

*Библиотека профессионала*

# Java™ 2

Том 2. Тонкости  
программирования

ВОСЬМОЕ ИЗДАНИЕ

# Core Java

## Volume II – Advanced Features

Eighth Edition

Cay S. Horstmann

Gary Cornell



PRENTICE  
HALL

Sun Microsystems Press

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Capetown • Sydney • Tokyo • Singapore • Mexico City

*Библиотека профессионала*

# Java™ 2

## Том 2. Тонкости программирования

Восьмое издание

Кей Хорстманн

Гари Корнелл



Москва • Санкт-Петербург • Киев  
2012

ББК 32.973.26-018.2.75

X82

УДК 681.3.07

Издательский дом “Вильямс”

Зав. редакцией *С. Н. Тригуб*

Перевод с английского *Я. П. Волковой, Д. Я. Иваненко*

Под редакцией *Ю. Н. Артеменко*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:

info@williamspublishing.com, http://www.williamspublishing.com

**Хорстманн, Кей С., Корнелл, Гари.**

X82 Java 2. Библиотека профессионала, том 2. Тонкости программирования, 8-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2012. — 992 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1482-8 (рус.)

**ББК 32.973.26-018.2.75**

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

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

Authorized translation from the English language edition published by Prentice Hall, Ptr., Copyright © 2008 Sun Microsystems, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Sun Microsystems, Inc., has intellectual property rights relating to implementations of the technology described in this publication. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents, foreign patents, or pending applications. Sun, Sun Microsystems, the Sun logo, J2ME, Solaris, Java, Javadoc, NetBeans, and all Sun and Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Figures 7–45, 7–46: “Foxkeh” © 2006 Mozilla Japan.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2012

*Научно-популярное издание*

**Кей С. Хорстманн, Гари Корнелл**

## **Java 2. Библиотека профессионала, том 2 Тонкости программирования, 8-е издание**

Верстка *Т. Н. Артеменко*

Художественный редактор *В. Г. Павлютин*

Подписано в печать 03.05.2012. Формат 70×100/16

Гарнитура Times. Печать офсетная

Усл. печ. л. 79,98. Уч.-изд. л. 55,95

Доп. тираж 1000 экз. Заказ № 0000

Первая Академическая типография “Наука”

199034, Санкт-Петербург, 9-я линия, 12/28

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1482-8 (рус.)

ISBN 978-01-3235479-0 (англ.)

© Издательский дом “Вильямс”, 2012

© Sun Microsystems, Inc., 2008

# Оглавление

<b>Предисловие</b>	<b>15</b>
<b>Глава 1. Потоки и файлы</b>	<b>19</b>
<b>Глава 2. Язык XML</b>	<b>103</b>
<b>Глава 3. Работа в сети</b>	<b>181</b>
<b>Глава 4. Работа с базами данных</b>	<b>227</b>
<b>Глава 5. Интернационализация</b>	<b>305</b>
<b>Глава 6. Расширенные средства Swing</b>	<b>355</b>
<b>Глава 7. Расширенные средства AWT</b>	<b>515</b>
<b>Глава 8. Компоненты JavaBeans</b>	<b>669</b>
<b>Глава 9. Средства защиты</b>	<b>737</b>
<b>Глава 10. Работа с удаленными объектами</b>	<b>823</b>
<b>Глава 11. Написание сценариев, компиляция кода и обработка аннотаций</b>	<b>865</b>
<b>Глава 12. Платформенно-ориентированные методы</b>	<b>915</b>
<b>Предметный указатель</b>	<b>971</b>

# Содержание

Об авторах	13
Благодарности	13
<b>Предисловие</b>	15
К читателю	15
Краткий тур по книге	15
Соглашения	17
Коды примеров	18
От издательства	18
<b>Глава 1. Потоки и файлы</b>	19
Потоки	20
Чтение и запись байтов	20
Коллекция потоков	22
Комбинирование потоковых фильтров	26
Текстовые входные и выходные данные	29
Запись текстовых выходных данных	30
Считывание текстовых входных данных	32
Сохранение объектов в текстовом формате	32
Наборы символов	37
Чтение и запись двоичных данных	41
Файлы с произвольным доступом	43
ZIP-архивы	49
Потоки объектов и сериализация	55
Понимание формата файлов, создаваемых при сериализации объектов	61
Изменение механизма сериализации, используемого по умолчанию	66
Сериализация одноэлементных множеств и перечислений, обеспечивающих безопасность типов	68
Управление версиями	69
Использование сериализации для клонирования	71
Управление файлами	73
Новые средства для обработки ввода-вывода	80
Отображаемые в памяти файлы	81
Структура буферных данных	87
Блокирование файлов	89
Регулярные выражения	91
<b>Глава 2. Язык XML</b>	103
Введение в XML	104
Структура XML-документа	106
Разбор XML-документа	109
Проверка XML-документов	120
Определения типов документов	121
XML-схема	128
Практический пример XML-документа	130
Поиск информации с помощью XPath	144

