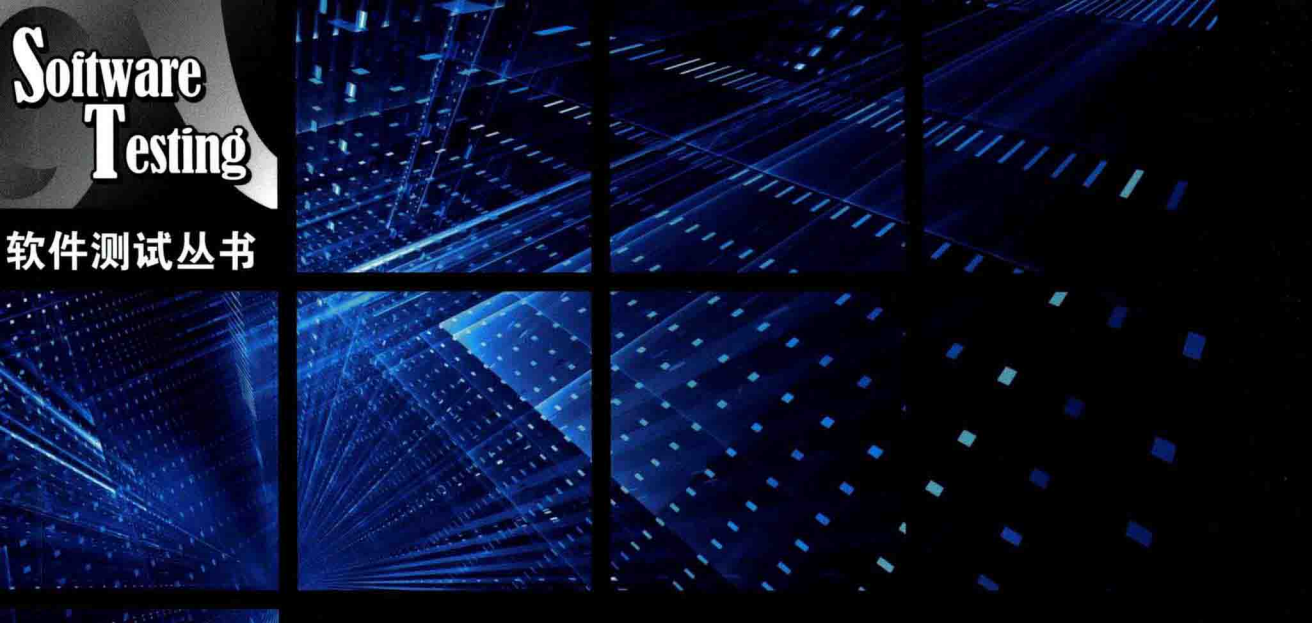


精通移动App测试实战：技术、工具和案例

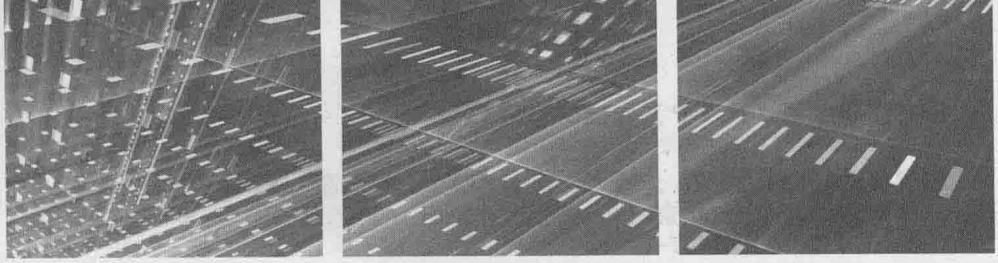
Proficiency in Mobile Phone System Testing: Techniques, Tools and Practices

