
Предисловие

К читателю

Книга, которую вы держите в руках, является вторым томом девятого издания, полностью обновленного по версии Java SE 7. В первом томе рассматривались основные языковые средства Java, а в этом томе речь пойдет о расширенных функциональных возможностях, которые могут понадобиться программисту для разработки программного обеспечения на высоком профессиональном уровне. Поэтому этот том, как, впрочем, и первый том настоящего и предыдущих изданий данной книги, нацелен на тех программистов, которые собираются применять технологию Java в работе над реальными проектами.

Следует, однако, иметь в виду, что опытным разработчикам, умеющим свободно обращаться с такими сложными языковыми средствами, как внутренние и обобщенные классы, читать первый том для понимания материала второго тома совсем не обязательно. И хотя в этом томе делаются ссылки (там, где требуется) на главы первого тома, который, мы надеемся, вы все-таки приобретете или уже приобрели, весь необходимый материал по основам Java нетрудно найти в любой начальной литературе по платформе Java.

Краткий обзор книги

В целом главы этого тома составлены независимо друг от друга. Это дает читателю возможность начинать изучение материала с той темы, которая интересует его больше всего, и вообще читать главы второго тома книги в любом удобном ему порядке.

Глава 1 посвящена организации ввода-вывода. В Java весь ввод-вывод осуществляется через так называемые *потоки ввода-вывода*. Такие потоки позволяют единообразно обмениваться данными между различными источниками, включая файлы, сетевые соединения и блоки памяти. В начале этой главы приводится подробное описание классов чтения и записи в потоки ввода-вывода, упрощающие обработку данных в уникоде. Далее в ней рассматривается внутренний механизм сериализации объектов, который делает простым и удобным сохранение и загрузку объектов. И в завершение главы обсуждаются регулярные выражения и библиотека NIO2 в версии Java SE 7, благодаря которой становятся очень удобными такие типичные операции ввода-вывода, как, например, чтение всех строк из файла.

Основной темой **главы 2** является XML. В ней показывается, каким образом осуществляется синтаксический анализ XML-файлов, генерируется разметка в коде XML и выполняются XSL-преобразования. В качестве примера демонстрируется порядок обозначения компоновки Swing-формы в формате XML. В этой главе рассматривается

также прикладной интерфейс API XPath, в значительной степени упрощающий поиск мелких подробностей в больших объемах данных XML.

В **главе 3** рассматривается сетевой прикладной интерфейс API. В Java чрезвычайно просто решаются сложные задачи сетевого программирования. В этой главе показывается, как устанавливаются сетевые соединения с серверами, реализуются собственные серверы и организуется связь по протоколу HTTP.

Глава 4 посвящена программированию баз данных. Основное внимание в ней уделяется интерфейсу JDBC (прикладному интерфейсу API для организации доступа к базам данных из приложений на Java), который позволяет прикладным программам на Java устанавливать связь с реляционными базами данных. В этой главе также показывается, как писать полезные программы для выполнения рутинных операций с настоящими базами данных, применяя только базовое подмножество интерфейса JDBC. (Для рассмотрения всех средств интерфейса JDBC потребовалась бы отдельная книга почти такого же объема, как и эта.) И в завершение главы приводятся краткие сведения об интерфейсе JNDI (Java Naming and Directory Interface — интерфейс именования и каталогов Java).

В **главе 5** обсуждаются вопросы интернационализации, важность которой, на наш взгляд, будет со временем только возрастать. Java относится к тем немногочисленным языкам программирования, где с самого начала предусматривалась возможность обработки данных в уникоде (Unicode), но поддержка интернационализации в Java этим не ограничивается. В частности, интернационализация прикладных программ на Java позволяет сделать их независимыми не только от платформы, но и от страны применения. В качестве примера в этой главе демонстрируется, как написать апплет для расчета времени выхода на пенсию с выбором английского, немецкого или китайского языка в зависимости от региональных настроек в браузере.

