

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 переключиться на API-интерфейс операционной системы целевой платформы при взаимодействии с устройством или службой, которая не поддерживается платформой Java. То, как подобное 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

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