
Предисловие

В конце 1995 года язык программирования Java вырвался на арену Интернета и моментально завоевал популярность. Технология Java обещала стать *универсальным связующим звеном*, соединяющим пользователей с информацией, откуда бы эта информация ни поступала — от веб-серверов, из баз данных, поставщиков информации или любого другого источника, который только можно было себе вообразить. И действительно, у Java есть все, чтобы выполнить эти обещания. Это весьма основательно сконструированный язык, получивший признание всех основных участников рынка за исключением корпорации Microsoft. Его встроенные средства защиты и безопасности обнадежили как программистов, так и пользователей программ на Java. Язык Java изначально обладал встроенной поддержкой для решения таких сложных задач, как сетевое программирование, взаимодействие с базами данных и многопоточная обработка.

С 1995 года было выпущено восемь главных версий комплекта Java Development Kit. За последние 17 лет прикладной программный интерфейс (API) увеличился от 200 до 3 тысяч классов. Теперь прикладной интерфейс API охватывает самые разные предметные области, включая конструирование пользовательских интерфейсов, управление базами данных, интернационализация, безопасность и обработка данных в формате XML.

Книга, которую вы держите в руках, является первым томом девятого издания. С выходом каждого издания мы, ее авторы, старались как можно быстрее следовать очередному выпуску Java Development Kit, каждый раз переписывая ее, чтобы вы, читатель, могли воспользоваться преимуществами новейших средств Java. Настоящее издание обновлено с учетом новых языковых средств, появившихся в версии Java Standard Edition (SE) 7.

Как и все предыдущие издания этой книги, мы по-прежнему *адресуем ее серьезным программистам, которые хотели бы пользоваться Java для разработки настоящих проектов*. Мы, авторы этой книги, представляем себе вас, дорогой читатель, как грамотного специалиста с солидным опытом программирования на других языках, кроме Java, и надеемся, что вам не нравятся книги, которые полны игрушечных примеров вроде программ управления тостерами или животными в зоопарке либо “прыгающим текстом”. Ничего подобного вы не найдете в этой книге. Наша цель — помочь вам понять язык Java и его библиотеки в полной мере, а не создать иллюзию такого понимания.

В книге вы найдете массу примеров кода, демонстрирующих почти все обсуждаемые языковые и библиотечные средства. Эти примеры мы намеренно сделали как можно более простыми, чтобы сосредоточиться на основных моментах. Тем не менее они в большинстве своем совсем не игрушечные и не “срезают острых углов”. Все они могут послужить вам неплохой отправной точкой для разработки собственного кода.

Мы предполагаем, что вы стремитесь (и даже жаждете) узнать обо всех расширенных средствах, которые Java предоставляет в ваше распоряжение. В частности, мы предлагаем подробное рассмотрение следующих тем.

- Объектно-ориентированное программирование.
- Рефлексия и прокси-классы.
- Интерфейсы и внутренние классы.
- Модель приемников событий.
- Проектирование графического пользовательского интерфейса инструментальными средствами библиотеки Swing.
- Обработка исключений.
- Обобщенное программирование.
- Архитектура коллекций.
- Многопоточное программирование.

В связи со стремительным ростом библиотеки классов Java одного тома оказалось недостаточно для описания всех языковых средств Java, о которых следует знать серьезным программистам. Поэтому книга была разделена на два тома. В первом томе, который вы держите в руках, главное внимание уделяется фундаментальным понятиям языка Java, а также основам программирования пользовательского интерфейса. А второй том посвящен средствам разработки приложений масштаба предприятия и усовершенствованному программированию пользовательских интерфейсов. В нем вы найдете подробное обсуждение следующих тем.

- Файлы и потоки ввода-вывода.
- Распределенные объекты.
- Базы данных.
- Расширенные компоненты графического пользовательского интерфейса.
- Собственные методы.
- Обработка данных в формате XML.
- Сетевое программирование.
- Усовершенствованная графика.
- Интернационализация.
- Компонент JavaBeans.
- Аннотации.

При написании книги ошибки и неточности неизбежны. И нам очень важно знать о них. Но мы бы, конечно, предпочли узнать о каждой из них только один раз. Поэтому мы разместили перечень часто задаваемых вопросов, исправлений ошибок и обходных приемов по адресу <http://horstmann.com/corejava>, куда вы можете обращаться за справкой.

Краткий обзор книги

В **главе 1** дается краткий обзор тех функциональных возможностей языка Java, которыми он отличается от других языков программирования. В ней сначала поясняется, что было задумано разработчиками Java и в какой мере им удалось воплотить задуманное в жизнь. Затем приводится краткая история развития язык Java и показывается, как он стал тем, чем он есть в настоящее время.

В **главе 2** сначала поясняется, как загрузить и установить инструментарий JDK, а также примеры программ к этой книге. Затем рассматривается весь процесс компиляции и запуска трех типичных программ на Java (консольного приложения, графического приложения и апплета) только средствами JDK, текстового редактора, специально ориентированного на Java, а также интегрированной среды разработки на Java.

В **главе 3** начинается обсуждение языка программирования Java. В этой главе излагаются самые основы: переменные, циклы и простые функции. Если у вас имеется опыт программирования на C или C++, вам нетрудно будет усвоить материал этой главы, поскольку синтаксис этих языковых средств, по существу, ничем не отличается в Java. А если вам приходилось программировать на языках, не похожих на C, например на Visual Basic, прочитайте эту главу с особым вниманием.