В **главе 6** представлен весь материал по библиотеке Swing, который не вошел в первый том данной книги, в том числе описание важных и сложных компонентов деревьев и таблиц. В ней демонстрируются основные способы применения панелей редактора, реализация в Java многодокументного интерфейса, индикаторы выполнения, применяемые в многопоточных программах, а также средства интеграции для рабочего стола вроде заставок и поддержки области уведомлений на панели задач. Опять же основное внимание уделяется только наиболее полезным конструкциям, которые разработчикам чаще всего приходится использовать на практике, поскольку даже краткое рассмотрение всей библиотеки Swing заняло бы не один том и представляло бы интерес только для очень узких специалистов.

В **главе 7** рассматривается прикладной интерфейс Java 2D API, которым можно пользоваться для рисования реалистичных графических изображений и спецэффектов. В ней также рассказывается о некоторых более развитых средствах библиотеки AWT (Abstract Windowing Toolkit — набор инструментальных средств для абстрактных окон), которые оказались слишком специфическими для рассмотрения в первом томе и, тем не менее, должны быть включены в арсенал средств всякого программирующего на Java. К их числу относятся средства вывода на печать и прикладные интерфейсы API, позволяющие выполнять операции вырезания, вставки и перетаскивания объектов.

В **главе 8** рассказывается обо всем, что требуется знать разработчикам о доступном на платформе Java прикладном интерфейсе API для программирования компонентов JavaBeans. В этой главе показывается, как создавать свои собственные компоненты JavaBeans, чтобы другие программисты могли манипулировать ими в интегрированных средах разработки (ИСР). И в завершение главы поясняется, как пользоваться

механизмом сохраняемости компонентов JavaBeans для длительного хранения данных в пригодном для этого формате, в отличие от сериализации объектов.

В **главе 9** представлена модель безопасности Java. Платформа Java с самого начала разрабатывалась с учетом безопасности, и в этой главе объясняется, что именно позволяет ей обеспечивать безопасность. Сначала в ней демонстрируется, как создавать свои собственные загрузчики классов и диспетчеры защиты для специальных приложений. Затем рассматривается прикладной интерфейс API для безопасности, который позволяет оснащать приложения важными средствами вроде механизма цифровых подписей сообщений и кода, а также авторизации, аутентификации и шифрования. И завершается эта глава демонстрацией примеров, в которых применяются такие алгоритмы шифрования, как AES и RSA.

Глава 10 посвящена распределенным объектам. В ней, в частности, подробно рассматривается протокол RMI (Remote Method Invocation — вызов удаленных методов). Этот протокол, по существу, представляет собой прикладной интерфейс API и позволяет обращаться с объектами Java, которые распределены среди многих компьютеров.

В **главе 11** описываются три разные технологии для обработки кода. Так, прикладные интерфейсы API для сценариев и компилятора дают возможность вызывать в программе на Java код, написанный на каком-нибудь языке создания сценариев, например JavaScript или Groovy, и компилировать его в код Java. Аннотации позволяют вводить в программу на Java произвольно выбираемую информацию (иногда еще также называемую *метаданными*). В этой главе показывается, каким образом обработчики аннотаций собирают аннотации на уровне источника и на уровне файлов классов и как с помощью аннотаций оказывается воздействие на поведение классов во время выполнения. Аннотации выгодно использовать только вместе с подходящими инструментальными средствами, и мы надеемся, что материал этой главы поможет читателю научиться выбирать именно те средства обработки аннотаций, которые в наибольшей степени отвечают его потребностям.

