
| Cookie-файлы   | 826        |
| Серверные сокеты по протоколу TCP/IP   | 827        |
| Дейтаграммы  | 827        |
| Класс <code>DatagramSocket</code>  | 828        |
| Класс <code>DatagramPacket</code>  | 829        |
| Пример обработки дейтаграмм  | 830        |
| <b>Глава 23. Класс <code>Applet</code></b>   | <b>833</b> |
| Два типа апплетов  | 833        |
| Основы разработки апплетов   | 834        |
| Класс <code>Applet</code>  | 835        |
| Структура апплетов   | 837        |
| Скелет апплета   | 838        |
| Инициализация и прекращение работы апплета   | 839        |

|   |            |
|---|------------|
| Переопределение метода update ()                    | 840        |
| Простые методы воспроизведения апплетов             | 841        |
| Запрос на повторное воспроизведение                 | 843        |
| Простой апплет с баннером                           | 844        |
| Применение строки состояния                         | 846        |
| HTML-дескриптор APPLET                              | 847        |
| Передача параметров апплетам                        | 848        |
| Усовершенствование апплета, воспроизводящего баннер | 850        |
| Методы getDocumentBase () и getCodeBase ()          | 851        |
| Интерфейс AppletContext и метод showDocument ()     | 852        |
| Интерфейс AudioClip                                 | 854        |
| Интерфейс AppletStub                                | 854        |
| Консольный вывод                                    | 854        |
| <b>Глава 24. Обработка событий</b>                  | <b>855</b> |
| Два механизма обработки событий                     | 855        |
| Модель делегирования событий                        | 856        |
| События   | 856        |
| Источники событий                                   | 857        |
| Приемники событий                                   | 857        |
| Классы событий                                      | 858        |
| Класс ActionEvent                                   | 859        |
| Класс AdjustmentEvent                               | 860        |
| Класс ComponentEvent                                | 861        |
| Класс ContainerEvent                                | 861        |
| Класс FocusEvent                                    | 862        |
| Класс InputEvent                                    | 863        |
| Класс ItemEvent                                     | 864        |
| Класс KeyEvent                                      | 864        |
| Класс MouseEvent                                    | 865        |
| Класс MouseWheelEvent                               | 867        |
| Класс TextEvent                                     | 868        |
| Класс WindowEvent                                   | 868        |
| Источники событий                                   | 870        |
| Интерфейсы приемников событий                       | 870        |
| Интерфейс ActionListener                            | 871        |
| Интерфейс AdjustmentListener                        | 871        |
| Интерфейс ComponentListener                         | 872        |
| Интерфейс ContainerListener                         | 872        |
| Интерфейс FocusListener                             | 872        |
| Интерфейс ItemListener                              | 872        |
| Интерфейс KeyListener                               | 872        |
| Интерфейс MouseListener                             | 873        |
| Интерфейс MouseMotionListener                       | 873        |
| Интерфейс MouseWheelListener                        | 873        |
| Интерфейс TextListener                              | 873        |
| Интерфейс WindowFocusListener                       | 873        |
| Интерфейс WindowListener                            | 873        |
| Применение модели делегирования событий             | 874        |
| Обработка событий от мыши                           | 874        |
| Обработка событий от клавиатуры                     | 877        |

