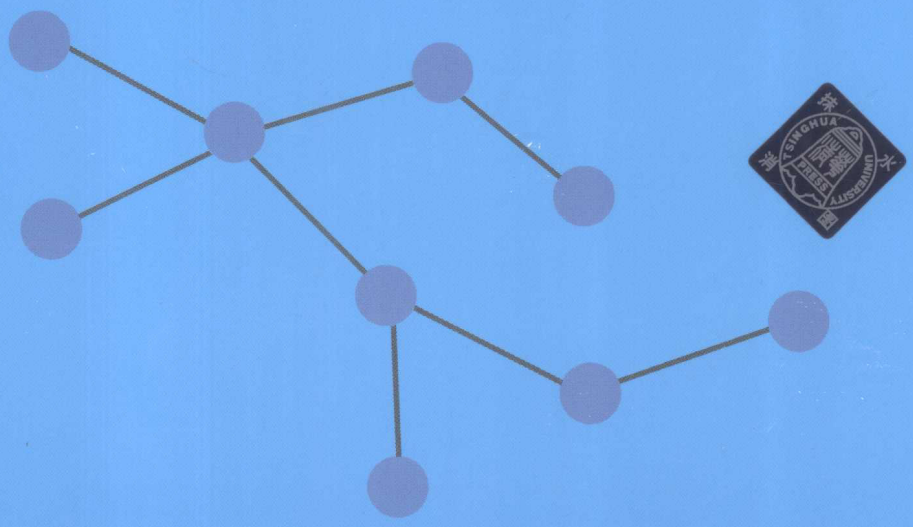
高等院校计算机任务驱动教改教材

移动互联网应用开发（基于Android平台）

李维勇 主 编 杜亚杰 石 建 副主编



清华大学出版社

高等院校计算机任务驱动教改教材

移动互联网应用开发

（基于Android平台）

李维勇 主 编 杜亚杰 石 建 副主编

清华大学出版社
北京

内 容 简 介

本书是基于 Android Lollipop 平台进行移动互联网应用开发的入门级教程，通过众多开源案例项目，全面系统地介绍移动互联网应用开发的方法、技巧和模式。

全书共分为 9 章，从 Android 应用设计者的角度系统讲解了从事 Android 移动互联网应用开发必须要掌握的 Android 平台的相关技术和特性，主要内容包括数据流与数据解析，网络连接与管理、Android 中的 Socket 编程与 HTTP 编程，Web 应用编程、开放接口编程、Google 云服务技术等，全面总结了 Android 网络编程的基本原理、设计理念和设计模式，最后通过一个综合的案例项目阐述了 Android 移动互联网应用开发的方法和技巧。

本书以案例贯穿全程，知识结构清晰，语言简洁，易于读者学习和提高，非常适合初学 Android 移动互联网应用开发的在校大学生和希望系统掌握 Android 互联网编程的开发人员。

图书在版编目(CIP)数据

移动互联网应用开发：基于 Android 平台/李维勇主编. --北京：清华大学出版社，2016
高等院校计算机任务驱动教改教材
ISBN 978-7-302-42121-4

Ⅰ. ①移… Ⅱ. ①李… Ⅲ. ①移动终端－技术开发－高等学校－教材 Ⅳ. ①TN87

中国版本图书馆 CIP 数据核字(2015)第 267362 号

责任编辑：张龙卿
封面设计：徐日强
责任校对：刘　静
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795764
印 刷 者：北京市人民文学印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：18.75　　**字　　数**：454 千字
版　　次：2016 年 3 月第 1 版　　**印　　次**：2016 年 3 月第1 次印刷
印　　数：1～2500
定　　价：36.00 元

产品编号：065950-01

编审委员会

出版说明

我国高职高专教育经过十几年的发展，已经转向深度教学改革阶段。教育部于2006年12月发布了教高[2006]第16号文件《关于全面提高高等职业教育教学质量的若干意见》，大力推行工学结合，突出实践能力培养，全面提高高职高专教学质量。

清华大学出版社作为国内大学出版社的领跑者，为了进一步推动高职高专计算机专业教材的建设工作，适应高职高专院校计算机类人才培养的发展趋势，根据教高[2006]第16号文件的精神，2007年秋季开始了切合新一轮教学改革的教材建设工作。该系列教材一经推出，就得到了很多高职院校的认可和选用，其中部分书籍的销售量超过了3万册。现重新组织优秀作者对部分图书进行改版，并增加了一些新的图书品种。

目前国内高职高专院校计算机网络与软件专业的教材品种繁多，但符合国家计算机网络与软件技术专业领域技能型紧缺人才培养培训方案，并符合企业的实际需要，能够自成体系的教材还不多。