Использование пространств имен	150
Потоковые анализаторы	152
Использование SAX-анализатора	152
Использование StAX-анализатора	157
Генерирование XML-документов	160
Создание XML-документа посредством StAX	164
XSLT-преобразования	171
<b>Глава 3. Работа в сети</b>	<b>181</b>
Подключение к серверу	182
Тайм-ауты сокетов	186
Internet-адреса	187
Реализация серверов	189
Обслуживание множества серверов	191
Одностороннее закрытие	195
Прерываемые сокет	196
Отправка электронной почты	202
Создание соединений с URL	207
URL и URI	207
Извлечение информации с помощью URLConnection	208
Отправка данных формы	218
<b>Глава 4. Работа с базами данных</b>	<b>227</b>
Структура JDBC	228
Типы JDBC-драйверов	229
Типичные примеры использования JDBC	230
Язык SQL	231
Конфигурирование JDBC	236
URL базы данных	237
JAR-файлы драйверов	237
Запуск базы данных	237
Регистрация класса драйвера	238
Соединение с базой данных	239
Выполнение операторов SQL	242
Управление соединениями, командами и результирующими наборами	244
Анализ SQL-исключений	245
Заполнение базы данных	247
Выполнение запроса	251
Предварительно подготовленные команды	252
Чтение и запись больших объектов	259
Синтаксис перехода в SQL	261
Множественные результаты	262
Получение автоматически генерируемых ключей	263
Прокручиваемые и обновляемые результирующие наборы	264
Прокрутка результирующего набора	264
Обновляемые результирующие наборы	266
Наборы строк	270
Кэшируемые наборы строк	270
Метаданные	273

## 8 Содержание

---

Транзакции	282
Точки сохранения	283
Пакетные обновления	283
Расширенные типы данных в языке SQL	285
Управление соединениями в Web- и производственных приложениях	287
Введение в LDAP	288
Настройка LDAP-сервера	289
Доступ к каталогу LDAP	292
<b>Глава 5. Интернационализация</b>	<b>305</b>
Региональные стандарты	306
Форматы чисел	311
Валюты	316
Дата и время	318
Сортировка	325
Избирательность сортировки	326
Декомпозиция	327
Форматирование сообщений	332
Форматы выбора	334
Текстовые файлы и наборы символов	336
Кодирование символов в файлах исходного кода	336
Пакеты ресурсов	337
Местонахождение пакетов ресурсов	337
Файлы свойств	338
Классы пакетов	339
Пример приложения	341
<b>Глава 6. Расширенные средства Swing</b>	<b>355</b>
Списки	356
Компонент JList	356
Модели списков	362
Вставка и удаление значений	367
Отображение значений	368
Таблицы	373
Простая таблица	374
Модели таблиц	377
Работа со строками и столбцами	381
Деревья	407
Простые деревья	407
Перечисление узлов	423
Визуализация узлов	425
Обработка событий деревьев	427
Специальные модели деревьев	434
Текстовые компоненты	442
Изменение отслеживания в текстовых компонентах	442
Поля форматированного ввода	446
Фильтры	448
Отображение HTML-данных посредством панели JEditorPane	469