|   |            |
|---|------------|
| Классы адаптеров  | 880        |
| Внутренние классы   | 882        |
| Анонимные внутренние классы   | 883        |
| <b>Глава 25. Введение в библиотеку AWT: работа с окнами, графикой и текстом</b>                   | <b>885</b> |
| Классы библиотеки AWT   | 886        |
| Основные положения об окнах   | 888        |
| Класс Component   | 889        |
| Класс Container   | 889        |
| Класс Panel   | 889        |
| Класс Window  | 890        |
| Класс Frame   | 890        |
| Класс Canvas  | 890        |
| Работа с обрамляющими окнами  | 890        |
| Установка размеров окна   | 891        |
| Соккрытие и отображение окна  | 891        |
| Установка заголовка окна  | 891        |
| Заккрытие обрамляющего окна   | 891        |
| Создание обрамляющего окна в апплете, построенном на основе библиотеки AWT                        | 892        |
| Обработка событий в обрамляющем окне  | 894        |
| Создание оконной прикладной программы   | 898        |
| Отображение информации в окне   | 899        |
| Поддержка графики   | 900        |
| Рисование линий   | 900        |
| Рисование прямоугольников   | 900        |
| Рисование эллипсов и окружностей  | 901        |
| Рисование дуг   | 901        |
| Рисование многоугольников   | 901        |
| Демонстрация методов рисования  | 902        |
| Изменение размеров графики  | 902        |
| Работа с цветом   | 904        |
| Методы из класса Color  | 905        |
| Установка режима рисования  | 907        |
| Работа со шрифтами  | 909        |
| Определение доступных шрифтов   | 910        |
| Создание и выбор шрифта   | 911        |
| Получение сведений о шрифте   | 913        |
| Управление форматированием выводимого текста  | 914        |
| Отображение многострочного текста   | 915        |
| Центровка текста  | 917        |
| Выравнивание многострочного текста  | 918        |
| <b>Глава 26. Применение элементов управления, диспетчеров компоновки и меню из библиотеки AWT</b> | <b>923</b> |
| Основные положения об элементах управления  | 924        |
| Ввод и удаление элементов управления  | 924        |
| Реагирование на элементы управления   | 925        |
| Исключение типа HeadlessException   | 925        |

|   |             |
|---|-------------|
| Метки   | 925         |
| Экранные кнопки   | 926         |
| Обработка событий от кнопок                                     | 927         |
| Флажки  | 930         |
| Обработка событий от флажков                                    | 931         |
| Кнопки-переключатели  | 932         |
| Элементы управления выбором                                     | 934         |
| Обработка событий от раскрывающихся списков                     | 935         |
| Использование списков   | 936         |
| Обработка событий от списков                                    | 938         |
| Управление полосами прокрутки                                   | 939         |
| Обработка событий от полос прокрутки                            | 941         |
| Текстовые поля  | 943         |
| Обработка событий в текстовых полях                             | 944         |
| Текстовые области   | 945         |
| Диспетчеры компоновки   | 947         |
| Класс <code>FlowLayout</code>                                   | 948         |
| Класс <code>BorderLayout</code>                                 | 950         |
| Вставки   | 952         |
| Класс <code>GridLayout</code>                                   | 953         |
| Класс <code>CardLayout</code>                                   | 954         |
| Класс <code>GridBagLayout</code>                                | 957         |
| Меню и строки меню  | 962         |
| Диалоговые окна   | 968         |
| Диалоговые окна выбора файлов                                   | 972         |
| О переопределении метода <code>paint()</code>                   | 973         |
| <b>Глава 27. Изображения</b>                                    | <b>975</b>  |
| Форматы файлов  | 975         |
| Основы работы с изображениями: создание, загрузка и отображение | 976         |
| Создание объекта класса <code>Image</code>                      | 976         |
| Загрузка изображения  | 977         |
| Воспроизведение изображения                                     | 977         |
| Интерфейс <code>ImageObserver</code>                            | 979         |
| Двойная буферизация   | 980         |
| Класс <code>MediaTracker</code>                                 | 983         |
| Интерфейс <code>ImageProducer</code>                            | 986         |
| Класс <code>MemoryImageSource</code>                            | 986         |
| Интерфейс <code>ImageConsumer</code>                            | 988         |
| Класс <code>PixelGrabber</code>                                 | 988         |
| Класс <code>ImageFilter</code>                                  | 991         |
| Класс <code>CropImageFilter</code>                              | 991         |
| Фильтр класса <code>RGBImageFilter</code>                       | 992         |
| Дополнительные классы для формирования изображений              | 1004        |
| <b>Глава 28. Утилиты параллелизма</b>                           | <b>1005</b> |
| Пакеты параллельного API  | 1006        |
| Пакет <code>java.util.concurrent.atomic</code>                  | 1008        |
| Пакет <code>java.util.concurrent.locks</code>                   | 1008        |
| Применение объектов синхронизации                               | 1008        |