我们组织国内对计算机网络和软件人才培养模式有研究并且有过一段实践经验的高职高专院校，进行了较长时间的研讨和调研，遴选出一批富有工程实践经验和教学经验的双师型教师，合力编写了这套适用于高职高专计算机网络、软件专业的教材。

本套教材的编写方法是以任务驱动、案例教学为核心，以项目开发为主线。我们研究分析了国内外先进职业教育的培训模式、教学方法和教材特色，消化吸收优秀的经验和成果。以培养技术应用型人才为目标，以企业对人才的需要为依据，把软件工程和项目管理的思想完全融入教材体系，将基本技能培养和主流技术相结合，课程设置重点突出、主辅分明、结构合理、衔接紧凑。教材侧重培养学生的实战操作能力，学、思、练相结合，旨在通过项目实践，增强学生的职业能力，使知识从书本中释放并转化为专业技能。

一、教材编写思想

本套教材以案例为中心，以技能培养为目标，围绕开发项目所用到的知识点进行讲解，对某些知识点附上相关的例题，以帮助读者理解，进而将知识转变为技能。

考虑到是以“项目设计”为核心组织教学，所以在每一学期配有相应的实训课程及项目开发手册，要求学生在教师的指导下，能整合本学期所学的知识内容，相互协作，综合应用该学期的知识进行项目开发。同时，在教材中采用了大量的案例，这些案例紧密地结合教材的各个知识点，循序渐进，由浅入深，在整体上体现了内容主导、实例解析、以点带面的模式，配合课程后期以项目设计贯穿教学内容的教学模式。

软件开发技术具有种类繁多、更新速度快的特点。本套教材在介绍软件开发主流技术的同时，帮助学生建立软件相关技术的横向及纵向的关系，培养学生综合应用所学知识的能力。

二、丛书特色

本系列教材体现目前工学结合的教改思想，充分结合教改现状，突出项目面向教学和任务驱动模式教学改革成果，打造立体化精品教材。

（1）参照和吸纳国内外优秀计算机网络、软件专业教材的编写思想，采用本土化的实际项目或者任务，以保证其有更强的实用性，并与理论内容有很强的关联性。

（2）准确把握高职高专软件专业人才的培养目标和特点。

（3）充分调查研究国内软件企业，确定了基于 Java 和. NET 的两个主流技术路线，再将其组合成相应的课程链。

（4）教材通过一个个教学任务或者教学项目，在做中学、学中做，以及边学边做，重点突出技能培养。在突出技能培养的同时，还介绍解决思路和方法，培养学生未来在就业岗位上的终身学习能力。

（5）借鉴或采用项目驱动的教学方法和考核制度，突出计算机网络、软件人才培训的先进性、工具性、实践性和应用性。

（6）以案例为中心，以能力培养为目标，并以实际工作的例子引入概念，符合学生的认知规律。语言简洁明了、清晰易懂，更具人性化。

（7）符合国家计算机网络、软件人才的培养目标；采用引入知识点、讲述知识点、强化知识点、应用知识点、综合知识点的模式，由浅入深地展开对技术内容的讲述。

（8）为了便于教师授课和学生学习，清华大学出版社正在建设本套教材的教学服务资源。清华大学出版社网站（www. tup. com. cn）免费提供教材的电子课件、案例库等资源。

高职高专教育正处于新一轮教学深度改革时期，从专业设置、课程体系建设到教材建设，依然是新课题。希望各高职高专院校在教学实践中积极提出意见和建议，并及时反馈给我们。清华大学出版社将对已出版的教材不断地修订、完善，提高教材质量，完善教材服务体系，为我国的高职高专教育继续出版优秀的高质量教材。

清华大学出版社

高职高专计算机任务驱动模式教材编审委员会

2014 年 3 月

前　言

国家“十二五”规划明确提出要加快发展新一代信息技术，移动互联网是我国未来发展战略的重点方向。移动终端 App 开发、移动互联网平台开发、基于 HTML5 的移动 Web 开发是移动互联网行业人才需求的三大方向。作为移动互联网应用开发工程师，基于 TCP/IP 的网络编程能力、终端轻量级数据库开发能力、基于典型公共云平台的应用开发能力是其必备的职业技能。

本书以 Android SDK Lollipop 5.0 为开发平台，以 Eclipse 为集成开发环境，结合笔者近年来在手机软件研发和教学中积累的经验，详细介绍 Android 平台移动互联网应用开发的相关知识。