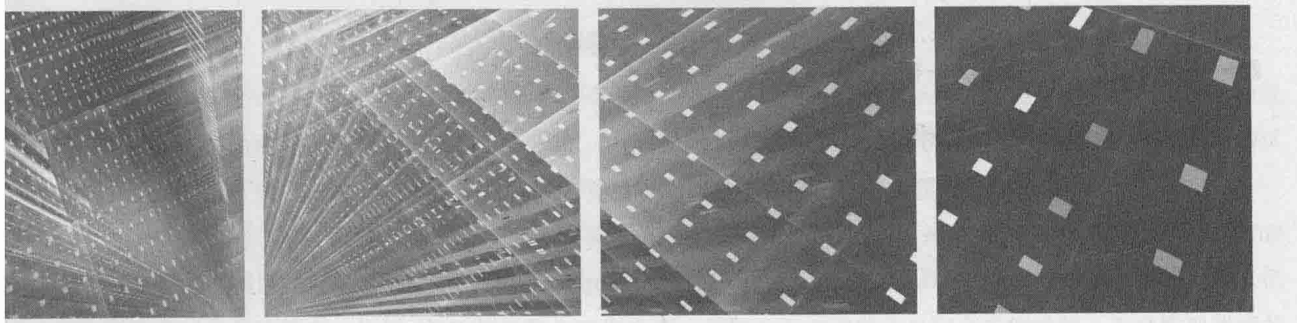
于涌 王磊 曹向志 编著

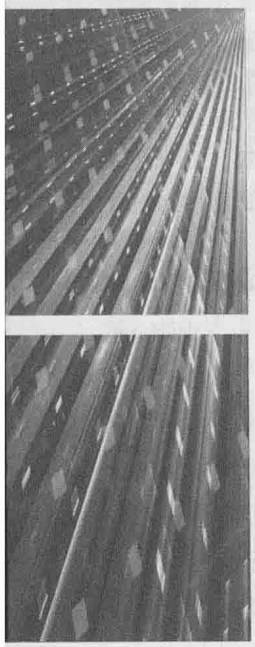
- **全面覆盖主流的移动App测试工具：**
  Monkey、MonkeyRunner、Robotium、UIAutomator、Appium。
- **移动App测试流程全覆盖：**
  操作系统剖析、系统调试、自动化测试脚本开发、测试用例实施、测试用例批量执行、持续集成等。
- **移动App自动化测试框架全实例讲解。**
- **移动App客户端性能测试全实例剖析：**
  移动端性能测试指标；
  移动端性能测试工具；
  LoadRunner在移动端性能测试中的应用。

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

软件测试丛书







精通移动App测试实战：技术、工具和案例

Proficiency in Mobile Phone System Testing: Techniques, Tools and Practices

于涌 王磊 曹向志 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

精通移动App测试实战 ：技术、工具和案例 / 于涌，王磊，曹向志编著. -- 北京 ：人民邮电出版社，2016.4
ISBN 978-7-115-41707-7

Ⅰ. ①精… Ⅱ. ①于… ②王… ③曹… Ⅲ. ①移动终端－应用程序－程序测试－研究 Ⅳ. ①TN929.53

中国版本图书馆CIP数据核字(2016)第037571号

内 容 提 要

本书全面讲解了移动平台测试方面的技术、技巧、工具和测试用例等实战知识。内容涵盖主流的测试工具，包括 JUnit、Monkey、MonkeyRunner、Robotium、UIAutomator、Appium，以及性能测试利器 LoadRunner、手机端性能监控工具 Emmagee 等；重点讲解移动平台的主要实战技术，如单元测试、功能测试、性能测试、UI 测试、手游测试、自动化测试、测试用例管理、持续集成、脚本录制等。书中结合实例对各个工具进行深入讲解，真正做到学以致用。本书既是一本真正帮助读者学习移动测试中用到的所有技术的实战教程，也是一本名副其实的、贴近实战的移动端测试权威指南。