## 22 Содержание

|   |             |
|---|-------------|
| Класс Semaphore   | 1008        |
| Класс CountdownLatch  | 1014        |
| Класс CyclicBarrier   | 1016        |
| Класс Exchanger   | 1018        |
| Класс Phaser  | 1020        |
| Применение исполнителя  | 1028        |
| Простой пример исполнителя  | 1029        |
| Применение интерфейсов Callable и Future  | 1031        |
| Перечисление TimeUnit   | 1034        |
| Параллельные коллекции  | 1035        |
| Блокировки  | 1036        |
| Атомарные операции  | 1039        |
| Параллельное программирование средствами Fork/Join Framework                      | 1040        |
| Основные классы Fork/Join Framework   | 1041        |
| Стратегия “разделяй и властвуй”   | 1045        |
| Первый простой пример вилочного соединения  | 1047        |
| Влияние уровня параллелизма   | 1049        |
| Пример применения класса RecursiveTask<V>   | 1053        |
| Асинхронное выполнение задач  | 1055        |
| Отмена задачи   | 1056        |
| Определение состояния завершения задачи   | 1056        |
| Перезапуск задачи   | 1056        |
| Предмет дальнейшего изучения  | 1057        |
| Рекомендации относительно вилочного соединения                                    | 1059        |
| Утилиты параллелизма в сравнении с традиционным подходом к многозадачности в Java | 1059        |
| <b>Глава 29. Поточковый API</b>   | <b>1061</b> |
| Основные положения о потоках данных   | 1061        |
| Потоковые интерфейсы  | 1062        |
| Получение потока данных   | 1065        |
| Простой пример потока данных  | 1066        |
| Операции сведения   | 1070        |
| Параллельные потоки данных  | 1073        |
| Отображение   | 1075        |
| Накопление  | 1079        |
| Итераторы и потоки данных   | 1083        |
| Применение итератора в потоке данных  | 1083        |
| Применение итератора-разделителя  | 1085        |
| Дальнейшее изучение потокового API  | 1087        |
| <b>Глава 30. Регулярные выражения и другие пакеты</b>                             | <b>1089</b> |
| Пакеты из базового API  | 1089        |
| Обработка регулярных выражений  | 1092        |
| Класс Pattern   | 1092        |
| Класс Matcher   | 1092        |
| Синтаксис регулярных выражений  | 1093        |
| Примеры, демонстрирующие совпадение с шаблоном                                    | 1094        |
| Два варианта сопоставления с шаблоном   | 1100        |
| Дальнейшее изучение регулярных выражений  | 1100        |
| Рефлексия   | 1101        |

|  |      |
|--|------|
| Удаленный вызов методов  | 1105 |
| Простое приложение “клиент–сервер”, использующее механизм RMI          | 1105 |
| Форматирование даты и времени средствами пакета <code>java.text</code> | 1109 |
| Класс <code>DateFormat</code>  | 1109 |
| Класс <code>SimpleDateFormat</code>                                    | 1111 |
| API даты и времени, внедренный в версии JDK 8                          | 1113 |
| Основные классы даты и времени   | 1113 |
| Форматирование даты и времени  | 1115 |
| Синтаксический анализ символьных строк даты и времени                  | 1117 |
| Дальнейшее изучение пакета <code>java.time</code>                      | 1118 |
| <b>Часть III. Введение в программирование ГПИ средствами Swing</b>     | 1119 |
| <b>Глава 31. Введение в библиотеку Swing</b>                           | 1121 |
| Происхождение библиотеки Swing   | 1121 |
| Построение библиотеки Swing на основе библиотеки AWT                   | 1122 |
| Главные особенности библиотеки Swing                                   | 1122 |
| Легковесные компоненты Swing   | 1123 |
| Подключаемый стиль оформления  | 1123 |
| Связь с архитектурой MVC   | 1124 |
| Компоненты и контейнеры  | 1125 |
| Компоненты   | 1125 |
| Контейнеры   | 1126 |
| Панели контейнеров верхнего уровня                                     | 1126 |
| Пакеты библиотеки Swing  | 1127 |
| Простое Swing-приложение   | 1127 |
| Обработка событий  | 1132 |
| Создание Swing-апплета   | 1135 |
| Рисование средствами Swing   | 1138 |
| Основы рисования   | 1138 |
| Вычисление области рисования   | 1139 |
| Пример рисования   | 1140 |
| <b>Глава 32. Исследование библиотеки Swing</b>                         | 1143 |
| Классы <code>JLabel</code> и <code>ImageIcon</code>                    | 1143 |
| Класс <code>JTextField</code>  | 1145 |
| Кнопки из библиотеки Swing   | 1147 |
| Класс <code>JButton</code>   | 1148 |
| Класс <code>JToggleButton</code>                                       | 1150 |
| Флажки   | 1153 |
| Кнопки-переключатели   | 1155 |
| Класс <code>JTabbedPane</code>   | 1157 |
| Класс <code>JScrollPane</code>   | 1160 |
| Класс <code>JList</code>   | 1162 |
| Класс <code>JComboBox</code>   | 1166 |
| Деревья  | 1168 |
| Класс <code>JTable</code>  | 1172 |
| Основные положения о меню  | 1175 |