本书共分为 9 章。第 1 章介绍 Java 中 I/O 数据流的基本知识，包括 Android 平台下字节流与字符流的处理，以及对象序列化与编码转换的技术。第 2 章介绍 XML 和 JSON 两种数据格式及其在 Android 中的解析方法，介绍了 JSON 解析工具 Gson 和 Json-lib 的使用方法。第 3 章介绍 Android 平台下网络连接与管理的方法，Wi-Fi 网络连接与管理的知识，以及网络服务优化的技术。第 4 章介绍网络编程的基本知识及相关 API，介绍了 Android 中 Socket 编程的方法和步骤，以及 UDP 编程和 NIO 编程的知识。第 5 章介绍 Android 平台中 HTTP 编程的方法，包括基于 HttpURLConnection 和 HttpClient 访问网络的方法，介绍了 Android-Async-Http 和 Volley 连接框架的使用技术。第 6 章介绍使用 WebView 构建 Web 应用的方法，以及使用 HTML5 开发 Web App 的方法。第 7 章介绍 Android 中 Web Service 编程的方法，并通过人人网等案例介绍基于开放平台实现互联网编程的技术。第 8 章介绍 Google 云开发技术，包括 Google 云备份技术、云信息技术和云存储技术等。第 9 章通过分析 Philm 项目的设计和实现过程，介绍移动互联网应用开发的技术架构与设计模式。

本书紧密结合初学者的学习习惯和认知规律，采用了大量简单而又实用的设计案例，使得读者在阅读时不会有障碍，并可通过简单的代码移植就能够生成新的应用。书中采用的开源案例项目把与 Android 开发相关的技术及设计完美结合，别具一格，弥补了一些 Android 设计人员知识的不足。

本书由李维勇担任主编，杜亚杰、石建担任副主编，李建林教授主审。本书的编写得到了南京信息职业技术学院、金审学院等院校的大力支持和

帮助，杭州虹云网络科技有限公司为教材案例项目的策划、开发和测试提供了大量信息，清华大学出版社的编辑为本书的策划和出版提供了宝贵的经验和支持，在此表示衷心感谢。同时，本书在编写过程中，参考了大量的相关资料，吸取了许多同人的宝贵经验，在此一并致谢。

由于作者水平所限，疏漏难免，敬请广大读者提出宝贵意见和建议。教材配套课件、习题答案及源代码均可从清华大学出版社网站下载。

编　者

2016 年 1 月

目 录

第1章 数 据 流

移动互联网对网络数据流的处理要关注四个基本方面：数据流的编码、字节顺序、数据格式对应和取数。在编写网络程序时，必须注意这些问题，以使程序能正确处理通信的内容。

1.1 Java 中的 I/O

Java 中 I/O 操作主要是指使用 Java 进行输入、输出操作。Java 所有的 I/O 机制都是基于数据流进行输入输出的，这些数据流表示了字符或者字节数据的流动序列。Java 的 I/O 流提供了读写数据的标准方法。任何 Java 中表示数据源的对象都会提供以数据流的方式读写其数据的方法。

1.1.1 I/O 流

流是一组有顺序的、有起点和终点的字节集合，是对数据传输的总称或抽象。即数据在两设备间的传输称为流，流的本质是数据传输。当 Java 程序需要从数据源读取数据时，会开启一个到数据源的流。数据源可以是文件、内存或者网络等。同样，当程序需要输出数据到目的地时也一样会开启一个流，数据目的地也可以是文件、内存或者网络等。流的创建是为了更方便地处理数据的输入输出。

Java 根据数据传输特性将流抽象为各种类，以便于更直观地进行数据操作。流的常见分类方式包括两种。

(1) 根据处理数据类型的不同，分为字节流和字符流。

字节流也称为原始数据，需要用户读入后进行相应的编码转换。而字符流的实现是基于自动转换的，读取数据时会把数据按照 JVM 的默认编码自动转换成字符。

字节流由 java. io. InputStream 和 java. io. OutputStream 处理，而字符流由 java. io. Reader 和 java. io. Writer 处理。Reader 和 Writer 是 Java 后来版本新增加的处理类，以便使数据的处理更方便。

字节流和字符流的区别如下。

① 读写单位不同：字节流以字节(8bit)为单位，字符流以 16 位的 Unicode 码表示的字符为基本处理单位，根据码表映射字符，一次可能读多个字节。

② 处理对象不同：字节流能处理所有类型的数据(如图像、视频等)，而字符流只能处理字符类型的数据。

因此，在实际编程时，如果只是处理纯文本数据，就优先考虑使用字符流。除此之外都

使用字节流。

(2) 根据数据流向不同,分为输入流和输出流。

程序从输入流读取数据源(图 1-1),程序向输出流写入数据(图 1-2)。对输入流只能进行读操作,对输出流只能进行写操作,程序中需要根据待传输数据的不同特性而使用不同的流。