Получение информации о выполнении задачи	476
Индикатор хода процесса	476
Мониторы задач	479
Отслеживание чтения данных из потока	483
Организаторы компонентов	488
Панели с границей раздела	488
Панели с вкладками	492
Панель рабочего стола и внутренние фреймы	498
Каскадное и мозаичное расположение фреймов	501
Запрет на изменение свойства	504
<b>Глава 7. Расширенные средства AWT</b>	<b>515</b>
Алгоритм рисования	516
Фигуры	518
Классы фигур	520
Области	532
Штрихи	533
Расцветки	541
Преобразование координат	543
Отсечение	547
Прозрачность и композиция	549
Параметры рисования	558
Чтение и запись изображений	564
Получение объектов для чтения и записи графических файлов	564
Чтение и запись файлов, содержащих несколько изображений	566
Управление изображениями	574
Создание растровых изображений	574
Фильтрация изображений	581
Печать	590
Вывод на печать графических изображений	590
Многостраничная печать	599
Предварительный просмотр	601
Службы печати	610
Потоковые службы печати	614
Атрибуты печати	614
Буфер обмена	621
Классы и интерфейсы для передачи данных	622
Передача текста	623
Интерфейс Transferable и типы документов	627
Передача изображений через буфер обмена	629
Передача объектов Java через системный буфер обмена	634
Использование локального буфера обмена для передачи ссылок на объекты	638
Копирование путем перетаскивания	638
Поддержка передачи данных в компонентах Swing	640
Источники перетаскивания	644
Приемники перетаскивания	646
Средства интеграции в платформу	654
Начальные экраны-заставки	654
Запуск настольных приложений	659
Системный лоток	664

<b>Глава 8. Компоненты JavaBeans</b>	669
Зачем нужны компоненты JavaBeans	670
Создание компонентов JavaBeans	672
Применение компонентов JavaBeans для создания приложений	674
Упаковка компонентов JavaBeans в JAR-файлах	675
Создание компонентов JavaBeans в интегрированной среде разработки	676
Соглашения об именовании свойств и событий компонента JavaBeans	682
Типы свойств компонентов JavaBeans	684
Простые свойства	684
Индексированные свойства	685
Связанные свойства	685
Свойства с ограничениями	687
Классы BeanInfo	693
Редакторы свойств	697
Создание редакторов свойств	700
Строковые редакторы свойств	700
Визуальные редакторы свойств	703
Настройщики	707
Создание класса настройщика	709
Постоянные компоненты JavaBeans	715
Использование средств обеспечения постоянства JavaBeans для работы с произвольными данными	719
Использование делегатов при создании объектов	719
Целый пример постоянного компонента JavaBeans	724
<b>Глава 9. Средства защиты</b>	737
Загрузчики классов	738
Иерархия загрузчиков классов	740
Использование загрузчиков классов в качестве пространств имен	742
Создание собственного загрузчика классов	742
Верификация байтового кода	748
Диспетчеры защиты и полномочия	753
Модель защиты Java	754
Файлы политики безопасности	758
Пользовательские права доступа	765
Реализация класса полномочий	766
Аутентификация пользователей	771
Регистрационные модули JAAS	777
Цифровые подписи	786
Дайджест сообщения	787
Подписание сообщений	793
Формат сертификата X.509	795
Проверка подписи	797
Проблема аутентификации	799
Подписание сертификатов	801
Запросы сертификатов	802
Подписание кода	803
Подписание JAR-файлов	804
Сертификаты разработчиков программ	808

Шифрование	810
Симметричные шифры	810
Генерация ключей	812
Потоки шифрования	816
Шифрование с открытым ключом	818
<b>Глава 10. Работа с удаленными объектами</b>	<b>823</b>
Клиент и сервер	824
Удаленные вызовы методов	827
Заглушки и передача параметров	827
Модель программирования RMI	829
Интерфейсы и реализации	829
Служба регистрации RMI	830
Развертывание программы	834
Регистрация активности RMI	837
Параметры и возвращаемые значения в удаленных методах	839
Передача удаленных объектов	839
Передача не удаленных объектов	839
Динамическая загрузка классов	841
Удаленные ссылки с множеством интерфейсов	846
Удаленные объекты и методы equals (), hashCode () и clone ()	847
Активизация удаленных объектов	847
Web-службы и JAX-WS	853
Использование JAX-WS	853
Клиент Web-службы	856
Служба электронного магазина Amazon	859
<b>Глава 11. Написание сценариев, компиляция кода и обработка аннотаций</b>	<b>865</b>
Написание сценариев для платформы Java	866
Получение механизма сценариев	866
Оценка сценариев и привязки	867
Перенаправление входных и выходных данных	869
Вызов сценарных функций и методов	870
Компиляция сценария	872
Пример: создание сценария обработки событий GUI-интерфейса	873
API-интерфейс компилятора	877
Компилирование простым путем	877
Использование заданий на компиляцию	877
Пример: Динамическая генерация кода Java	882
Использование аннотаций	887
Пример: аннотирование обработчиков событий	888
Синтаксис аннотаций	893
Стандартные аннотации	897
Аннотации для компиляции	897
Аннотации для управления ресурсами	898
Мета-аннотации	898
Обработка аннотаций на уровне исходного кода	900
Разработка байт-кода	907
Изменение байт-кодов во время загрузки	912