Глава 12 посвящена собственным методам, которые позволяют вызывать функции, специально написанные для конкретной платформы, например Microsoft Windows. Очевидно, что данное языковое средство является спорным, ведь применение собственных методов сводит на нет все межплатформенные преимущества Java. Тем не менее всякий, серьезно программирующий на Java приложения для конкретных платформ, должен уметь обращаться с собственными методами. Ведь иногда возникают ситуации, когда требуется обращаться к прикладному интерфейсу API операционной системы целевой платформы для взаимодействия с устройствами или службами, которые не поддерживаются на платформе Java. В этой главе показано, как это сделать, на примере организации доступа из программы на Java к прикладному интерфейсу API системного реестра Windows.

Как обычно, все главы второго тома были полностью обновлены по самой последней версии Java. Весь устаревший материал был изъят, а новые прикладные интерфейсы API, появившиеся в версии Java SE 7, подробно рассматриваются в соответствующих местах.

Условные обозначения

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



НА ЗАМЕТКУ! Этой пиктограммой выделяются примечания.



СОВЕТ. Этой пиктограммой выделяются советы.



ВНИМАНИЕ! Этой пиктограммой выделяются предупреждения о потенциальной опасности.



НА ЗАМЕТКУ C++! В этой книге имеется немало примечаний к синтаксису C++, где разъясняются отличия между языками Java и C++. Вы можете пропустить их, если у вас нет опыта программирования на C++ или же если вы склонны воспринимать этот опыт как страшный сон, который лучше забыть.

Язык Java сопровождается огромной библиотекой в виде прикладного программного интерфейса (API). При упоминании вызова какого-нибудь метода из прикладного интерфейса API в первый раз в конце соответствующего раздела приводится его краткое описание. Эти описания не слишком информативны, но, как мы надеемся, более содержательны, чем те, что представлены в официальной оперативно доступной документации на прикладной интерфейс API. Имена интерфейсов выделены *полужирным*, как это делается в официальной документации. А число после имени класса, интерфейса или метода обозначает версию JDK, в которой данное средство было внедрено, как показано ниже.

Название прикладного программного интерфейса 1.2

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

Листинг 1.1. Исходный код из файла `ScriptTest.java`

Примеры исходного кода

Все примеры исходного кода, приведенные в этом томе в частности и в данной книге вообще, доступны в архивированном виде на посвященном ей веб-сайте по адресу <http://horstmann.com/corejava>.

Благодарности

Написание книги всегда требует значительных усилий, а ее переписывание не намного легче, особенно если учесть постоянные изменения в технологии Java. Чтобы сделать книгу полезной, необходимы совместные усилия многих преданных людей, и мы с удовольствием выражаем признательность всем, кто внес свой посильный вклад в наше общее дело.

Большое число сотрудников издательств Prentice Hall оказали неоценимую помощь, хотя и остались в тени. Я хотел бы выразить им свою признательность за их усилия. Как всегда, самой горячей благодарности заслуживает мой редактор из издательства Prentice Hall Грег Доенч (Greg Doench) — за сопровождение книги на протяжении всего процесса ее написания и издания, а также за то, что он позволил мне пребывать в блаженном неведении относительно многих скрытых деталей этого процесса. Я благодарен Джули Нахил (Julie Nahil) за оказанную помощь в подготовке книги к изданию, а также Дмитрию и Алине Кирсановым — за литературное редактирование и набор рукописи книги. Приношу также свою благодарность моему соавтору по прежним изданиям Гари Корнеллу (Gary Cornell), который с тех пор обратился к другим занятиям.

Выражаю большую признательность многим читателям прежних изданий, которые сообщали о найденных ошибках и внесли массу ценных предложений по улучшению книги. Я особенно благодарен блестящему коллективу рецензентов, которые тщательно просмотрели рукопись книги, устранив в ней немало досадных ошибок.