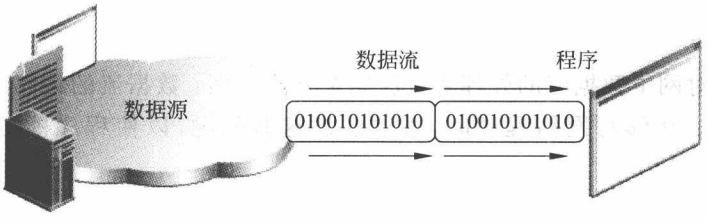


图 1-1 输入流

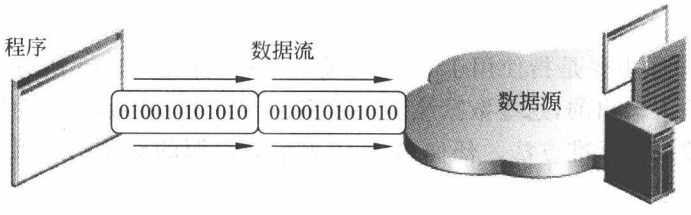


图 1-2 输出流

1.1.2 Java I/O 模型

Java 的 I/O 模型设计得非常优秀,它使用装饰者(Decorator)模式[①]来实现对各种输入输出流的封装。

Java I/O 主要包括如下三个部分。

(1) 流式部分:I/O 的主体部分。

(2) 非流式部分:主要包含一些辅助流式部分的类,如 File 类、RandomAccessFile 类和 FileDescriptor 等类。

(3) 其他类:文件读取部分与安全相关的类,如 SerializablePermission 类,以及与本地操作系统相关的文件系统的类,如 FileSystem 类、Win32FileSystem 类和 WinNTFileSystem 类。

对于流式部分,java.io 是大多数面向数据流的输入/输出类的主要软件包,按照功能可以分为 4 组。

- 基于字节操作的 I/O 接口:InputStream 和 OutputStream。
- 基于字符操作的 I/O 接口:Writer 和 Reader。
- 基于磁盘操作的 I/O 接口:File。
- 基于网络操作的 I/O 接口:Socket。

前两组主要是根据传输数据的数据格式,后两组主要是根据传输数据的方式。此外,

① 装饰者模式是面向对象编程领域中一种在不必改变原类文件和使用继承的情况下动态地往一个类中添加新行为的设计模式。

Java 也对块传输提供支持，在核心库 java. nio 中采用的便是块 I/O。流 I/O 的好处是简单易用，缺点是效率较低。块 I/O 效率很高，但编程比较复杂。

在 Java 中 I/O 操作的一般流程如下(以文件操作为例)：

① 使用 File 类打开一个文件；

② 通过字节流或字符流的子类，指定输出的位置；

③ 进行读/写操作；

④ 关闭输入/输出流。

提示：Java 中的 I/O 操作属于资源操作，使用完毕后一定要记得关闭。

1.1.3　I/O 异常

异常(Exception，又称为例外)是特殊的运行错误对象，对应着 Java 语言特定的运行错误处理机制。Java 的异常类是处理运行时错误的特殊类，每一种异常类对应一种特定的运行错误。所有的 Java 异常类都是系统类库中 Exception 类的子类。

Java 最常见的异常之一就是 java. io. IOException，包括以下 3 种。

(1) public class EOFException

非正常到达文件尾或输入流尾时抛出的异常。

(2) public class FileNotFoundException

当文件找不到时抛出的异常。

(3) public class InterruptedIOException

当 I/O 操作被中断时抛出的异常。

1.2　字　节　流

字节流主要是操作 byte 类型数据，以 byte 数组为主，主要操作类包括 InputStream 和 OutputStream。

1.2.1　InputStream

InputStream 是所有的输入字节流的父类，它是一个抽象类，必须依靠其子类实现各种功能。继承自 InputStream 的流都是向程序中输入数据的，且数据单位为字节(8bit)。

1. 主要方法

InputStream 提供的主要方法包括以下几种。

(1) public abstract int read() throws IOException

read()方法从输入流中读取一个 byte 的数据，返回 0～255 的 int 值。如果因为已经到达流末尾而没有可用的字节，则返回值为－1。在输入数据可用、检测到流末尾或者抛出异常前，此方法一直阻塞。

