普通高等教育创新型人才培养规划教材

```
Connection con = null;
Statement stm = null;
ResultSet rs=null;
try {
    ...
} catch (Exception e) {
    ...
} finally {
        if(rs!=null){  rs.close();}
        if (stm != null) {  stm.close
        if (con != null) {  con.close
}
```

Java语言程序设计基础

JAVA YUYAN CHENGXU SHEJI JICHU

毕 静 主 编
王 岩 副主编

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

普通高等教育创新型人才培养规划教材

Java 语言程序设计基础

毕　静　主　编
王　岩　副主编

北京航空航天大学出版社

内 容 简 介

本书的主要内容是 Java 语言程序设计基础以及其中涉及的面向对象程序设计思想。Java 语言基础部分主要介绍编程语言基础和 Java 的一些语言特点。面向对象程序设计部分重点介绍面向对象的思想，相关概念和如何利用 Java 语言实现面向对象。然后介绍 Java 所特有的一些概念接口和包等。接下来是异常处理，图形用户界面，多线程编程，输入输出流和网络编程，涉及 Java 的具体编程功能应用。

本书可作为初学者的入门教材，也适于高等学校计算机科学及电子信息学科等专业本科学生学习使用。

图书在版编目(CIP)数据

Java 语言程序设计基础 / 毕静主编. -- 北京 : 北京航空航天大学出版社，2017.3

ISBN 978-7-81124-801-2

Ⅰ. ①J… Ⅱ. ①毕… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 020965 号

Java 语言程序设计基础

毕 静 主 编

王 岩 副主编

责任编辑 张艳学

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：goodtextbook@126.com 邮购电话：(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本：710×1 000 1/16 印张：14.5 字数：309 千字

2017 年 3 月第 1 版 2017 年 3 月第 1 次印刷 印数：2 000 册

ISBN 978-7-81124-801-2 定价：29.00 元

前　言

本书内容是基于作者近10年的“面向对象程序设计及Java”课程的教学经验，以及研究和开发经验编写的。Java语言是目前应用最广泛的面向对象程序设计语言之一，几乎渗透到当前各种行业的应用中。现在软件程序设计的主要方法采用面向对象的方法，很多开发工具开发环境都是面向对象的。所以面向对象的思想是计算机相关专业学生必须要了解的一个知识内容。计算机相关行业的很多用人单位都希望招收一些有Java基础的人员，以及在考研复试中一些高校也设置了Java相关的复试科目，可见Java的受欢迎程度。

本书注重理论与实践相结合，注重基本知识的理解与基本技能的培养，使读者具有基本的Java程序设计的能力。可以掌握Java类的定义，继承、封装和多态的主要思想，包括：类的创建、对象的创建、子类创建、方法的重载和方法的覆盖。可以实现Java语言图形界面设计，重点掌握图形组件和事件模型，可以编写简单的图形界面程序。可以实现Java语言的多线程编程，掌握用Runnable接口和Thread类定义多线程的方法，并可以实现简单的线程控制。掌握使用Socket通信机制实现客户与服务器的通信，编写基本的网络通信程序。

本书只是给大家学习Java提供一个入门素材，讲述的都是Java中最基础的部分，实际上Java包含很多内容，本书无法全部包括，只能为大家以后更深入学习Java的其他部分提供基础。本书的第1章和第5章主要由刘芳老师负责编写，第2章、第8章和第9章由王岩老师编写，第3章、第4章、第6章和第7章由毕静老师负责编写。在编写本书的过程中，许多同事付出了辛勤的劳动，一些研究生也给予了很大的帮助，在此一并感谢。本书中的相关程序例子，都提供了电子版的代码。

编　者

2016年10月

目　　录

第 1 章　Java 语言概述

Java 伴随着互联网的迅猛发展而发展，成了最流行的网络编程语言之一，其跨平台性标志着真正的分布式系统的到来。而且 Java 是免费使用的，这也使得它格外受欢迎。“Java 语言靠群体的力量而非公司的力量”是 Sun 公司的口号之一，这个观点也获得了广大软件开发商的认同。Java 目前非常流行，因此微软公司推出了与之竞争的 NET 平台以及模仿 Java 的 C# 语言。后来 Sun 公司被 Oracle(甲骨文)公司并购，Java 也随之成为甲骨文公司的产品。在 Oracle 网站 http://www.oracle.com/technetwork/java/index.html 上可以免费得到 JDK(Java Devolp kit)。