## 12 Содержание

---

<b>Глава 12. Платформенно-ориентированные методы</b>	915
Вызов функции на языке C из Java-программы	916
Числовые параметры и возвращаемые значения	922
Использование функции <code>printf()</code> для форматирования чисел	922
Строковые параметры	924
Получение доступа к полям объектов	929
Получение доступа к полям экземпляра	929
Получение доступа к статическим полям	933
Кодирование сигнатуры	934
Вызов Java-методов	935
Методы экземпляра	936
Статические методы	937
Конструкторы	938
Альтернативные вызовы методов	938
Доступ к элементам массивов	942
Обработка ошибок	945
API вызова	950
Пример обращения к реестру Windows	955
Общие сведения о реестре Windows	955
Java-интерфейс для доступа к реестру	956
Организация доступа к реестру посредством платформенно-ориентированных методов	957
<b>Предметный указатель</b>	971

## Об авторах

**Кей Хорстманн** (Kay Horstmann) – профессор факультета вычислительной техники в Университете Сан-Хосе, обладатель звания “Чемпион по Java” и частый докладчик на многих отраслевых конференциях. Является соавтором книги *JavaServer Faces. Библиотека профессионала* (ИД “Вильямс”, 2008 г.).

**Гари Корнелл** (Gary Cornell) имеет более чем 20-летний опыт обучения и написания книг по программированию. Является одним из основателей издательства Apress. Он написал множество книг-бестселлеров для профессионалов, попал в финал конкурса Jolt Award, а также стал обладателем звания “Выбор читателей” в журнале *Visual Basic Magazine*.

## Благодарности

Написание книги всегда требует приложения огромных усилий, да и переделка тоже, похоже, не дается намного легче, особенно при такой высокой скорости развития технологии Java. Выход книги в свет подразумевает участие массы людей, и я с большим удовольствием хотел бы поблагодарить всех членов команды, работавших над этой книгой.

Многие сотрудники из издательства Prentice Hall и Sun Microsystems Press оказали ценную помощь и при этом остались незамеченными. Я хочу, чтобы они знали, что я очень благодарен им за их усилия. Как обычно, искренние слова благодарности моему издателю, Грегу Доэнчу (Greg Doench), из Prentice Hall за управление процессом написания и изготовления книги и позволение мне пребывать в благодатном неведении обо всех тех, кто за этим стоит. Благодарю Ванессу Мур (Vanessa Moore) за замечательную поддержку в вопросах производства. Отдельное спасибо моему соавтору по предыдущим изданиям Гари Корнеллу (Gary Cornell), который после этого погрузился уже в другие приключения.

Спасибо многочисленным читателям предыдущих выпусков за предоставление информации о допущенных ошибках и привнесение массы стоящих предложений по улучшению. Особую благодарность хотелось бы выразить команде рецензентов, которые тщательнейшим образом изучили все детали данного манускрипта и избавили меня от массы других глупых ошибок.

