第一部分 认识 Java

面向对象程序设计概述

学习目标

- 理解面向过程和面向对象程序设计思想的区别
- 掌握 Java 语言的基本特点
- 熟悉 Java 语言的各种开发平台

重点难点

- 面向对象程序设计思想的基本概念
- Java 语言的基本特点
- Java 语言开发平台的适用场合

1.1 程序设计思想的发展

1.1.1 面向过程的程序设计思想及语言

"面向过程"是一种以事件为中心的编程思想,即分析出解决问题所需要的步骤,然后用函数把这些步骤一步一步实现,使用的时候一个一个依次调用。面向过程是一种基础的方法,是自上向下步步求精,将一个复杂任务按照功能进行拆分,并逐层细化到便于理解和描述的程度,最终形成由若干独立模块组成的树状结构,所以面向过程最重要的是模块化的设计思想。当程序规模不是很大时,面向过程的方法还会体现出一种优势,因为程序的流程很清楚,按照模块对应函数的方法可以很好地组织。例如一个学生早上起来通常要做的事情可以大致概括为

最初问世的高级语言都属于面向过程的编程语言,用来实现程序的顺序、选择、循环这三种结构,比较著名的语言有 Pascal、Modula、Bliss、Euclid、Edison、Gypsy、CLU、XCY-2、Smalltalk、C等。C语言作为最具代表性的面向过程程序设计语言于 1973 年被研制出来。C语言的成功得益于它有一个好的机会,那时的高级语言基本都不适合开发系统软件,系统软件基本都是用机器语言或汇编语言编写的,而 C 的许多类汇编语言特征却使它大获成功。C语言的表达式比较简洁,具有丰富的运算符,有比较现代的控制结构与数据结构,它不是一种大语言,也不是一种很高级的语言,后来其应用范围已不限于系统软件开发,它逐渐成为最流行的语言之一。C语言的主要不足在于它并没有完全体现出好的程序设计思想,由于过于注重开发与运行效率而使其程序的可读性较差。

"面向过程"程序设计方法也有它的缺陷,主要表现在以下方面:

- (1) 难以适应大型软件的设计:在大型多文件软件系统中,随着数据量的增大,由于数据与数据处理相对独立,程序变得越来越难以理解,文件之间的数据沟通也变得困难。
- (2)程序可重用性差:"面向过程"程序设计方法不具备建立"软件部件"的工具,即使是面对老问题,数据类型的变化或处理方法的改变也必将导致重新设计。这种额外开销与可重用性相矛盾,称为"重复投入"。

这些由结构化程序设计的特点所导致的缺陷,其本身无法克服,而越来越多的大型程序设计又要求必须克服它们,这最终导致了"面向对象"设计方法的产生。

1.1.2 面向对象的程序设计思想及语言

"面向对象"程序设计是一种解决问题时整体看待一个实体的思想。它是从现实世界中客观存在的事物出发来构造软件系统,强调直接以现实世界中的事物为中心来思考问题、认识问题,并根据这些事物的本质特点,把它们抽象地表示为系统中的对象,作为系统的基本构成单位。简单地来说,就是将我们编程时独立考虑的相关的一些属性和一些操作都"打个包",整体来对待,也就是将我们现实世界的事物看成属性(数据)和它本身的操作(方法)来整体对待。把数据和方法整合到一起就形成了类,类声明得到对象,编程时有效地利用类的继承性,会在很大程度上提高编程效率。如果使用面向对象的方法来处理上面提到的"学生"实例,可能就只需抽象出一个学生的类,它包括四个方法,分别是起床、穿衣、洗漱、去学校,但是具体的顺序就不能体现出来了。

与面向过程程序设计相比,面向对象程序设计有以下优点:

- (1)数据抽象的概念可以在保持外部接口不变的情况下改变内部实现,从而减少甚至避免对外界的干扰。
- (2)通过继承可大幅减少冗余的代码,并可以方便地扩展现有代码,提高编码效率,也减少了出错概率,降低了软件维护的难度。

Chapte

- (3)结合面向对象分析、设计,允许将问题中的对象直接映射到程序中,减少了软件开发过程中中间环节的转换过程。
- (4)通过对对象的辨别、划分可以将软件系统分割为若干相对为独立的部分,在一定程度上便于控制软件复杂度。
- (5)通过对象的聚合、联合,可以在保证封装与抽象的原则下实现对象的内在结构以及外在功能的扩充,从而实现对象由低到高的升级。