1.1　Java 概述

Java 是 Sun 公司推出的面向对象程序设计语言，特别适于 Internet 应用程序开发，已得到了业界的广泛认可。许多计算机产业的大公司购买了 Java 许可证，这些公司包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、TOSHIBA 以及 Microsoft。众多的软件开发商也开始支持 Java 软件产品。

1.1.1　Java 的发展

1995 年，Sun 推出了 Java 语言，并于 1996 年 1 月 23 日发布了 JDK 1.0。这个版本包括两部分：运行环境(JRE)和开发环境(JDK)。

JDK(Java Develop Kit)是 Sun 公司推出的 Java 开发工具包，它包括 Java 类库、Java 编译器、Java 解释器、Java 运行环境和 Java 命令行工具。JDK 本身仍使用较原始的命令行用户接口，其本身没有提供源程序编辑环境，也没有提供可视化的集成开发环境，而是由一些其他 Java 开发工具提供集成开发环境，如 Eclipse，Jcreator，Jbuilder 等。JRE 中包括核心 API、集成 API、用户界面 API、发布技术、Java 虚拟机(JVM)五个部分。

在 JDK 1.0 时代，JDK 除了 AWT(一种用于开发图形用户界面的 API)外，其他的库并不完整。Sun 在 1997 年 2 月 18 日发布了 JDK 1.1。JDK 1.1 相对于JDK 1.0 最大的改进就是为 JVM 增加了 JIT(即时编译)编译器。JIT 和传统的编译器不同，传统的编译器是编译一条，运行完后将其扔掉，而 JIT 会将经常用到的指令保存在内容中，在下次调用时就不需要再编译了。这使得 JDK 在效率上有了非常大的提升。

1998 年，JDK 1.0 已发展到 JDK 1.1.8。JDK 1.x 经过了 9 个小版本的发展，已

经初具规模。1998 年 12 月 4 日，Sun 发布了 Java 历史上最重要的一个 JDK 版本：JDK 1.2。JDK 1.2 与 JDK 1.0 有很大的区别，所以开始使用“Java 2”这一名称，即 JDK 1.2 以后的版本都称 Java 2。

J2SDK(Java 2 Software Develop Kit)扩展了许多新特性，同时废弃了原版本的许多方法。新特性重点是用新的方法构建程序，如使用类库或者使用应用程序接口。JDK 1.2 被分成了 J2EE、J2SE 和 J2ME 三大块，得到了强烈的市场反响。此外，JDK 1.2 还将它的 API 分成了三大类。Swing 是 Java 的另一个图形库，也是最引人注目的新特性。它不但有各式各样先进的组件，而且组件风格可抽换。Swing 并不是为了取代 AWT 而存在的，事实上 Swing 是建立在 AWT 之上的。另外 Java2 还在多线程、集合类和非同步类上做了大量的改进。在 Java2 时代 Sun 对 Java 进行了很多革命性的改进，而这些改进一直沿用到现在，对 Java 的发展产生了深远的影响。

从 JDK 1.2 开始，Sun 以平均两年一个版本的速度推出新的 JDK。在 2000 年 5 月 8 日，Sun 对 JDK 1.2 进行了重大升级，推出了 JDK 1.3。Sun 在 JDK 1.3 中同样进行了大量的改进，主要是在一些类库(如数学运算、新的 Timer API 等)和 JNDI 接口方面增加了 DNS 的支持和 JNI 的支持，这使得 Java 可以访问本地资源、支持 XML 以及用新的 Hotspot 虚拟机代替传统的虚拟机。

2002 年 2 月 13 日 Sun 发布了 JDK 历史上最为成熟的版本：JDK 1.4。这次 Sun 将主要精力放到了 Java 的性能上。在 JDK 1.4 中，对 Hotspot 虚拟机的锁机制进行了改进，使 JDK 1.4 的性能有了质的飞跃。同时 Compaq、Fujitsu、SAS、Symbian、IBM 等公司的参与，也使 JDK 1.4 成为发展最快的一个 JDK 版本。到 JDK 1.4 为止，已经可以使用 Java 实现大多数的应用了。