В рецензировании этого и предыдущих изданий участие принимали: Чак Эллисон (Chuck Allison, ассистент редактора в журнале *C/C++ Users Journal*), Лэнс Андерсон (Lance Anderson, из Sun Microsystems), Алек Битон (Alec Beaton, из PointBase, Inc.), Клиф Берг (Cliff Berg, из iSavvix Corporation), Джошуа Блок (Joshua Bloch, из Sun Microsystems), Дэвид Браун (David Brown), Корки Картрайт (Corky Cartwright), Фрэнк Коэн (Frank Cohen, из PushToTest), Крис Крэйн (Chris Crane, из devXsolution), доктор Николас де Лилло (Dr. Nicholas J. De Lillo, из колледжа Манхэттена), Ракеш Дхупар (Rakesh Dhoora, из Oracle), Роберт Эванс (Robert Evans, ведущий специалист из лаборатории прикладной физики при университете Джона Хопкинса), Дэвид Гери (David Geary, из Sabreware), Брайан Гоэтс (Brian Goetz, главный консультант из Quietix Corp.), Анжела Гордон (Angela Gordon, из Sun Microsystems), Дэн Гордон (Dan Gordon, из Sun Microsystems), Роб Гордон (Rob Gordon), Джон Грэй (John Gray, из университета Хартфорда), Камерон Грегори (Cameron Gregory, из olabs.com), Марти Холл (Marty Hall, из лаборатории прикладной физики при университете Джона Хопкинса), Винсент Харди (Vincent Hardy, из Sun Microsystems), Дэн Харки (Dan Harkey, из Государственного университета в Сан-Хосе), Уильям Хиггинс (William Higgins, из IBM),

## 14 Благодарности

---

Владимир Иванович (Vladimir Ivanovic, из PointBase), Джерри Джексон (Jerry Jackson, из ChannelPoint Software), Тим Киммет (Tim Kimmel, из Preview Systems), Крис Лаффра (Chris Laffra), Чарли Лаи (Charlie Lai, из Sun Microsystems), Анжелика Лэнгер (Angelika Langer), Дуг Лэнгстон (Doug Langston), Хэнг Лау (Hang Lau, из университета имени Мак-Гилла), Марк Лоуренс (Mark Lawrence), Дуг Ли (Doug Lea, из SUNY Oswego), Грегори Лонгшор (Gregory Longshore), Боб Линч (Bob Lynch, из Lynch Associates), Филип Милн (Philip Milne, консультант), Марк Морриссей (Mark Morrissey, из Выпускного института штата Орегон), Махеш Нилаканта (Mahesh Neelakanta, из Атлантического университета, штат Флорида), Хао Фам (Hao Pham), Пол Филион (Paul Philion), Блейк Регсделл (Blake Ragsdell), Илбер Рамадани (Ylber Ramadani, из университета Райерсона), Стюарт Регес (Stuart Reges из Университета Аризоны), Рич Розен (Rich Rosen, из Interactive Data Corporation), Питер Сандерс (Peter Sanders, из университета ESSI, Ницца, Франция), доктор Пол Сангера (Dr. Paul Sanghera, из Государственного университета в Сан-Хосе и колледжа в Бруксе), Пол Севинк (Paul Sevinc, из Teamup AG), Деванг Ша (Devang Shah, из Sun Microsystems), Ричард Сливчак (Richard Slywczak, из исследовательского центра NASA/Glenn), Брэдли Смит (Bradley A. Smith), Стивен Стелтинг (Steven Stelting, из Sun Microsystems), Кристофер Тэйлор (Christopher Taylor), Люк Тэйлор (Luke Taylor, из Valtech), Джордж Тируватукал (George Thiruvathukal), Ким Топли (Kim Topley, автор книги *Core JFC*), Джанет Трауб (Janet Traub), Пол Тима (Paul Tuma, консультант), Питер Ван дер Линден (Peter van der Linden, из Sun Microsystems) и Берт Уолш (Burt Walsh).

*Кей Хорстманн*  
Сан-Франциско, 2008 г.

# ПРЕДИСЛОВИЕ

## К читателю

Книгу, которую вы сейчас держите в своих руках, является вторым томом восьмого издания книги, которое полностью адаптировано под Java SE 6. В первом томе рассказывалось о базовых функциональных возможностях языка Java; в этом же томе рассматриваются более сложные темы, которые могут понадобиться программисту для профессиональной разработки программного обеспечения. Поэтому, как и в первом томе и предыдущих изданиях настоящей книги, здесь по-прежнему расчет делается на тех программистов, которые планируют применять технологию Java для работы над реальными проектами.

Просим обратить внимание, что опытным разработчикам, умеющим работать с такими сложными средствами языка, как внутренние и обобщенные классы, читать первый том для понимания содержимого второго тома, вовсе не обязательно. Несмотря на то что в этом томе, конечно же, где необходимо, встречаются ссылки на разделы предыдущего тома (который, мы надеемся, вы все-таки приобретете или уже приобрели), весь необходимый базовый материал можно легко найти в любой более или менее полной вводной книге по платформе Java.