面向对象程序设计语言与传统的面向过程编程语言的主要区别在于:传统过程性语言中把数据以及处理它们的子程序当作互不相关的成分分别处理,而面向对象语言则把这两者统一作为对象封装一起进行处理。面向对象的思想是 G. Booch 在研究 Ada 软件开发方法时首先提出来的,它的一个重要概念就是类。完全面向对象的语言有 Smalltalk、C++、Eiffel、Java等。C++是在 C 语言的基础上扩充而成的,是 C 语言的超集。它在 C 语言的基础上扩充了类、对象、多继承、运算符重载等面向对象的概念。Eiffel 是一种以可再用性与可扩充性作为其主要目标的对象式语言,于 20 世纪 80 年代中期问世。Eiffel 是一种强类型语言,其主要成分是用于表示抽象数据类型的类。随着各种范型的程序设计方法的研究,各种范型相互渗透,目前许多语言都不止体现了一种范型,如 Modula-2、Ada 等强制式语言中都体现了面向对象式范型的思想,而 C++、Eiffle 等对象式语言中则充满了强制式语言的思想。在 20 世纪 80 年代中期问世的还有一种有名的多范型语言 Nial,它支持强制式与函数式范型,而新扩充的 Nial 语言还可以支持逻辑式与对象式范型,把各种范型的思想都尽可能融合在一种语言中,以支持不同要求、不同风格的程序设计应用。

1.2 Java 程序设计语言

1.2.1 Java 语言概述

1991年,Sun 公司为了进军家用电子消费市场,成立了一个代号为 Green 的项目组, Oak 系统出现。Oak 以 C++语言为蓝本,吸收了 C++中符合面向对象程序设计要求的部分,同时加入了一些满足网络设计要求的部分。1994年,Green 项目组成员认真分析计算机网络应用的特点,认为 Oak 满足网络应用所要求的平台独立性、系统可靠性和安全性等。1995年 5 月 23 日,Sun 公司正式发布了 Java 产品。

Java 语言一经推出, 就受到了业界的关注。Netscape 公司第一个认可 Java 语言, 并于 1995 年 8 月将 Java 解释器集成到它的主打产品 Navigator 浏览器中。接着, Microsoft 公司在 Internet Explorer 浏览器中认可了 Java 语言。Java 语言开始了自己的发展历程。

Java 语言是简单的、面向对象的语言,它具有分布式特点、安全性,又可以实现多线程, 更主要的是它与平台无关,解决了困扰软件界多年的软件移植问题。 面向对象(object-oriented)程序设计模式是近代软件工业的一种革新,它提供软件的弹性度(flexibility)、模块化(modularity)与重复使用率(reusability),降低开发时间与成本。 Java 是完全对象化的程序语言,编程重点在于产生对象、操作对象以及如何使对象能一起协调工作,以实现程序的功能。

(2) 语法简单。

Java 语言的语法结构类似于 C 语言和 C++, 熟悉 C++的程序设计人员不会对它感到陌生。 与 C++相比, Java 对复杂特性的省略和实用功能的增加使得开发变得简单而可靠。

(3) 平台无关性。

平台无关性是指 Java 能运行于不同的系统平台。Java 引进虚拟机概念,Java 虚拟机(Java Virtual Machine,JVM)建立在硬件和操作系统之上,用于实现对 Java 字节码文件的解释和执行,为不同平台提供统一的 Java 接口。这使得 Java 应用程序可以跨平台运行,非常适合网络应用。

(4) 安全性。

安全性是网络应用系统必须考虑的重要问题。Java 设计的目的是提供一个网络/分布式的计算环境,因此,Java 特别强调安全性。Java 程序运行之前会利用字节确认器进行代码的安全检查,确保程序不会存在非法访问本地资源、文件系统的可能,保证了程序在网络间传送运行的安全性。

(5) 分布式应用。

Java 为程序开发提供了 Java.net 包,该包提供了一组类,使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外, Java 还提供了专门针对互联网应用的一整套类库,供开发人员进行网络程序设计。

(6) 多线程。

Java 语言内置了多线程控制,可使用户程序并行执行。利用 Java 的多线程编程接口,开发人员可以方便地写出多线程的应用程序。Java 语言提供的同步机制可保证各线程对共享数据的正确操作。在硬件条件允许的情况下,这些线程可以直接分布到各个 CPU 上,充分发挥硬件性能,提高程序执行效率。