本书适合测试初学者、测试工程师、测试经理、移动开发人员和游戏开发人员学习借鉴，也可以作为大专院校相关专业师生和培训学校的教学用书。

◆ 编　　著　于　涌　王　磊　曹向志
责任编辑　张　涛
责任印制　张佳莹　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：28.25
字数：693 千字　　2016 年 4 月第 1 版
印数：1－3 000 册　　2016 年 4 月河北第 1 次印刷

定价：69.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第 8052 号

前　言

本书是测试专家、畅销书作者多年实战经验的总结，涵盖主流的测试工具，包括众多的测试实例，涵盖单元测试、功能测试、性能测试、UI 测试、手游测试、自动化测试、测试用例管理、持续集成等移动测试中用到的所有实战技术，是一本贴近实战的移动端测试参考大全。本书主要内容如下。

书中讲解了单元测试，介绍了 JUnit 框架、单元测试实施、创建基于 Android 的测试项目和应用 JUnit 对 Android 项目进行单元测试；讲解了 Android 提供的一个通用的调试工具 ADB，借助这个工具，可以很好地调试开发的程序，包括 ADB 相关指令实例讲解、获取手机处理器信息指令实例讲解、手机模拟器相关的一些操作命令实例讲解、模拟器相关命令实例讲解、创建 Android 项目相关命令实例讲解、基于控制台命令行相关命令使用；讲解了 Android 系统自带的一个命令行工具 Monkey，Monkey 可以向被测试的应用程序发送伪随机的用户事件（如按键、触屏、手势等），Monkey 测试是一种测试软件稳定性、健壮性的快速有效的方法。包括 Monkey 工具使用、Monkey 测试示例、Monkey 相关参数讲解、Monkey 相关命令介绍、Monkey 脚本执行等；结合实例讲解了由 Google 开发、用于 Android 系统自动化测试的 MonkeyRunner 工具，包括 MonkeyRunner 工具使用、MonkeyRunner 测试示例、MonkeyRunner 脚本手工编写、MonkeyRunner 样例脚本等；书中还讲解了一款 Android 自动化测试框架 Robotium，它主要针对 Android 平台的应用进行黑盒自动化测试，提供了模拟各种手势操作（如点击、长按、滑动等）、查找和断言机制的 API，能够对各种控件进行操作。用 Robotium 结合 Android 官方提供的测试框架可以达到对应用程序进行自动化测试的目的，如用 Robotium 实现对 APK 或有源码的项目实施测试、用 Robotium Recorder 录制脚本、用 Robotium 获取控件，以及测试用例脚本的批量运行和持续集成等；讲解了 UI 测试工具 UIAutomator，它包含了创建 UI 测试的各种 API 和执行自动化测试的引擎；UIAutomator 接口丰富、易用，可以支持所有 Android 事件操作，非常适合做 UI 测试；Appium 是一个自动化测试开源工具，支持 iOS 和 Android 平台上的移动原生应用、移动 Web 应用和混合应用测试；Appium 是一个跨平台的工具，它允许测试人员使用同样的接口基于不同的平台（iOS、Android）编写自动化测试脚本，这样大大增加了 iOS 和 Android 测试用例的复用性，还讲解了自动化测试工具 Appium 实战、Appium 环境部署、Appium 元素定位的 3 个利器、多种界面控件的定位方法、多种界面控件的操作方法、捕获异常和创建快照等；书中最后结合案例讲解了移动平台的性能测试，性能测试的 8 大分类，移动端的性能指标，移动端性能测试工具，如手机端的性能监控工具 Emmagee、LoadRunner 在移动端性能测试中的应用等。TraceView 是 Android 平台自带的一个很好用的性能分析工具，它可以通过图形化的方式让我们了解要跟踪的应用程序的性能；Systrace 是 Android 4.1（API：16）以后引入的一个用于做性能分析的工具，该工具可以定时收集和监测 Android 设备的相关信息，它显示了每个线程或者进程在给定的时间里占用 CPU 的情况；Emmagee 是网易杭州研究院 QA 团队开发的一个简单易用的 Android 性能监测工具，主要用于监控单个手机应用的 CPU、内存、流量、启动耗时、电量、电流等性能状态的变化，且用户可以自定义配置监控的采样频率及性能的实时显示，并最终生成一份性能统计文件；LoadRunner 的最新版本为 LoadRunner 12.0，结合目前移动市场性能测试的需要，LoadRunner 也提供了一些基于移动平台