虽然从 JDK 1.4 开始，Java 的性能有了显著的提高，但 Java 又面临着另一个问题，那就是太复杂了。虽然 Java 是纯面向对象语言，但它对一些高级的语言特性并不支持。因此，在 2004 年 10 月，Sun 发布 JDK 1.5，同时将 JDK 1.5 改名为 J2SE 5.0。JDK 1.4 的主题是性能，而 J2SE 5.0 的主题是易用。Sun 之所以将版本号 1.5 改为 5.0，就是预示着 J2SE 5.0 较以前的 J2SE 版本有着很大的改进。Sun 不仅为 J2SE 5.0 增加了诸如泛型、增强的 for 语句、可变数目参数、注释、自动拆箱和装箱等功能，也更新企业级规范，如通过注释等新特性改善了 EJB 的复杂性，并推出了 EJB 3.0 规范。同时又针对 JSP 的前端界面设计而推出了 JSF。这个 JSF 类似于 ASP.NET 的服务端控件，通过它可以很快地建立起复杂的 JSP 界面。

J2SE 6.0 不仅在性能、易用性方面得到了前所未有的提高，而且还提供了如脚本、全新的 API(Swing 和 AWT 等 API 已经被更新)等支持。而且 J2SE 6.0 是专为 Vista 而设计的，它在 Vista 上将会拥有更好的性能。目前已经推出了 J2SE 7.0。

1.1.2 Java 技术体系

目前，Java 已经发展成为庞大的技术体系。这个技术体系主要有 3 个分支：Java

SE、Java EE 和 Java ME。

1. Java SE(Java Platform Standard Edition,Java 平台标准版)

这是适用于桌面系统的 Java 标准平台。Java SE 的程序运行在台式 PC 或便携式计算机上。Java SE 的实现主要包括:Java SE Development Kit(JDK)和 Java SE Runtime Environment(JRE)。本书要讲的程序主要是基于这个平台的。当然本书学习的是 Java 基础,对于其他平台的学习也有用。Java SE 是另外两个平台的基础,无论学习 Java EE 还是 Java ME,都必须要有较好的 Java SE 基础。Java SE 提供了编写与运行 Java Applet 与 Application 的编译器、开发工具、运行环境与 Java API 。

2. Java EE(Java Platform Enterprise Edition,Java 平台企业版)

Java EE 是 Sun 公司推出的企业级应用程序平台,能够开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序,其程序运行在工作站或服务器上。Java EE 是在 Java SE 的基础上构建的,可提供分布式企业软件组件架构的规范(组件模型)、Web 服务、管理和通信 API,可以用来实现企业级的面向服务体系结构(Service-Oriented Architecture,SOA)和 Web 2.0 应用程序,具有更高的性能、简化的集成性以及便捷性和 Java EE 服务器之间的互操作性。Java EE 包括 Enterprise JavaBeans(EJB),Java Servlets API 以及 Java Sever Pages(JSP)等技术,并为企业级应用的开发提供了各种服务和工具。例如,如果要做一个大型电子商务网站,就可以在服务器端编写 Java EE 程序。同样,Java EE 程序也运行在 JVM 中。

随着 Java 技术的发展,Java EE 平台得到了迅速的发展,成为 Java 语言中最活跃的体系之一。如今,Java EE 不仅仅是指一种标准平台,更多地表达着一种软件架构和设计思想。

3. Java ME(Java Platform MicroEdition,Java 平台微型版)

Java 语言的前身是 Oak 项目,原本就是为嵌入式领域设计的,但却没有顺利进入嵌入式领域,而是随着 Internet 的发展占领了 PC 端以及 Server 端,随后又回到了嵌入式领域。Java ME 是适用于小型设备和智能卡的 Java 嵌入式平台,包括智能卡、移动通信、电视机顶盒等。Java ME 还可以用来实现在手机上的游戏、照相、媒体播放功能等。Java ME 在移动设备上越来越流行,并开始与 Symbian、BREW 和 .NET Compact Framework 展开竞争。