1.2 2 Java 语言的三大开发体系

Java 发展到今天,已从编程语言发展成为全球第一大通用开发平台。Java 技术已被计算机行业主要公司所采纳。1999年,Sun 公司推出了以 Java 2 平台为核心的 J2SE、J2EE 和 J2ME 三大平台。随着三大平台的迅速推进,全球形成了一股巨大的 Java 应用浪潮。

1. Java 2 Platform, Standard Edition (J2SE)

Java 2 平台标准版,适用于桌面系统应用程序的开发,以及低端的服务器。本书例程就是利用 J2SE 5.0 版的相关图形 API 包来开发的。J2SE 包含那些构成 Java 语言核心的类,比如,

Chapte

数据库连接、接口定义、输入输出、网络编程。

2. Java 2 Platform, Enterprise Edition (J2EE)

Java 2 平台企业版,是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理等相关复杂问题的体系结构。J2EE 技术的核心就是 Java 平台或 Java 2 平台的标准版,J2EE 技术主要用于分布式的网路程序的开发,构建企业级的服务器应用,如电子商务网站和 ERP 系统。J2EE 包含 J2SE 的类,并且还包含用于开发企业级应用的类。J2EE 包含了 EJB,Servlet,JSP,XML,JDBC,CORBA,以及能够在 Internet 应用中保护数据的安全模式等技术。

它包含标准版中的任何组件,并增加了附加库,以支持目录管理、交易管理以及企业消息处理等功能。

3. Java 2 Platform, Micro Edition (J2ME)

对于 Java 2 平台微型版,Sun 公司将 J2ME 定义为"一种以广泛的消费性产品为目标、高度优化的 Java 运行环境"。自 1999 年 6 月在 JavaOne Developer Conference 上声明之后,J2ME 进入了小型设备开发的行列。通过 Java 的特性,遵循 J2ME 规范开发的 Java 程序可以运行在各种不同的小型设备上,用于嵌入式的消费型产品中,如寻呼机、手机、掌上电脑、智能卡、机顶盒、汽车导航系统或其他无线设备。

简单地说,J2SE 用于小型程序,J2EE 用于大型程序,J2ME 用于手机等程序。J2SE 包含于 J2EE 中,J2ME 包含了 J2SE 的核心类,但新添加了一些专用类。语言都是相同的,只是捆绑的类库 API 不同。也就是说,J2SE 是基础;压缩一点,再增加一些 CLDC 等方面的特性就是 J2ME;扩充一点,再增加一些 EJB 等企业应用方面的特性就是 J2EE。

1.3 思考及习题

- 1. 简述面向过程与面向对象程序设计的区别。
- 2. Java 语言的特点有哪些?
- 3. Java 语言包括哪三种平台? 分别适用于什么场合?

hapte

初识 Java 程序设计

学习目标

- 了解 Java 开发环境
- 掌握 Java 语言的运行机制
- 了解 Java 虚拟机

重点难点

- Java 环境变量的配置
- Eclipse 的使用

2.1 典型案例

【案例描述】

使用 Eclipse 编写程序,在控制台输出多行信息,运行结果如图 2-1-1 所示。

【运行结果】

2.2 Java 开发环境的搭建

2.2.1 Java 虚拟机的简介

Java 语言具有一个优点,就是平台无关性,即 Java 编写的程序可以在 Windows、Linux、Mac OS 等操作系统中运行。正是 Java 虚拟机(Java Virtual Machine,JVM)为 Java 语言实现了平台无关性。

使用 Java 语言编写的程序需要经过 Java 编译器编译,这时会生成与平台无关的字节码 (*.class 文件)。这种字节码是不可执行的,必须使用 JVM 来解释执行。因此,我们无论在任何系统上编写 Java 程序,只要该平台安装了 JVM,便可以运行。其中不同的操作系统需要对应的 JVM,运行过程如图 2-2-1 所示。

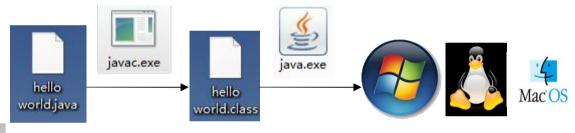


图 2-2-1

2.2.2 JDK 的下载与安装

JDK (Java Development Kit),即 Java 标准开发工具箱,是 Oracle 公司提供一套用于开发 Java 应用程序的开发工具,提供了编译、运行 Java 程序所需要的各种工具和资源,包括 Java 编译器、Java 运行环境,以及常用 Java 类库等。