## Краткий тур по книге

Главы в этой книге по большей части не зависят друг от друга, что дает читателю возможность начинать изучение материала с той темы, которая интересует его больше всего, и вообще читать все главы в любом удобном ему порядке.

- **Глава 1** посвящена обработке входных и выходных данных. В Java весь ввод и вывод обрабатывается посредством так называемых *потоков* (streams). Потоки позволяют однотипным образом иметь дело с передачей данных между различными источниками данных, такими как файлы, сетевые соединения и блоки памяти. Эта глава сначала включает детальное описание классов чтения (Reader) и записи (Writer), которые облегчают работу с Unicode. Далее в ней демонстрируются процессы, происходящие внутри в случае применения механизма сериализации объектов, который делает сохранение и загрузку объектов простой и удобной. И, наконец, в заключительной части этой главы рассказывается о классах “новых средств для обработки ввода-вывода” (являвшихся новыми на тот момент, когда они добавлялись в Java SE 1.4), которые поддерживают эффективные файловые операции, а также о библиотеке регулярных выражений.
- **Глава 2** посвящена XML. Здесь показывается, как выполнять синтаксический анализ файлов XML, как генерировать XML и как использовать трансформации XSL. В качестве примера описывается процесс определения компоновки формы Swing из XML. Новым дополнением в этой главе является рассмотрение API-интерфейса XPath, который значительно упрощает поиск мелких деталей в больших объемах XML.
- **Глава 3** посвящена сетевому API-интерфейсу. Язык Java позволяет феноменально легко выполнять сложные операции по программированию сетевых задач. В этой главе демонстрируется, как создавать сетевые соединения с серверами, как реализовывать свои собственные серверы и как открывать HTTP-подключения.

- **Глава 4** посвящена программированию баз данных. Основное внимание здесь уделяется интерфейсу JDBC (Java Database Connectivity API – API-интерфейс для организации доступа к базам данных из Java-приложений), который позволяет Java-программам подключаться к реляционным базам данных. Помимо этого здесь также показывается, как писать полезные программы для обработки реальных рутинных операций баз данных с применением только базового поднабора средств API-интерфейса JDBC. (Для рассмотрения всех средств API-интерфейса JDBC потребовалась бы книга почти такого же размера, как эта.) В завершение главы приводится краткая информация об иерархических базах данных, а также об интерфейсе JNDI (Java Naming and Directory Interface – интерфейс именования и каталогов Java) и протоколе LDAP (Lightweight Directory Access Protocol – облегченный протокол доступа к каталогам).
- **Глава 5** посвящена такой технологии, как интернационализация, важность которой, по нашему мнению, будет только возрастать. Язык программирования Java является одним из нескольких языков, которые с самого начала предусматривали возможность обработки Unicode, но поддержка интернационализации в Java позволяет идти гораздо дальше. В частности, путем интернационализации приложений Java разработчик может делать так, чтобы они пересекали границы не только платформы, но и страны. Например, в этой главе демонстрируется, как можно написать апплет для вычисления времени выхода на пенсию, применяющий английский, немецкий или китайский языки, в зависимости от выбранных в браузере региональных и языковых настроек.
- **В главе 6** содержится весь материал по Swing, который не вошел в первый том, в частности, описание важных и сложных компонентов деревьев и таблиц. Здесь демонстрируются основные способы применения панелей редактора, Java-реализация многодокументного интерфейса, индикаторы хода выполнения, которые используются в многопоточных программах, и “средства интеграции для рабочего стола” вроде заставок и поддержки для области пиктограмм панели задач. Опять-таки, основное внимание уделяется только наиболее полезным конструкциям, с которыми разработчикам наверняка доведется встречаться на практике, поскольку даже краткое рассмотрение всей библиотеки Swing заняло бы целых несколько томов и представляло бы интерес только для очень узких специалистов.
- **Глава 7** посвящена API-интерфейсу Java 2D, которым можно пользоваться для создания реалистичных изображений и специальных эффектов. В ней также рассказывается и о некоторых более совершенных средствах AWT (Abstract Windowing Toolkit – набор инструментальных средств для абстрактных окон), которые оказались слишком специфическими для рассмотрения в первом томе, но которые, тем не менее, должны присутствовать в наборе инструментов каждого программиста. К их числу относятся средства печати и API-интерфейсы для обеспечения возможности выполнения операций вырезания, вставки и перетаскивания.
- **В главе 8** рассказывается обо всем, что разработчикам необходимо знать о доступном для платформы Java API-интерфейсе программирования компонентов под названием JavaBeans. Здесь показано, как создавать свои собственные bean-компоненты так, чтобы другие программисты могли манипулировать ими в интегрированных средах разработки. Завершается глава описанием способов применения предлагаемого в Java механизма постоянства для сохранения данных в таком формате, который, в отличие от получаемого при сериализации объектов, подходит для длительного хранения.