Java ME 主要针对的设备是嵌入式和消费类的设备。这些设备受内存和处理器的限制,所以 Java ME 所包含的类库也比较小,相对 Java SE 的类库来说做了一些剪裁,虚拟机的功能也相对简单。与 Java SE 和 Java EE 相比,Java ME 总体的运行环境和目标更加多样化,但其中每一种产品的用途却更为单一,资源限制也更加严格。

1.1.3　Java 语言特点

Sun 公司对 Java 编程语言的定义是:Java 编程语言是简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。

1. 跨平台性

跨平台性是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。这是软件发展的趋势和编程人员追求的目标。计算机硬件的种类繁多，操作系统也各不相同，不同的用户和公司有自己不同的计算机环境偏好，而软件为了能在这些不同的环境里正常运行，就需要独立于这些平台。

在 Java 语言中，Java 自带的虚拟机很好地实现了跨平台性。Java 源程序代码经过编译后生成二进制的字节码是与平台无关、但可被 Java 虚拟机识别的一种机器码指令。Java 虚拟机提供了一个字节码到底层硬件平台及操作系统的屏障，使得 Java 语言具备跨平台性。

2. 面向对象

面向对象是指以对象为基本粒度，其下包含属性和方法。对象的说明用属性表达，而通过使用方法来操作这个对象。面向对象技术使得应用程序的开发变得简单易用，节省代码。Java 是一种面向对象的语言，继承了面向对象的诸多优点，如代码扩展、代码复用等。

3. 安全性

安全性可以分为四个层面，即语言级安全性、编译时安全性、运行时安全性、可执行代码安全性。

语言级安全性指 Java 的数据结构是完整的对象，这些封装过的数据类型具有安全性。编译时要进行 Java 语言和语义的检查，保证每个变量对应一个相应的值，编译后生成 Java 类。运行时 Java 类需要类加载器载入，并经由字节码校验器校验之后才可以运行。Java 类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。

4. 多线程

多线程在操作系统中已得到了最成功的应用。多线程是指允许一个应用程序同时存在两个或两个以上的线程，用于支持事务并发和多任务处理。Java 除了内置的多线程技术之外，还定义了一些类、方法等来建立和管理用户定义的多线程。

5. 简单易用

Java 源代码的书写不拘泥于特定的环境，可以用记事本、文本编辑器等编辑软件来实现，然后将源文件进行编译；编译通过后可直接运行，通过调试则可得到想要的结果。

Java 语言省略了 C++语言中所有难以理解、容易混淆的特性，例如头文件、指针、内存管理、运算符重载、虚拟基础类等。它更加严谨、简洁。

6. 解释执行

Java 语言是一种解释型语言，在运行 Java 时，需要先将 Java 源程序编译成字节码，然后再利用解释器将字节码解释成本地系统的机器指令。由于字节码与环境无关，且类似于机器指令，因此，在不同的环境下，不需要重新对 Java 源程序进行编译，

直接利用解释器进行解释执行即可。当然，随着 Java 编译器和解释器的不断改进，运行效率也在不断提高。

7. 高性能

由于 Java 是一种解释型语言，所以执行效率相对会低一些，但 Java 语言采用了两种手段，这使得其性能还是不错的。

① Java 语言源程序编写完成后，先使用 Java 伪编译器进行伪编译，将其转换为中间码（也称为字节码），再解释；

② 提供了一种“准实时”（Just-in-Time，JIT）编译器，当需要更快的速度时，可以使用 JIT 编译器将字节码转换成机器码，然后将其缓冲下来，这样速度就会更快。

1.2　JDK 的安装及 Java 应用程序

下面就进入 Java 语言程序设计的第一个步骤。首先从相关网站下载合适的 JDK，然后安装 Java 编程及运行环境，并编写第一个简单的 Java 应用程序。

1.2.1　JDK 的安装及环境变量的配置

开发 Java 程序首先要配置好环境变量，而 Java 运行环境的配置比较麻烦。下面介绍 JDK 的安装过程，这里选用的 JDK 是 jdk-6u33-windows-x64 版本。