Java 运行环境也称为 JRE (Java Runtime Environment), 是运行 Java 程序的必要条件。可以说 JRE 里包括 JVM。

注意: 当需要开发 Java 程序,则选择安装 JDK,安装 JDK 之后,实际也就包含了 JRE,就可以执行 Java 程序。如果只是运行 Java 程序,则安装 JRE 即可。实际上,Oracle 网站上提供 JDK、JRE 的下载,而不提供 JVM 的下载。

1. JDK 的下载

JDK 官方下载网址为: http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u75-oth-JPR, 不同平台的 JDK 如图 2-2-2 所示。

You must accept the Oracle Binary Co	ode License Agre software.	ement for Java SE to download this
Accept License Agreement Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	119.43 MB	jdk-7u75-linux-i586.rpm
Linux x86	136.77 MB	jdk-7u75-linux-i586.tar.gz
Linux x64	120.83 MB	jdk-7u75-linux-x64.rpm
Linux x64	135.66 MB	jdk-7u75-linux-x64.tar.gz
Mac OS X x64	185.86 MB	jdk-7u75-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.55 MB	jdk-7u75-solaris-i586.tar.Z
Solaris x86	95.87 MB	jdk-7u75-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.66 MB	jdk-7u75-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u75-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	138.66 MB	jdk-7u75-solaris-sparc.tar.Z
Solaris SPARC	98.56 MB	jdk-7u75-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u75-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.37 MB	jdk-7u75-solaris-sparcv9.tar.gz
Windows x86	127.8 MB	jdk-7u75-windows-i586.exe
Windows x64	129.52 MB	idk-7u75-windows-x64.exe

图 2-2-2

选择对应平台的 JDK 进行下载。

2. JDK 的安装

(1) 双击下载的 JDK 可执行文件开始安装, JDK 安装的初始界面如图 2-2-3 所示。



图 2-2-3

(2) 单击"下一步"按钮,可以通过"更改"按钮选择 JRE 的安装路径,如图 2-2-4 所示。



图 2-2-4

(3) 单击"下一步"按钮,出现提示安装进度界面,如图 2-2-5 所示,可以通过"更改"按钮选择 JRE 的安装路径,如图 2-2-6 所示。



图 2-2-5

2 Chapter



图 2-2-6

(4) 单击"下一步"按钮进入JRE 安装进度界面,如图 2-2-7 所示。



图 2-2-7

(5) 安装成功后如图 2-2-8 所示。



图 2-2-8

2.2.3 配置系统环境变量

(1) JDK 安装完成需要进行环境变量的配置。右击"计算机",选择"属性",出现如图 2-2-9 所示页面。单击"高级系统设置",选择"高级",如图 2-2-10 所示。

系统属性

用户配置文件

启动和故障恢复

与您登录有关的桌面设置

系统启动、系统失败和调试信息

计算机名 硬件 高級 系统保护 远程 要进行大多数更改,您必须作为管理员登录。

视觉效果,处理器计划,内存使用,以及虚拟内存

23

设置(S)...

设置(E)...

设置(T)...

环境变量 (N)...





(2) 单击 "环境变量" 按钮,设置 PATH,单击"新建" 按钮,如图 2-2-11、图 2-2-12 所示。

2 Chapter

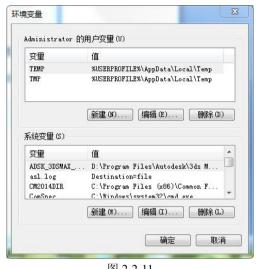




图 2-2-11

图 2-2-12

(3) 在"变量名"文本框中输入"PATH", 在"变量值"文本框中输入"C:\Program Files\Java\jdk1.7.0 75\bin"。

补充 1: 用户变量与系统变量的区别,用户变量部分表示用于设置当前用户的环境变量,而不需要影响其他的登录用户,而系统变量部分用于设置整个系统的环境变量。因此,可以根据自己的要求选择在哪个部分添加环境变量。

补充 2: "C:\Program Files\Java\jdk1.7.0_75\bin"其实就是 Java 文件中 bin 文件夹的安装路 径,只需要找到它复制粘贴即可,如图 2-2-13 所示。



