前 言

Java 语言是第一个完全融入网络的语言，Java 语言之所以适合编写网络应用程序，归功于它的以下优势：

（1）Java 语言与生俱来就是平台无关的。Java 程序能够运行在不同的平台上，运行在不同平台上的 Java 程序能够方便地进行网络通信。

（2）Java 语言具有完善的安全机制，可以对程序进行权限检查。这对网络程序至关重要。

（3）JDK 类库提供了丰富的网络类库（如套节字 API、JavaMail API 和 JDBC API 等），大大简化了网络程序的开发过程。

本书将展示如何利用 Java 网络类库来快速便捷地创建网络应用程序，致力于完成以下任务：

- 实现访问 HTTP 服务器的客户程序。
- 实现 HTTP 服务器。
- 实现多线程的服务器，以及非阻塞的服务器。
- 解析并展示 HTML 页面。
- 通过 JDBC API 访问数据库。
- 通过 JavaMail API 接收和发送电子邮件。
- 利用 RMI 框架实现分布式的软件系统。
- 进行安全的网络通信，对数据加密，验证身份，保证数据的完整性。
- 运用第三方开源软件框架，如 Axis、Spring 和 CXF，开发 Web 服务，实现分布式的软件系统。

本书的组织结构和主要内容

本书结合大量典性的实例，详细介绍了用 Java 来编写网络应用程序的技术。本书内容包括：Java 网络编程的基础知识、 套接字编程、非阻塞通信、创建 HTTP 服务器与客户程序、数据报通信、对象的序列化与反序列化、Java 反射机制、RMI 框架、JDBC API、JavaMail API、MVC 设计模式、安全网络通信、XML 数据处理和 Web 服务。以下图 P-1 展示了本书各个章之间的顺序渐进关系。

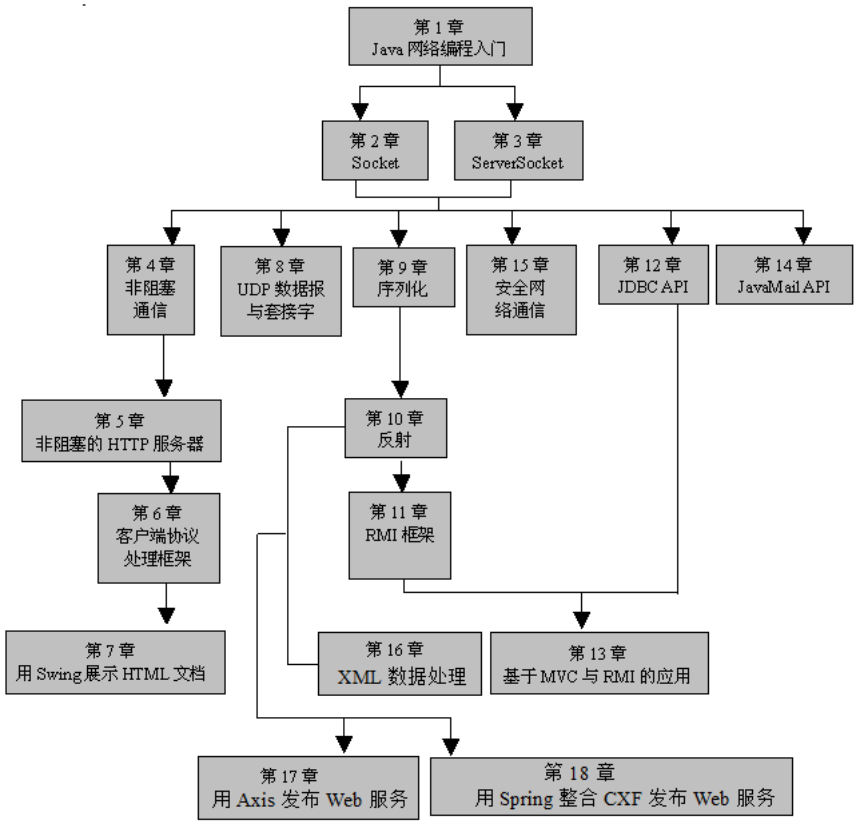

图 P-1　本书各章之间的顺序渐进关系

从图 P-1 可以看出，套接字（Socket）是 Java 网络编程的基础，第 2 章和第 3 章分别详细介绍了 Socket 与 ServerSocket 的用法。本书第 1 章介绍了分层的网络体系结构，Java 网络程序位于最上层——应用层，并且通过套节字访问底层网络，也可以说，套节字为应用层封装了底层网络传输数据的细节。Java 网络程序都采用客户/服务器模式，客户端发出获得特定服务的请求，服务器接收请求，执行客户端所请求的操作，然后向客户端发回响应。在介绍服务器端编程时，探讨了服务器端实现并发响应多个客户请求的两种方式：一种方式是运用线程池（第 3 章），还有一种方式是采用非阻塞通信（第 4 章）。在介绍客户端编程时，介绍了 JDK 提供的一种通用的客户端协议处理框架（第 6 章）。