- **Глава 9** посвящена модели безопасности Java. Платформа Java с самого начала разрабатывалась с учетом безопасности, и в этой главе объясняется, что именно позволяет ей эту безопасность обеспечивать. Первым делом здесь демонстрируется, как создавать свои собственные загрузчики классов и диспетчеры защиты для специальных приложений. Далее рассказывается об API-интерфейсе безопасности, который позволяет оснащать приложения важными средствами, вроде механизма создания цифровых подписей для сообщений и кода, механизма авторизации и аутентификации и механизма шифрования. Завершается глава демонстрацией примеров с применением таких алгоритмов шифрования, как AES и RSA.
- **Глава 10** посвящена распределенным объектам. В частности, здесь подробно рассматривается протокол RMI (Remote Method Invocation – удаленный вызов методов). Этот протокол, по сути, представляет собой API-интерфейс и позволяет работать с Java-объектами, которые разбросаны по множеству компьютеров. Кроме того, здесь еще также вкратце рассказывается о Web-службах и приводится пример, в котором программа Java взаимодействует с Web-службой Amazon.
- В **главе 11** описываются три различных технологии для обработки кода. API-интерфейсы сценариев и компилятора, представленные в версии Java SE 6, позволяют программе вызывать код, написанный на каком-то языке сценариев, например, JavaScript или Groovy, и компилировать код на языке Java. Аннотации позволяют добавлять в Java-программу произвольно выбираемую информацию (еще иногда также называемую метаданными). В этой главе показывается, как обработчики аннотаций могут собирать аннотации на уровне источника и на уровне файлов классов, и как аннотации могут применяться для оказания влияния на поведение классов во время выполнения. Аннотации выгодно использовать только вместе с соответствующими инструментальными средствами, и мы надеемся, что наш материал поможет читателю научиться выбирать именно те средства обработки аннотаций, которые наиболее отвечают его потребностям.
- **Глава 12** посвящена платформенно-ориентированным методам, которые позволяют вызывать методы, предназначенные для конкретной машины, например, для Microsoft Windows API. Очевидно, что данное средство является спорным, ведь использование платформенно-ориентированных методов ведет к исчезновению межплатформенных преимуществ платформы Java. Тем не менее, каждый серьезный программист, занимающийся написанием Java-приложений для конкретных платформ, должен обязательно владеть им. Время от времени все-таки встречаются ситуации, когда требуется переключаться на API-интерфейс операционной системы целевой платформы при взаимодействии с устройством или службой, которая не поддерживается платформой Java. То, как удобно делается, как раз и иллюстрируется в данной главе на примере осуществления из Java-программы доступа к API-интерфейсу системного реестра Windows.

Как обычно, все главы были полностью переделаны под последнюю версию Java. Весь устаревший материал был изъят, а новые API-интерфейсы, появившиеся только в версии Java SE 6, рассматриваются в мельчайших деталях.

## Соглашения

Как это принято во многих компьютерных книгах, моноширинный шрифт используется для представления исходного кода.



Такой пиктограммой выделяются замечания.

---



Такой пиктограммой выделяются советы.

---



Такой пиктограммой выделяются предупреждения об опасности.

---



В нашей книге есть множество замечаний, касающихся C++, в которых объясняются отличия между языками Java и C++. Вы можете пропустить их, если не имеете опыта работы с C++ или если склонны воспринимать этот опыт как страшный сон, который лучше забыть.