(4) 单击"环境变量"按钮,设置 CLASSPATH,单击"新建"按钮,如图 2-2-14 所示。



图 2-2-14

2 Chaptei

- (5) 在"变量名"文本框中输入"CLASSPATH", 在"变量值"文本框中输入".;C:\Program Files\Java\jdk1.7.0_75\lib\dt.jar;C:\Program Files\Java\jdk1.7.0_75\lib\tools.jar"。
- 补充 1: Java 版本如果是 1.5 以上的只需要配置 "PATH"环境变量即可,如果是 Java EE 仍需要配置 CLASSPATH。
- 补充 2: "变量值"为 Java 的 lib 文件夹中 tools.jar 与 dt.jar 的路径,但是千万注意必须在变量值的最前面加上".;",不然配置不会成功。它表示当前目录,用于识别当前目录的 Java 类。tools.jar 与 dt.jar 两个路径之间添加";"。
 - (6) 单击 "开始" → "运行", 输入 "cmd", 如图 2-2-15 所示。

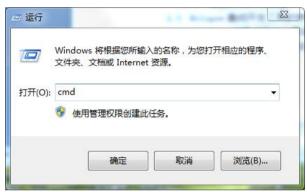


图 2-2-15

(7) 输入 "Java",如图 2-2-16 所示,输入 "Javac",如图 2-2-17 所示,则环境变量配置成功。

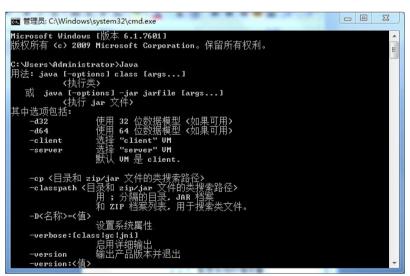


图 2-2-16

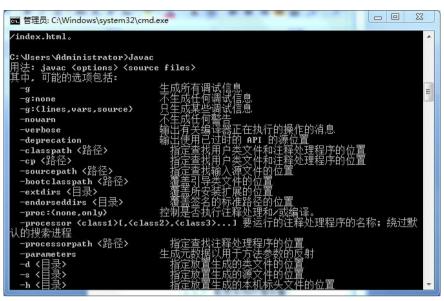


图 2-2-17

2.3 Eclipse 集成开发工具的使用

2.3.1 Java 的集成开发工具简介

对于初学者来说,使用 JDK 开发 Java 程序可以很快理解程序中代码之间的关系,有利于理解 Java 的设计思想,在学习中可以不使用任何 IDE(集成开发环境)工具来学习编程,Windows 平台可以使用记事本,Linux 可以使用 VI 等工具。但是,倘若进行大规模的开发时,便会显得捉襟见肘,因此,Java 的集成开发工具是我们未来进行 Java 学习中必不可少的工具。其中包括: Eclipse、NetBeans、JBuilder、JDeveloper 等 IDE 工具。如果想知道详细内容可以通过网络等途径查找。

本教材选取 Eclipse 作为 Java 开发工具。

2.3.2 Eclipse 的下载与安装

- (1) Eclipse 下载地址为http://www.eclipse.org/downloads/,选择符合系统版本的 Eclipse 下载。
- (2) 下载后解压下载的文件。如图 2-3-1 所示,双击 eclipse.exe 即可进行 Java 语言的开发。

2 Chapter

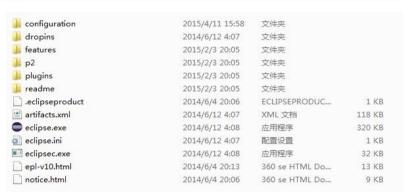


图 2-3-1

2.3.3 Eclipse 的基本使用

(1) 双击 eclipse.exe,如图 2-3-2 所示。



图 2-3-2

(2) 设置 Eclipse 工作空间路径,如图 2-3-3 所示。

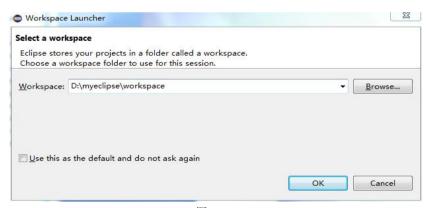


图 2-3-3

- (3) 单击 "Browse" 选择路径后,单击 "OK" 按钮,进入系统工作界面,如图 2-3-4 所示。
- (4) 单击 "File" → "New" → "Java Project", 在工作区创建新项目, 如图 2-3-5 所示。