## 24 Содержание

|  |      |
|--|------|
| <b>Глава 33. Введение в меню Swing</b>                                     | 1175 |
| Краткий обзор классов JMenuBar, JMenu и JMenuItem                          | 1177 |
| Класс JMenuBar   | 1177 |
| Класс JMenu  | 1178 |
| Класс JMenuItem  | 1180 |
| Создание главного меню   | 1180 |
| Ввод мнемоники и оперативных клавиш в меню                                 | 1184 |
| Ввод изображений и всплывающих подсказок в пункты меню                     | 1187 |
| Классы JRadioButtonMenuItem и JCheckBoxMenuItem                            | 1188 |
| Создание всплывающего меню   | 1190 |
| Создание панели инструментов   | 1193 |
| Действия   | 1196 |
| Составление окончательного варианта программы MenuDemo                     | 1202 |
| Дальнейшее изучение библиотеки Swing                                       | 1208 |
| <br>   |      |
| <b>Часть IV. Введение в программирование ГПИ средствами JavaFX</b>         | 1209 |
| <br>   |      |
| <b>Глава 34. Введение в JavaFX</b>   | 1211 |
| Основные понятия JavaFX  | 1212 |
| Пакеты JavaFX  | 1212 |
| Классы подмостков и сцены  | 1213 |
| Узлы и графы сцены   | 1213 |
| Компоновки   | 1213 |
| Класс приложения и методы его жизненного цикла                             | 1214 |
| Запуск JavaFX-приложения   | 1214 |
| Скелет JavaFX-приложения   | 1215 |
| Компиляция и выполнение JavaFX-приложения                                  | 1218 |
| Поток исполнения приложения  | 1219 |
| Метка – простейший элемент управления в JavaFX                             | 1219 |
| Применение кнопок и событий  | 1221 |
| Основа обработки событий в JavaFX  | 1222 |
| Элемент управления экранной кнопкой  | 1223 |
| Демонстрация обработки событий на примере экранных кнопок                  | 1224 |
| Рисование непосредственно на холсте  | 1227 |
| <br>   |      |
| <b>Глава 35. Элементы управления JavaFX</b>                                | 1233 |
| Классы Image и ImageView   | 1233 |
| Ввод изображения в метку   | 1236 |
| Применение изображения в экранной кнопке                                   | 1238 |
| Класс ToggleButton   | 1240 |
| Класс RadioButton  | 1243 |
| Обработка событий изменения в группе кнопок-переключателей                 | 1246 |
| Другой способ управления кнопками-переключателями                          | 1248 |
| Класс CheckBox   | 1251 |
| Класс ListView   | 1254 |
| Представление списка с полосами прокрутки                                  | 1258 |
| Активизация режима одновременного выбора<br>нескольких элементов из списка | 1258 |
| Класс ComboBox   | 1260 |
| Класс TextField  | 1263 |
| Класс ScrollPane   | 1265 |
| Класс TreeView   | 1269 |