安装分为两个步骤：

① 首先要准备好 JDK 的安装文件 jdk-6u33-windows-x64。

② 配置环境变量 path 和 classpath。

下面简单介绍安装过程。如图 1.1 和图 1.2 所示，启动安装程序，设置安装路径。

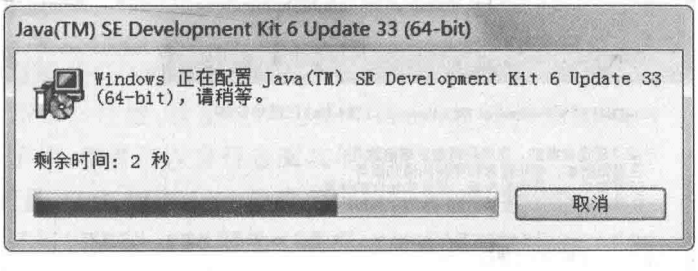


图 1.1　启动 JDK 安装程序

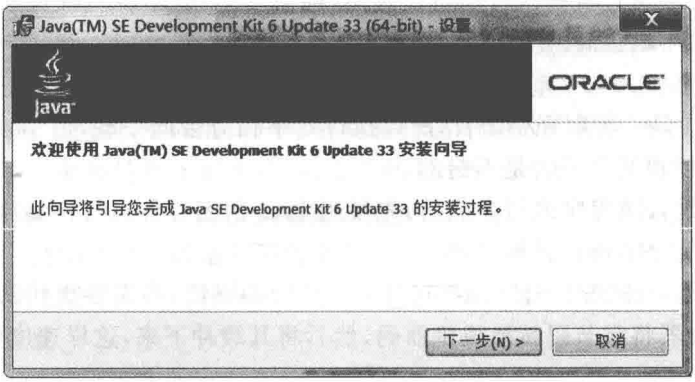
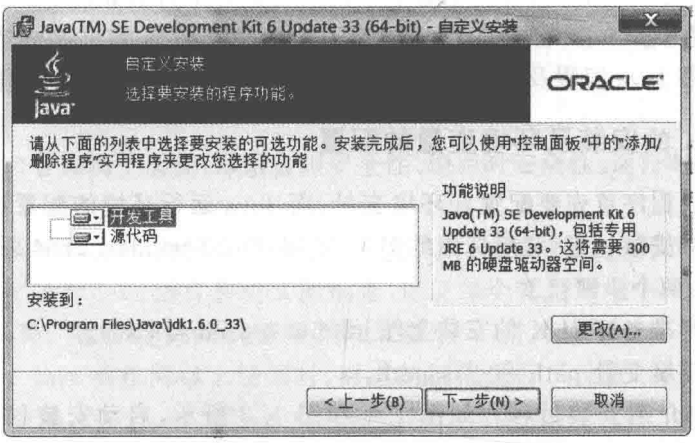


图 1.1　启动 JDK 安装程序(续)



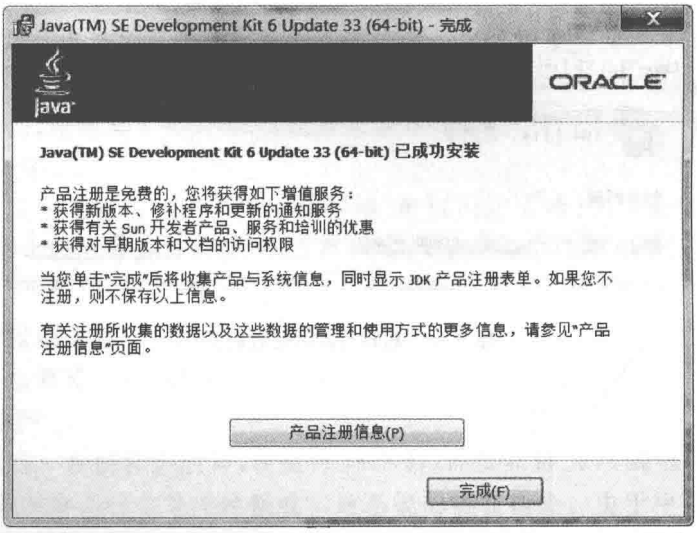


图 1.2　安装路径设置为 C:\Program Files\Java\jdk1.6.0_33\

之后，默认安装设置即可。在编译 Java 程序时需要用到 javac 命令，执行 Java 程序需要用到 Java 命令，而这两个命令并不是 Windows 自带的命令，所以需要配置好环境变量，这样就可以在任何目录下使用这两个命令了。设置环境变量的方法：在“我的电脑”上单击右键，选择属性→高级→环境变量→path，如图 1.3 所示。