2 Chapter

图 2-3-4

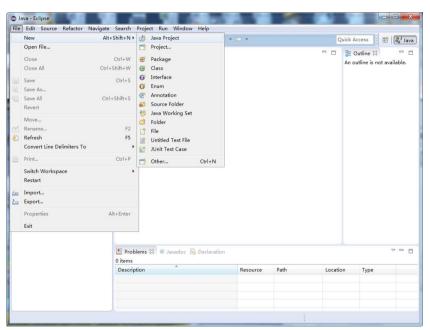


图 2-3-5

补充: 创建新项目的方法有若干种。也可单击 "File" → "New" → "Project",选择 "Java Project",如图 2-3-6 所示。

Z Chapte

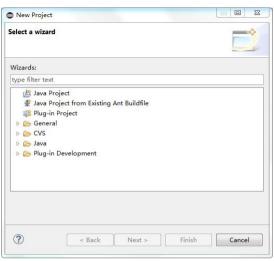


图 2-3-6

(5) 在 "Project name" 文本框中输入项目名称,其他保持默认状态,然后单击 "Finish" 按钮,完成项目创建,如图 2-3-7 所示。



图 2-3-7

(6) 单击 "File" → "New" → "Package", 如图 2-3-8 所示, 输入包名称, 单击 "Finish" 按钮, 创建包, 如图 2-3-9 所示。

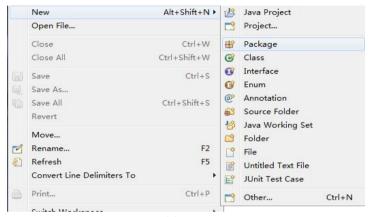


图 2-3-8

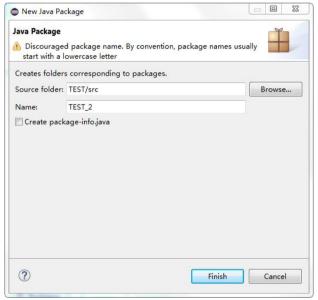


图 2-3-9

补充: 当创建项目后,在开发项目的过程中,系统会把所有的源文件(.java)放到 src 文件夹的默认包中。随着创建的文件数越来越多,管理效率会下降。添加包可以理解为添加子文件夹,用于分类管理,提高效率。

(7) 单击 "File" → "New" → "Class", 在"Name" 文本框中输入新的类名, 单击 "Finsih" 按钮, 创建 Java 文件, 如图 2-3-10 所示。

图 2-3-10

补充 1: Package: 表示存放源文件的包名。

补充 2: ☑ public static void main(String[] args) 选中表示该类可以单独运行。

(8) 双击生成的*.java 文件,在工作界面左侧编写代码,如图 2-3-11 所示。

图 2-3-11

. Chapter 补充:双击*.java文件后,可以看到初始的部分代码,这是系统自动生成的模板代码。

(9) 代码编写完毕后,单击 "File"→ "New"→ "Save"保存,保存后单击运行按钮 · , 执行结果在程序运行控制台(Console)显示,如图 2-3-12 所示。



图 2-3-12

2.4 上机实践

(1) 在 DOS 命令行编译和运行 Java 程序,输出结果为"Hello World!!!"。

分析:新建记事本,在记事本中编写 Java 程序代码,另存到 D 盘,文件名为 Helloworld.java。在"命令提示符"窗口输入"d:",按 Enter 键,如图 2-4-1 所示进入 D 盘;输入 Javac Helloworld.java 命令编译程序,此时 D 盘出现名为 Helloworld.class 的文件,输入 Java Helloworld 命令运行程序,出现结果,如图 2-4-2 所示。



图 2-4-1

源代码:

```
public class Helloworld {
    public static void main(String[] args) {
        System.out.println("Hello World!!!");
    }
}
```

运行结果:

D:∖>Java Helloworld Hello World!!!

图 2-4-2

(2) 使用 Eclipse 编写程序,分两行输出自己的姓名和年龄(提示: 使用两条输出语句,分别输出姓名和年龄)。

分析: 启动 Eclipse, 依次新建项目及类; 使用输出语句输出姓名和年龄, 结果如图 2-4-3 所示。

源代码:

```
public class MyMessage {
    public static void main(String[] args) {
        System.out.println("我的姓名: 李华");
        System.out.println("我的年龄: 19岁");
    }
}
```

运行结果:

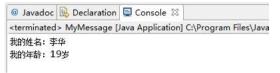


图 2-4-3

2.5 思考及习题

一、简答题

- 1. 简述如何配置 Java 的开发环境,并在自己的计算机上安装配置 Java。
- 2. 如何下载 Eclipse,并进行安装?

二、编程题

1. 编写 Java 程序,输出结果如下所示。



2. 编写程序, 实现界面如图 2-5-1 所示。

图 2-5-1

2 Chapte