(2) public int read(byte[] b) throws IOException

read(byte[] b)从输入流中读取 b. length 个字节数据，并将其存储在缓冲区数组 b 中。返回值是读取的字节数。

(3) public int read(byte[] b, int off, int len) throws IOException

将输入流中最多 len 个数据字节读入偏移量为 off 的 b 数组中。但读取的字节也可能小于该值。以整数形式返回实际读取的字节数。

注意:

① 如果数组 b 的长度为 0,则不读取任何字节并返回 0;否则,尝试读取至少一个字节。

② 如果因为流位于文件末尾而没有可用的字节,则返回值为 -1;否则,至少读取一个字节并将其存储在 b 中。

③ read(byte[] b)方法将读取的第一个字节存储在元素 b[0]中,下一个存储在 b[1]中,其他依次类推。读取的字节数最多等于 b 的长度。设 k 为实际读取的字节数;这些字节将存储在 b[0]到 b[k-1]的元素中,不影响 b[k]到 b[b.length-1]的元素。

④ read(byte[] b, int off, int len)方法将读取的第一个字节存储在元素 b[off]中,下一个存储在 b[off+1]中,其他依次类推。读取的字节数最多等于 len。设 k 为实际读取的字节数;这些字节将存储在 b[off]到 b[off+k-1]的元素中,不影响 b[off+k]到 b[off+len-1]的元素。在任何情况下,b[0]到 b[off]的元素以及 b[off+len]到 b[b.length-1]的元素都不会受到影响。

⑤ read(byte[] b)和 read(byte[] b, int off, int len)这两个方法都是用来从流里读取多个字节的,但是这两个方法经常读取不到自己想要读取个数的字节。如 read(byte[] b)方法往往希望程序能读取到 b.length 个字节,而实际情况是,系统往往读取不了这么多。事实上,这个方法并不能保证读取这么多个字节,它只能最多读取这么多个字节(最少 1 个)。因此,如果要让程序读取 count 个字节(除非中途遇到 I/O 异常或者到了数据流的结尾),使用以下代码:

```
01  byte[] b = new byte[count];
02  int readCount = 0;
03  while (readCount < count) {
04      readCount += in.read(bytes, readCount, count - readCount);
05  }
```

(4) public int available() throws IOException

返回输入流中可以读取的字节数,这个方法必须由继承 InputStream 类的子类对象调用才有用。在一次读取多个字节时,经常调用 available()方法,这个方法可以在读写操作前先得知数据流里有多少个字节可以读取。如果这个方法用在从本地文件读取数据时,一般不会遇到问题。但如果是用于网络操作,可能会遇到输入阻塞,当前线程将被挂起。如果 InputStream 对象调用这个方法,它只会返回 0。

为了保证调用 available()方法从网络获取 count 个数据,使用以下代码:

```
01  int count = 0;
02  while (count == 0) {
03      count = in.available();
```

```
04  }
05  byte[] b = new byte[count];
06  in.read(b);
```

(5) public long skip(long n) throws IOException

跳过和丢弃此输入流中数据的 n 个字节。出于各种原因，skip()方法结束时跳过的字节数可能小于该数，也可能为 0。导致这种情况的原因很多，跳过 n 个字节之前已到达文件末尾只是其中一种可能。返回跳过的实际字节数。如果 n 为负值，则不跳过任何字节。

(6) public void close() throws IOException

关闭流并释放内存资源。

在 Android 中，存放在 Assets 目录(存放路径是 project/assets/file.name)中的文件会被原封不动地拷贝到 APK 中，而不会像其他资源文件那样被编译成二进制的形式。下面的代码演示了使用 InputStream 读取 Assets 中文件的方法。

```
01  public class InputStreamActivity extends Activity {
02      private String fileName = "InputStreamExample.txt";
03
04      private TextView mTextView;
05
06      @Override
07      protected void onCreate(Bundle savedInstanceState) {
08          super.onCreate(savedInstanceState);
09          setContentView(R.layout.activity_input_stream);
10
11          mTextView = (TextView) findViewById(R.id.tv);
12          mTextView.setText(getFromAsset(fileName));
13      }
14
15      private String getFromAsset(String fileName) {
16          String res = "";
17          try {
18              InputStream in = getResources().getAssets().open(fileName);
19              int length = in.available();
20              byte[] buffer = new byte[length];
21              in.read(buffer);
22              res = EncodingUtils.getString(buffer, "GB2312");
23          } catch (Exception e) {
24              e.printStackTrace();
25          }
26          return res;
27      }
28
29  }
```

2. 主要子类

ByteArrayInputStream、StringBufferInputStream、FileInputStream 是三种基本的介质