|   |      |
|---|------|
| Эффекты и преобразования                                  | 1273 |
| Эффекты   | 1274 |
| Преобразования  | 1275 |
| Демонстрация эффектов и преобразований                    | 1276 |
| Ввод всплывающих подсказок                                | 1279 |
| Отключение элементов управления                           | 1280 |
| <b>Глава 36. Введение в меню JavaFX</b>                   | 1281 |
| Основные положения о меню                                 | 1281 |
| Краткий обзор классов MenuBar, Menu и MenuItem            | 1283 |
| Класс MenuBar   | 1283 |
| Класс Menu  | 1284 |
| Класс MenuItem  | 1285 |
| Создание главного меню                                    | 1286 |
| Ввод мнемоники и оперативных клавиш в меню                | 1291 |
| Ввод изображений в пункты меню                            | 1293 |
| Классы RadioMenuItem и CheckMenuItem                      | 1294 |
| Создание контекстного меню                                | 1296 |
| Создание панели инструментов                              | 1300 |
| Составление окончательного варианта приложения MenuDemo   | 1302 |
| Дальнейшее изучение JavaFX                                | 1308 |
| <b>Часть V. Применение Java</b>                           | 1309 |
| <b>Глава 37. Компоненты Java Beans</b>                    | 1311 |
| Общее представление о компонентах Java Beans              | 1311 |
| Преимущества компонентов Java Beans                       | 1312 |
| Самоанализ  | 1312 |
| Шаблоны проектирования для свойств компонентов Java Beans | 1313 |
| Шаблоны проектирования для событий                        | 1314 |
| Методы и шаблоны проектирования                           | 1315 |
| Применение интерфейса BeanInfo                            | 1315 |
| Привязанные и ограниченные свойства                       | 1315 |
| Сохраняемость компонентов Java Beans                      | 1316 |
| Настройщики   | 1316 |
| Прикладной программный интерфейс Java Beans API           | 1317 |
| Класс Introspector  | 1319 |
| КлассPropertyDescriptor                                   | 1319 |
| КлассEventSetDescriptor                                   | 1319 |
| КлассMethodDescriptor                                     | 1319 |
| Пример компонента Java Bean                               | 1320 |
| <b>Глава 38. Введение в сервлеты</b>                      | 1323 |
| Предпосылки для разработки сервлетов                      | 1323 |
| Жизненный цикл сервлета                                   | 1324 |
| Варианты разработки сервлетов                             | 1325 |
| Применение контейнера сервлетов Tomcat                    | 1325 |
| Простой пример сервлета                                   | 1327 |
| Создание и компиляция исходного кода сервлета             | 1327 |
| Запуск контейнера сервлетов Tomcat на выполнение          | 1328 |
| Запуск веб-браузера и запрос сервлета                     | 1328 |

|  |      |
|--|------|
| Прикладной программный интерфейс Servlet API | 1329 |
| Пакет javax.servlet                          | 1329 |
| Интерфейс Servlet                            | 1330 |
| Интерфейс ServletConfig                      | 1330 |
| Интерфейс ServletContext                     | 1331 |
| Интерфейс ServletRequest                     | 1331 |
| Интерфейс ServletResponse                    | 1332 |
| Класс GenericServlet                         | 1333 |
| Класс ServletInputStream                     | 1333 |
| Класс ServletOutputStream                    | 1333 |
| Класс ServletException                       | 1333 |
| Ввод параметров сервлета                     | 1334 |
| Пакет javax.servlet.http                     | 1335 |
| Интерфейс HttpServletRequest                 | 1336 |
| Интерфейс HttpServletResponse                | 1337 |
| Интерфейс HttpSession                        | 1338 |
| Класс Cookie                                 | 1339 |
| Класс HttpServlet                            | 1340 |
| Обработка HTTP-запросов и ответов HTTP       | 1341 |
| Обработка HTTP-запросов типа GET             | 1341 |
| Обработка HTTP-запросов типа POST            | 1342 |
| Применение cookie-файлов                     | 1343 |
| Отслеживание сеансов связи                   | 1346 |

## **Приложение А. Применение документирующих комментариев в Java** 1347

|  |      |
|--|------|
| Дескрипторы утилиты javadoc                    | 1347 |
| Дескриптор \$author                            | 1348 |
| Дескриптор {@code}                             | 1349 |
| Дескриптор @deprecated                         | 1349 |
| Дескриптор {@docRoot}                          | 1349 |
| Дескриптор @exception                          | 1349 |
| Дескриптор {@inheritDoc}                       | 1349 |
| Дескриптор {@link}                             | 1350 |
| Дескриптор {@linkplain}                        | 1350 |
| Дескриптор {@literal}                          | 1350 |
| Дескриптор @param                              | 1350 |
| Дескриптор @return                             | 1350 |
| Дескриптор @see                                | 1350 |
| Дескриптор @serial                             | 1351 |
| Дескриптор @serialData                         | 1351 |
| Дескриптор @serialField                        | 1351 |
| Дескриптор @since                              | 1351 |
| Дескриптор @throws                             | 1352 |
| Дескриптор {@value}                            | 1352 |
| Дескриптор @version                            | 1352 |
| Общая форма документирующих комментариев       | 1352 |
| Результаты, выводимые утилитой javadoc         | 1353 |
| Пример применения документирующих комментариев | 1353 |

## **Предметный указатель** 1355