的协议和相应的工具，本书中都会有讲解。

写作过程中，作者倾尽全力，由于时间紧，加之水平有限，书中错误在所难免，诚请广大读者给予指正，以便再版时修正完善，本书答疑 QQ 群为 191026652，本书编辑联系邮箱为 zhangtao@ptpress.com.cn。

本书适合测试初学者、测试工程师、测试经理、移动开发人员和游戏开发人员学习使用，也可以作为大专院校相关专业师生和培训学校的教学用书。

目　录

Chapter

1

第1章

Android 系统基础内容介绍

工欲善其事必先利其器，因为本书主要是针对移动平台讲解测试方面的内容，所以对移动平台目前主流的 Android 系统有一个了解十分必要，下面我们就一起来了解一下这个操作系统相关的知识内容。

1.1 Android 系统介绍

Android 一词的原意指“机器人”，同时也是 Google 于 2007 年 11 月 5 日宣布的基于 Linux 平台的开源手机操作系统的名称，该平台由操作系统、中间件、用户界面和应用软件组成。

Android 的 Logo 是由 Ascender 公司设计的，诞生于 2010 年，其设计灵感源于男女厕所门上的图形符号。布洛克绘制了一个简单的机器人，它的躯干就像锡罐的形状，头上还有两根天线，Android 小机器人便诞生了。

1.2 Android 系统架构

从图 1-1 中我们不难发现 Android 的系统架构采用了分层的架构，分为 4 个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和 Linux 内核层。那么它们每层都是用来做什么的呢？

1. *应用程序层*

应用层是用 Java 语言编写的运行在 Android 平台上的程序，比如一些手机游戏和基于手机端的应用等，如图 1-1 所示，最上面的 Applications 层。