图 1.3　系统环境变量 path 的配置

在 path 后面加上 C:\Program Files\Java\jdk1.6.0_33\bin；这是安装 JDK 的路径，如图 1.4 所示。注意：这里使用的是 Windows7 操作系统，至于其他的操作系统，配置会有所不同，读者可以自行查阅其他的资料。

设置环境变量：path 指出 Java 提供的可执行文件的路径；classpath 指出 Java 包的路径。此外，还可以安装 JDK 帮助文档。这样就可以在任何目录下使用 javac 和 java 这两个命令，用户编程时还可以调用 Java 类库中的资源。

JDK 安装之后在安装目录下可以看到以下一些文件夹。

① src.zip：核心 API 所有类的源文件；

② bin：包含编译器、解释器等可执行文件；

③ demo：包含源代码的程序示例；

④ include：编写 JNDI 等程序需要的 C 语言头文件；

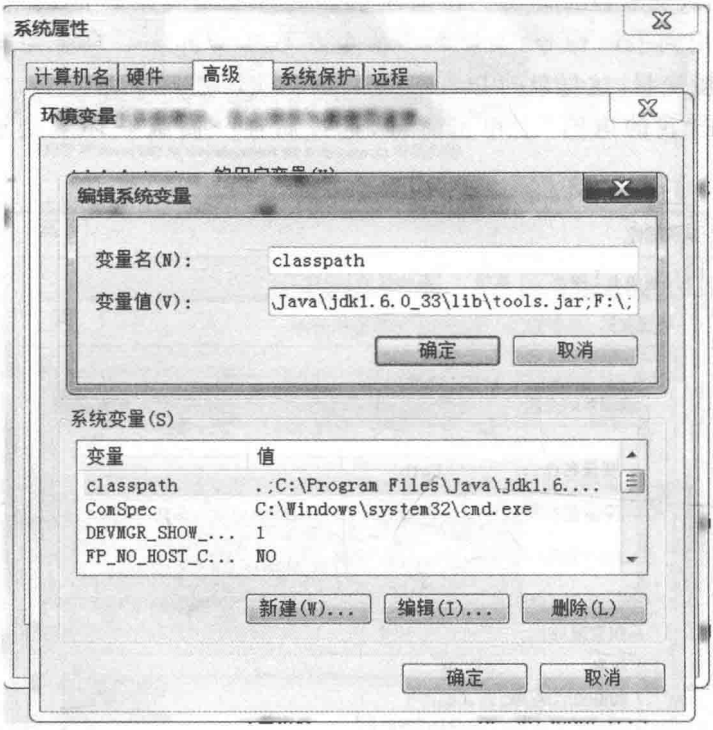


图 1.4　添加环境变量 classpath

⑤ jre:Java 运行时环境;

⑥ lib:Java 类库。

1.2.2　Java 应用程序

安装好 Java 环境后,就可以编写、编译、运行 Java 应用程序了。Java 源程序文件(*.java)通过编译器(javac.exe)编译生成字节码文件(*.class),通过 Java 虚拟机中的 Java 解释器(java.exe)来解释执行其字节码文件。下面介绍一个最为简单的 Java 应用程序的编写、编译和运行过程。

编写源文件:C 语言源文件由若干个函数组成,Java 源文件由若干个类组成。对于没有安装第三方编辑环境的用户,可以通过记事本编写代码,然后修改记事本文件名并把扩展名修改为.java。

【例 1.1】显示字符串的 Application 应用程序。此文件的文件名必须为 Hello.java。

```
public class Hello
{
    public static void main(String args[])
    {
```