利用 Java 网络 API，可以实现基于各种应用层协议（比如 HTTP 协议和 FTP 协议）的服务器程序与客户程序，本书侧重介绍了 HTTP 服务器（第 5 章）与 HTTP 客户程序（第 7 章）的实现方法，HTTP 客户程序也称为浏览器。

本书还介绍了两种分布式的软件架构：RMI（第 11 章）和 Web 服务（第 17 章和第 18 章）。这些分布式架构主要解决的问题是，如何让客户端调用服务器端的远程对象的方法。RMI 是 JDK 自带的，它要求客户端与服务器端都是 Java 程序，而 Web 服务允许用任意编程语言编写的客户程序与服务器程序能够通信。本书详细介绍了 RMI 框架的用法。RMI 框架在其实现中封装了用套接字通信的细节，此外，RMI 框架的实现会把客户端的方法调

用请求信息序列化为字节序列，把它发送给服务器端，然后在服务器端再通过反序列化把字节序列还原为方法调用请求。RMI 框架还运用了动态代理机制，为客户端提供了远程对象的代理。客户端实际上直接访问的是远程对象的代理。为了帮助读者理解 RMI 框架的实现原理，本书第 9 章和第 10 章分别介绍了 Java 序列化以及反射机制。在介绍反射机制时，介绍了动态代理。

本书第 17 章和第 18 章介绍了在开源软件框架 Axis、Spring 和 CXF 中创建和发布 Web 服务的方法。这些框架软件封装了客户端和服务器端底层通信的细节，使得开发人员只要利用框架软件的 API、注解和配置文件，就能方便地编写与具体业务领域相关的服务程序和客户程序。

本书还介绍了两个常用的客户端的网络 API：JDBC API（第 12 章）和 JavaMail API（第 14 章），这两个 API 分别用于访问数据库服务器和邮件服务器，在它们的实现中都封装了用套接字与服务器通信的细节。Java 客户程序可以通过 JDBC API 来访问各种数据服务器，还可以通过 JavaMail API 来访问各种邮件服务器。

本书第 13 章介绍了一个运用了 MVC 设计模式和 RMI 框架的综合应用。MVC 设计模式把实际的软件应用分为视图、控制器和模型三个层次，每个层次相对独立。本书的范例把模型作为远程对象放置到 RMI 的服务器端，把视图和控制器放置到 RMI 的客户端。

本书第 15 章介绍了 SSLSocket，它支持 SSL（Server Socket Layer）协议和 TLS（Transport Layer Security）协议。运用 SSLSocket，可以实现安全的网络通信，网络上传输的是被加密的数据，并且通信两端还能验证对方的身份。

本书在介绍以上技术时，采用 UML 建模语言中的类框图以及时序图来展示对象模型以及类与类之间的协作关系。此外，本书还把一些常见的设计模式，如静态代理模式、动态代理模式和 MVC 设计模式等运用到实际例子中。阅读本书，读者不仅可以掌握 Java 网络编程的实用技术，还可以进一步提高按照面向对象的思想来设计和编写 Java 软件的能力。

这本书是否适合您

阅读本书，要求读者已经具备了 Java 编程的基础知识。对于还不熟悉 Java 语言的读者，建议先阅读作者的另一本书《Java 面向对象编程》，本书是它的姊妹篇。《Java 面向对象编程》自 2006 年 7 月份出版后，一直畅销至今，受到了广大 IT 读者的欢迎。本书围绕着网络编程，进一步介绍了 Java 语言的一些高级特性，如套接字编程、非阻塞通信、数据报通信、对象的序列化与反序列化、Java 反射机制、RMI 框架、Web 服务、JDBC API 和 JavaMail API 等，这些特性是作为一个高级 Java 开发人员必须具备的。深入了解这些高级特性，有助于开发人员熟练地开发分布式的软件系统，或者轻松地学习和掌握现有的分布式软件架构。

本书一方面由浅入深组织内容，迎合 Java 网络编程初学者的需求，一方面与实际项目紧密结合，介绍了线程池、非阻塞通信和动态代理等高级话题，可作为 Java 开发人员的参考手册。本书还可以作为高校的 Java 教材，以及企业培训教材。

致谢

本书在编写过程中得到了 Oracle 公司在技术上的大力支持。此外 JavaThinker.net 网站的网友为本书的编写提供了有益的帮助，在此表示衷心的感谢！尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正，以下网址是作者为本书提供的技术支持网址，读者可通过它下载与本书相关的资源（如源代码、软件安装程序和视频课程、PPT 讲义等），还可以与其他读者交流学习心得，以及对本书提出宝贵意见：

http://www.javathinker.net/javanet.jsp