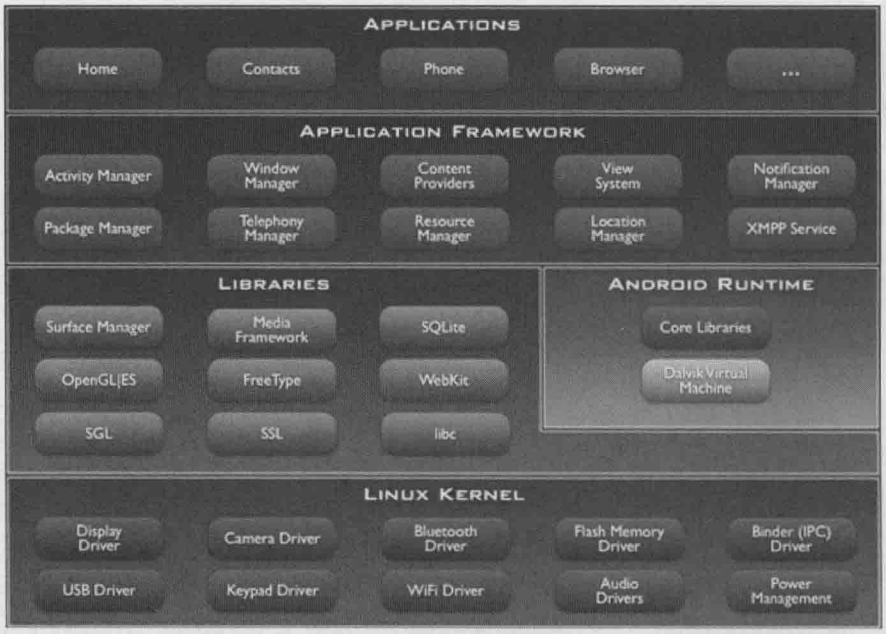


图 1-1 Android 系统架构图

2. 应用程序框架层

应用程序框架层是编写 Google 发布的核心应用时所使用的 API 框架，开发人员可以使用这些框架来开发自己的应用程序，这样可以简化程序开发的架构设计，如图 1-1 所示，第二层，即 Application Framework 层，其提供的主要 API 框架如下。

活动管理器：主要用来管理应用程序声明周期，并提供常用的导航退回功能。

窗口管理器：主要用来管理所有的窗口程序。

内容提供器：它可以让一个应用访问另一个应用的数据，或共享它们自己的数据。

视图管理器：主要用来构建应用程序，比如列表、表格、文本框及按钮等。

通知管理器：主要用来设置在状态栏中显示的提示信息。

包管理器：主要用来对 Android 系统内的程序进行管理。

电话管理器：主要用来对联系人及通话记录等信息进行管理。

资源管理器：主要用来提供非代码资源的访问，例如本地字符串、图形及布局文件等。

位置管理器：主要用来提供使用者的当前位置等信息，如 GPRS 定位。

XMPP Service：XMPP 服务。

3. 系统运行库层

系统运行库层主要提供 Android 程序运行时需要的一些类库，这些类库一般是使用 C/C++ 语言编写的。另外，该层还包含了 Android 运行库。如图 1-1 所示，第三层，系统运行库层中包含的主要库如下。

libc：C 语言标准库，系统最底层的库，C 语言标准库通过 Linux 系统来调用。

Surface Manager：主要管理多个应用程序同时执行时各个程序之间的显示与存取，并且为多个应用程序提供 2D 和 3D 图层的无缝融合。

SQLite：关系数据库。

OpenGL|ES：3D 效果的支持。

Media Framework：Android 系统多媒体库，该库支持多种常见格式的音频、视频的回放和录制。

WebKit：Web 浏览器引擎。

SGL：2D 图形引擎库。

SSL：位于 TCP/IP 协议与各种应用层协议之间，为数据通信提供支持。

FreeType：位图及矢量库。

系统运行库层中还包含了一个 Dalvik 虚拟机，相对于桌面系统和服务器系统运行的虚拟机而言，它不需要很快的 CPU 计算速度和大量的内存空间。因此，它非常适合在移动终端上使用。

4. 系统内核层

Android 的核心系统服务基于 Linux 2.6 内核，该内核拥有安全性、内存管理、进程管理、网络协议栈和驱动模型等。同时它也作为硬件和软件栈之间的抽象层，而 Android 更多的是需要一些与移动设备相关的驱动程序，比如显示驱动、USB 接口驱动、蓝牙驱动、电源驱动、Wi-Fi 驱动等，如图 1-1 所示，最下面即为该层。

1.3 Android 权限系统

Android 操作系统其实是一个多用户的 Linux 操作系统，每个 Android 应用都使用不同的用户，运行在自己的安全沙盘里。系统为应用的所有文件设置权限，这样一来只有同一个用户的应用可以访问它们。每个应用都有自己单独的虚拟机，这样应用的代码在运行时是隔离的，即一个应用的代码不能访问或意外修改其他应用的内部数据。

每个应用都运行在单独的 Linux 进程中，当应用被执行时，Android 都会为其启动一个 Java 虚拟机，因此不同的应用运行在相互隔离的环境中。Android 系统采用最小权限原则确保系统的安全性。也就是说，每个应用默认只能访问满足其工作所需的功能，而不能访问其无权使用的功能。那么我们要实现移动平台的自动化测试时，比如应用 Robotium，就涉及到它和被测试应用的交互，如果是上面的机制是不是意味着我们没有办法实施自动化测试呢？当然能够解决该类问题，不同的应用可以运行在相同的进程中，要实现这个功能，就必须保证应用使用相同的密钥签名、在 AndroidManifest.xml 文件中为这些应用分配相同的 Linux 用户 ID。同时，如果应用需要用到照相、Wi-Fi、蓝牙、SD 卡的读写操作等都需要进行授权。