---



### Интерфейс прикладного программирования

Язык Java сопровождается огромной библиотекой интерфейса прикладного программирования (API). Используя вызов API впервые, в конце соответствующего раздела мы добавляем краткое описание, сопровождаемое такой пиктограммой. Эти описания не слишком информативны, но, как мы надеемся, более информативны, чем те, что представлены в официальной онлайн-документации по API. Мы снабжаем каждую врезку по API номером версии, в которой средство впервые было представлено, чтобы помочь тем читателям, которые не используют самую последнюю версию Java.

Программы с доступным исходным кодом организованы в виде примеров, как показано ниже.

#### Листинг 11.1. Содержимое файла `ScriptTest.java`

---

## Коды примеров

Коды примеров, рассмотренных в книге, доступны как на Web-сайте, посвященном книге (<http://horstmann.com/corejava>), так и на Web-сайте издательства.

## От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши координаты:

E-mail: [info@williamspublishing.com](mailto:info@williamspublishing.com)

WWW: <http://www.williamspublishing.com>

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152

---

*Глава*

1

# ПОТОКИ И ФАЙЛЫ

- ▶ Потоки
- ▶ Текстовые входные и выходные данные
- ▶ Чтение и запись двоичных данных
- ▶ ZIP-архивы
- ▶ Потоки объектов и сериализация
- ▶ Управление файлами
- ▶ Новые средства для обработки ввода-вывода
- ▶ Регулярные выражения

**В** этой главе речь пойдет об интерфейсах прикладного программирования (Application Programming Interface – API), которые доступны в Java для организации ввода и вывода данных. В частности, здесь читатель сможет узнать о том, как получать доступ к файлам и каталогам, а также как считывать и записывать данные в двоичном и текстовом формате. Кроме того, в этой главе еще также будет рассмотрен механизм сериализации, позволяющий сохранять объекты так же легко, как и текстовые или числовые данные. Далее будут перечислены несколько улучшений, которые были добавлены в пакет “новых средств для обработки ввода-вывода” (new I/O – NIO) с соответствующим названием `java.nio`, представленный впервые еще в версии Java SE 1.4. В завершение этой главы мы поговорим о регулярных выражениях, хотя они и не имеют непосредственного отношения к потокам и файлам. Нам не удалось найти лучшего места для обсуждения этой темы, как его, очевидно, не удалось найти и команде разработчиков Java, присоединивших спецификацию API регулярных выражений к запросу спецификации по новым средствам для обработки ввода-вывода в Java SE 1.4.

## Потоки

В API-интерфейсе Java объект, из которого можно считывать последовательность байтов, называется *потоком входных данных* (input stream), а объект, в который можно записывать последовательность байтов – *потоком выходных данных* (output stream). В роли таких источников и пунктов назначения для последовательностей байтов чаще всего выступают файлы, но также могут выступать и сетевые подключения или даже блоки памяти. Абстрактные классы `InputStream` и `OutputStream` образуют основу иерархии классов ввода-вывода (input/output – I/O).

Из-за того, что байтовые (byte-oriented) потоки являются неудобными для обработки информации, хранящейся в кодировке Unicode (напоминаем, что в кодировке Unicode на каждый символ приходится несколько байтов), для обработки Unicode-символов предусмотрена отдельная иерархия классов, все из которых наследуются от абстрактных классов `Reader` и `Writer`. Эти классы позволяют выполнять операции чтения и записи на базе не однобайтовых символов, а двухбайтовых единиц кода Unicode.

## Чтение и запись байтов

Класс `InputStream` имеет следующий абстрактный метод:

```
abstract int read()
```

Этот метод считывает один байт и возвращает либо считанный байт, либо `-1`, если наталкивается на конец потока входных данных. Разработчик конкретного класса потока входных данных может переопределять этот метод так, чтобы он предоставлял какую-то полезную функциональную возможность. Например, в классе `FileInputStream` этот метод выполняет считывание одного байта из файла. `System.in` представляет собой предопределенный объект подкласса `InputStream`, который позволяет считывать информацию с клавиатуры.

У класса `InputStream` еще также имеются и неабстрактные методы для считывания массива байтов или для пропуска ряда байтов. Эти методы вызывают абстрактный метод `read`, благодаря чему подклассам нужно переопределять только один метод.

Аналогичным образом, класс `OutputStream` определяет следующий абстрактный метод, который записывает один байт в место выходных данных:

```
abstract void write(int b)
```