Среди рецензентов этого и предыдущих изданий хотелось бы отметить Чака Аллисона (Chuck Allison, выпускающего редактора издания *C/C++ Users Journal*), Ланса Андерсона (Lance Anderson) из компании Oracle, Алека Битона (Alec Beaton) из PointBase, Inc., Клиффа Берга (Cliff Berg), Джошуа Блоха (Joshua Bloch), Дэвида Брауна (David Brown), Корки Картрайта (Corky Cartwright), Френка Коена (Frank Cohen) из PushToTest, Криса Крейна (Chris Crane) из devXsolution, доктора Николаса Дж. Де Лилло (Dr. Nicholas J. De Lillo) из Манхэттенского колледжа, Ракеша Дхупара (Rakesh Dhoopar) из компании Oracle, Дэвида Джири (David Geary) из Sabreware, Джима Гиша (Jim Gish) из Oracle, Брайана Гоеца (Brian Goetz) из Oracle, Анджелу Гордон (Angela Gordon), Дэна Гордона (Dan Gordon), Роба Гордона (Rob Gordon), Джона Грэя (John Gray) из Хартфордского университета, Камерона Грегори (Cameron Gregory, olabs.com), Марти Холла (Marty Hall) из лаборатории прикладной физики в университете имени Джона Хопкинса, Винсента Харди (Vincent Hardy) из Adobe Systems, Дэна Харки (Dan Harkey) из университета штата Калифорния в Сан-Хосе, Вильяма Хиггинса (William Higgins) из IBM, Владимира Ивановича (Vladimir Ivanovic) из PointBase, Джерри Джексона (Jerry Jackson) из CA Technologies, Тима Киммета (Tim Kimmet) из Walmart, Криса Лаффра (Chris Laffra), Чарли Лаи (Charlie Lai) из компании Apple, Анжелику Лангер (Angelika Langer), Дуга Лэнгстона (Doug Langston), Ханг Лау (Hang Lau) из университета имени Макгилла, Марка Лоуренса (Mark Lawrence), Дуга Ли (Doug Lea) из SUNY Oswego, Грегори Лонгшора (Gregory Longshore), Боба Линча (Bob Lynch) из Lynch Associates, Филиппа Милна (Philip Milne), консультанта, Марка Моррисси (Mark Morrissey) из научно-исследовательского института штата Орегон, Махеша Нилаканта (Mahesh Neelakanta) из Атлантического университета штата Флорида, Хао Фам (Hao Pham), Пола Филиона (Paul Phillion), Блейка Рагсдейла (Blake Ragsdell), Ильбера Рамадани (Ylber Ramadani) из университета имени Райерсона, Стюарта Реджеса (Stuart Reges) из университета штата Аризона, Рича Розена

(Rich Rosen) из Interactive Data Corporation, Питера Сандерса (Peter Sanders) из университета ЭССИ (ESSI), г. Ницца, Франция, доктора Пола Сангеру (Dr. Paul Sanghera) из университета штата Калифорния в Сан-Хосе и колледжа имени Брукса, Поля Севинка (Paul Sevinc) из Teamur AG, Деванг Ша (Devang Shah) из Oracle, Ричарда Сливчака (Richard Slywczak) из Исследовательского центра имени Гленна, НАСА, Бредли А. Смита (Bradley A. Smith), Стивена Стелтинга (Steven Stelting) из Sun Microsystems, Кристофера Тэйлора (Christopher Taylor), Люка Тэйлора (Luke Taylor) из Valtech, Джорджа Тхируватукала (George Thiruvathukal), Кима Топли (Kim Topley), автора книги *Core JFC, Second Edition*, Джанет Трауб (Janet Traub), Пола Тиму (Paul Tuma), консультанта, Питера Ван Дер Линдена (Peter van der Linden) из Motorola Mobile Devices, Берта Уолша (Burt Walsh), Джо Уанга (Joe Wang) из Oracle и Дана Ксю (Dan Xu) из Oracle.

Кей Хорстманн,
Сан-Франциско, декабрь 2012 г.

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 веб-сайт 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 электронные адреса:

E-mail: info@williamspublishing.com
WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1
в Украине: 03150, Киев, а/я 152