1.4 Android 相关的一些属性简介

Activity（活动）：我们在后续的图书内容阅读过程中经常会看到这个词，那么什么是活动呢，就像我们在操作一些应用软件，比如 Word，它出现的每一个功能界面，比如在编辑文件、改变字体大小后，我们单击工具条的“保存”按钮；或者是一个拼车的手机应用，我们约车的时候，其也会提供一个界面，需要我们指定出发的地点、目的地、出发时间等信息，单击“确认预约”按钮。它们都是软件系统和我们用户的一个交互，这个和我们交互的界面就叫一个“活动”。

Service（后台服务）：后台服务通常没有交互的图形界面，是多用于处理长时间任务，而不影响前台用户体验的组件。如我们一边看着“微信”应用的朋友圈内容，一边欣赏着手机的音乐，怡然自得的时候是否知道其有一个后台播放音乐的服务呢？

Content Provider（内容供应组件）：内容供应组件用来管理应用的可共享部分的数据。例如，应用将数据存储在文件系统或者 SQLite 数据库中，通过内容供应组件，其他的应用也可以对这些数据进行查询。例如，我们手机自带联系人信息，其他的应用只要有相应的权限就可以通过查询内容供应组件来查询该联系人的相关信息。

Broadcast Receivers（广播接收组件）：在 Android 里面有各种各样的广播，电池的使用状态、电话的接收和短信的接收等都会产生一个广播，应用程序开发者也可以监听这些广播并做出程序逻辑的处理。

1.5 搭建 Android 开发环境

基于移动平台的自动化测试，通常都需要我们有一定的语言基础、单元测试基础和 IDE（Integrated Development Environment，集成开发环境）。软件是用于程序开发环境的应用程序，

一般包括代码编辑器、编译器、调试器和图形用户界面工具。它是集成了代码编写、编译、调试和分析等一体化的辅助开发人员开发软件的应用软件，目前应用比较广泛的 IDE 有 VisualStudio、Eclipse 等。

根据工作环境和个人喜好不同，既可以在 Windows 系统环境下部署 Android 开发环境，也可以在 Linux 系统环境下部署 Android 开发环境，关于这方面的资料在互联网上可大量查询。鉴于目前大多数测试人员应用 Windows 系统，这里主要以 Windows 7 系统环境为例，向大家讲解如何在 Windows 7 64 位系统环境下搭建 Android 开发环境。

1.5.1 JDK 的安装与配置

Android 应用程序开发使用 Java 语言，所以我们首先要搭建 Java 程序开发运行环境。Java 的开发环境称为 JDK（Java Development Kit），是 Sun Microsystems 针对 Java 应用开发人员开发的产品，JDK 已经成为使用最广泛的 Java SDK（Software Development Kit，软件开发工具包）。

可以访问“http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html”这个地址来下载最新的 JDK，如图 1-2 所示。