Ныне объектно-ориентированное программирование (ООП) — господствующая методика программирования, и ей в полной мере отвечает язык Java. В **главе 4** представлено понятие *инкапсуляции* — первой из двух фундаментальных составляющих объектной ориентации, а также механизмы, реализующие ее в Java: классы и методы. В дополнение к правилам языка Java здесь также приводятся рекомендации по правильному объектно-ориентированному проектированию. И, наконец, в этой главе будет представлен замечательный инструмент `javadoc`, форматирующий комментарии из исходного кода в набор веб-страниц с перекрестными ссылками. Если у вас имеется опыт программирования на C++, можете лишь бегло просмотреть эту главу. А тем, кому раньше не приходилось программировать на объектно-ориентированных языках, придется потратить больше времени на усвоение принципов ООП, прежде чем изучать Java дальше.

Классы и инкапсуляция — это лишь часть методики ООП, и поэтому в **главе 5** представлен еще один ее краеугольный камень — *наследование*. Наследование позволяет взять существующий класс и модифицировать его в соответствии с конкретными потребностями программирующего. Это — основополагающий прием программирования на Java. Механизм наследования в Java очень похож на аналогичный механизм в C++. Опять же программирующие на C++ могут сосредоточить основное внимание лишь на языковых отличиях в реализации наследования.

В **главе 6** поясняется, как пользоваться в Java понятием *интерфейса*. Интерфейсы дают возможность выйти за пределы простого наследования, описанного в главе 5. Овладение интерфейсами позволит в полной мере воспользоваться объектно-ориентированным подходом к программированию на Java. В этой главе рассматривается также удобное языковое средство Java, называемое *внутренними классами*. Внутренние классы помогают сделать код более ясным и кратким.

С **главы 7** начинается серьезное прикладное программирование. Всякий программирующий на Java должен знать хотя бы немного о программировании графических пользовательских интерфейсов, и в этом томе вы найдете его основы. В этой главе будет показано, как создаются окна, как в них выполняется раскраска, рисуются

геометрические фигуры, форматируется текст многими шрифтами и как изображения выводятся на экран.

В **главе 8** подробно обсуждается модель событий AWT — *абстрактного оконного инструментария*. Здесь будет показано, как писать код, реагирующий на такие события, как щелчки кнопкой мыши или нажатия клавиш. Попутно вам предстоит ознакомиться с тем, как правильно обращаться с базовыми элементами графического пользовательского интерфейса вроде кнопок и панелей.

Глава 9 посвящена более подробному обсуждению инструментальных средств Swing. Набор инструментов Swing позволяет строить межплатформенный графический пользовательский интерфейс. В этой главе вы ознакомитесь с различными видами кнопок, текстовых компонентов, рамок, ползунков, комбинированных списков, меню и диалоговых окон. Но знакомство с некоторыми из более совершенных компонентов будет отложено до второго тома настоящего издания.

В **главе 10** будет показано, как разворачивать прикладные программы, будь то приложения или апплеты. В частности, в ней описывается, как пакетировать программы в файлы формата JAR и как доставлять приложения через Интернет с помощью технологии Java Web Start и механизмов апплетов. И, наконец, в этой главе поясняется, каким образом программы на Java способны сохранять и извлекать информацию о конфигурации после своего разворачивания.

Глава 11 посвящена *обработке исключений* — надежному механизму Java, призванному учитывать тот факт, что непредвиденные ситуации могут возникать и в грамотно написанных программах. Исключения обеспечивают эффективный способ отделения кода нормальной обработки от кода обработки ошибок. Но даже после оснащения прикладной программы проверкой всех возможных исключительных ситуаций в ней все-таки может произойти неожиданный сбой. Во второй части этой главы будет представлено немало полезных советов по организации отладки программ. И, наконец, в главе рассматривается весь процесс отладки на конкретном примере.

В **главе 12** дается краткий обзор *обобщенного программирования* — главного преимущества версии Java SE 5.0. Обобщенное программирование делает прикладные программы легче читаемыми и более безопасными. В этой главе будет показано, как применяется строгая типизация, исключается потребность в неприглядном и небезопасном приведении типов и как преодолеваются трудности на пути совместимости с предыдущими версиями Java.

Глава 13 посвящена архитектуре коллекций на платформе Java. Всякий раз, когда требуется сначала собрать множество объектов, а в дальнейшем извлечь их, приходится обращаться к коллекции, которая наилучшим образом подходит для конкретных условий, вместо того чтобы сбрасывать их в обычный массив. В этой главе будут продемонстрированы те преимущества, которые дают стандартные, предварительно подготовленные коллекции.

Глава 14 завершает первый том настоящего издания обсуждением многопоточной обработки, которая позволяет программировать выполняемые параллельно задачи. (Под *поток* здесь понимается поток управления командами в программе.) В этой главе будет показано, как создаются потоки и осуществляется их синхронизация. Средства многопоточной обработки существенно изменились после выхода версии Java SE 5.0, и в этой главе рассказывается о ее новых механизмах.

В **приложении А** перечислены зарезервированные слова языка Java.

Условные обозначения

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



НА ЗАМЕТКУ! Этой пиктограммой выделяются замечания.



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 `InputTest/InputTest.java`

Примеры исходного кода

Все примеры исходного кода, приведенные в этой книге, доступны в архивированном виде на посвященном ей веб-сайте по адресу <http://horstmann.com/corejava>. Их можно извлечь из файла архива любой из распространенных программ разархивирования или с помощью утилиты `jar`, входящей в состав набора Java Development Kit (JDK). Подробнее об установке JDK и примеров кода речь пойдет в главе 2.