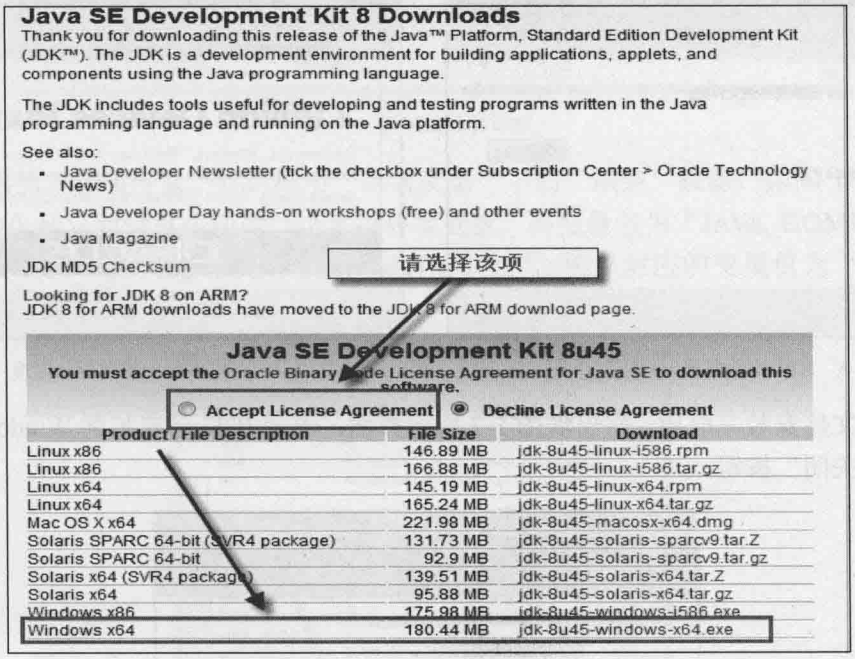


图 1-2 JDK 下载界面信息

从图 1-2 中我们可以看到 Oracle 提供了基于不同操作系统的 JDK 包，这里因为我们应用的是 Windows 7 64 位的操作系统，所以要下载图 1-2 所示的“jdk-8u45-windows-x64.exe”文件。

接下来双击“jdk-8u45-windows-x64.exe”文件，将出现图 1-3 所示界面。

单击“下一步”按钮，将出现图 1-4 所示界面。

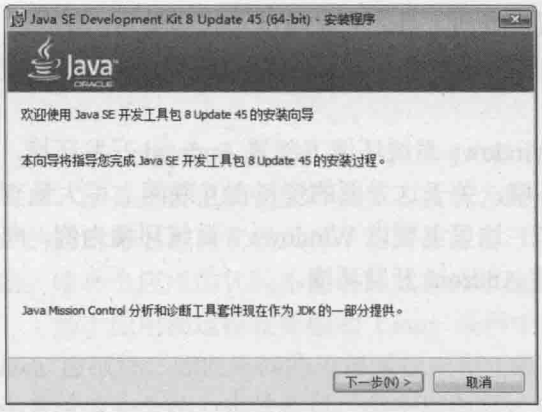


图 1-3 JDK 安装向导－安装程序

图 1-4 JDK 安装向导－定制安装

单击“下一步”按钮，将出现图 1-5 所示界面，这里我们选择其默认的安装目录，不做改变。

单击“下一步”按钮，将出现图 1-6 所示界面，开始安装 JDK 的相关文件。

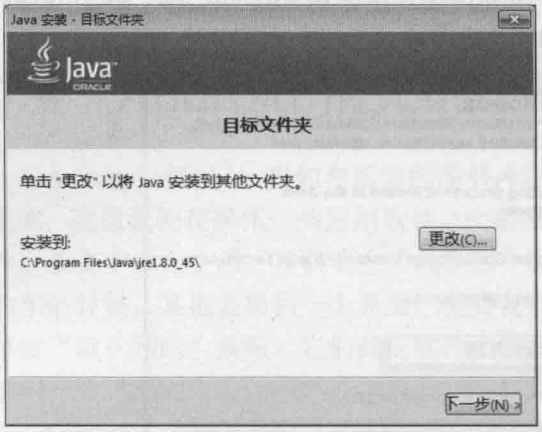


图 1-5 JDK 安装向导－目标文件夹

图 1-6 JDK 安装向导－进度

待相关文件安装完成后，将出现图 1-7 所示界面，表示 JDK 已安装到 Windows 7 操作系统，单击“关闭”按钮。

图 1-7 JDK 安装向导－完成