项目 1

Java 项目创建与运行

1.1 任务 1：搭建 Java 程序的开发环境

技能目标

(1) 熟悉 JDK(Java Development Kit)的下载与安装过程。

(2) 熟练配置 JDK 环境变量并进行测试。

(3) 熟悉 Eclipse 集成开发环境的下载与汉化过程。

知识目标

(1) 了解 JDK 的概念与功能。

(2) 了解 JRE 的概念与功能。

(3) 了解 Eclipse 集成开发环境的概念与功能。

工作任务

(1) 在 Oracle 公司官方网站下载 JDK。

(2) 在 Windows 系统下安装 JDK。

(3) 配置 JDK 环境变量。

(4) 使用命令行工具测试 JDK。

(5) 在 Eclipse 官方网站下载 Eclipse 集成开发环境。

(6) 下载汉化包或在线汉化 Eclipse。

1.1.1 JDK 下载与安装

JDK 是 Oracle 公司针对 Java 开发人员发布的免费软件开发工具包，它是整个 Java 的核心。自从 Java 推出以来，JDK 已经成为使用最广泛的开发工具包之一。该工具包含 Java 语言的编译工具、运行工具以及执行程序的环境(Java Runtime Environment，JRE)。普通用户并不需要安装 JDK 来运行 Java 程序，而只须安装 JRE。而程序开发者必须安装 JDK 来完成程序的编译和调试。

1. JDK 的下载

要获得 JDK 最新版本，可以到 Oracle 公司的官方网站上进行下载，下载地址为 http://www.oracle.com/technetwork/java/index.html。

在笔者截稿时最新的下载版本是 Java SE 6 Update 25。JDK 支持目前各种流行的操作系统。由于 Windows 操作系统有 32 位及 64 位两个版本，下载时单击对应版本的文件链接下载，如图 1.1 所示。

Java SE Development Kit 6 Update 25		
Product / File Description	File Size	Download
Linux x86 - RPM Installer	76.85 MB	jdk-6u25-linux-i586-rpm.bin
Linux x86 - Self Extracting Installer	81.11 MB	jdk-6u25-linux-i586.bin
Linux x64 - RPM Installer	77.06 MB	jdk-6u25-linux-x64-rpm.bin
Linux x64 - Self Extracting Installer	81.36 MB	jdk-6u25-linux-x64.bin
Solaris x86 - Self Extracting Binary	81.00 MB	jdk-6u25-solaris-i586.sh
Solaris x86 - Packages - tar.Z	136.67 MB	jdk-6u25-solaris-i586.tar.Z
Solaris SPARC - Self Extracting Binary	85.96 MB	jdk-6u25-solaris-sparc.sh
Solaris SPARC - Packages - tar.Z	141.11 MB	jdk-6u25-solaris-sparc.tar.Z
Solaris SPARC 64-bit - Self Extracting Binary	12.24 MB	jdk-6u25-solaris-sparcv9.sh
Solaris SPARC 64-bit - Packages - tar.Z	15.58 MB	jdk-6u25-solaris-sparcv9.tar.Z
Solaris x64 - Self Extracting Binary	8.49 MB	jdk-6u25-solaris-x64.sh
Solaris x64 - Packages - tar.Z	12.25 MB	jdk-6u25-solaris-x64.tar.Z
Windows x86	76.66 MB	jdk-6u25-windows-i586.exe
Windows x64	67.27 MB	jdk-6u25-windows-x64.exe

图 1.1　JDK 下载

2. JDK 的安装

Windows 操作系统上的 JDK 安装程序是一个可执行程序，在安装过程中可以选择安装路径以及安装的组件等，如果没有特殊要求可以选择默认设置。程序默认安装在 C:\Program Files\Java 目录下。

JDK 包括以下用于 Java 开发的组件。

(1) javac——编译器，将后缀名为.java 的源代码编译成后缀名为.class 的字节码。

(2) java——运行工具，运行.class 的字节码。

(3) jar——打包工具，将相关类文件打包成一个文件。

(4) javadoc——文档生成器，从源码注释中提取文档，注释需符合规范。

(5) jdb debugger——调试查错工具。

(6) jps——显示当前 Java 程序运行的进程状态。

(7) javap——反编译程序，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。

(8) appletviewer——运行和调试 Applet 程序的工具，不需要使用浏览器。

(9) javah——产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件。

(10) javaws——运行 JNLP(Java Network Launch Protocol)，用来在网络中部署应用程序的一种协议。

(11) extcheck——检测 jar 包冲突的工具。

(12) apt——apt(Annotation Processing Tool)是一种处理注释的工具，它对源代码文件进行检测找出其中的 Annotation，对 Annotation 进行处理。

(13) jhat——java 堆分析工具。

(14) jstack——栈跟踪程序。

(15) jstat——JVM (Java Virtual Machine)Java 虚拟机检测统计工具。JVM 是虚

构出来的计算机，是通过在实际计算机上仿真模拟各种计算机功能来实现的。

(16) jstatd——jstat 守护进程。

(17) jinfo——获取正在运行或崩溃的 Java 程序配置信息。

(18) jmap——获取 Java 进程内存映射信息。

(19) idlj——IDL-to-Java 编译器.将 IDL(Interactive Data Language)交互式数据语言转化为 Java 文件，是一种数据分析和图像化应用程序及编程语言。

(20) policytool——GUI 图形用户界面的策略文件创建和管理工具。

(21) jrunscript——命令行脚本运行。

JDK 中还包括完整的 JRE，用于产品环境的各种库类，如基础类库 rt.jar，以及给开发人员使用的补充库，如国际化与本地化的类库、IDL 库等。JDK 中还包括各种样例程序用于展示 Java 类库中的各部分。

3. JDK 的配置

JDK 提供的编译和运行工具都是基于命令行的，所以需要进行 Windows 系统下环境变量的设置，这样就可以在任意路径下直接使用 bin 目录下的各种组件和 Java 类库。配置的参数为操作系统中的 Path 和 CLASSPATH 环境变量。

在桌面上右击“计算机”图标，在弹出的“系统属性”对话框中选择“高级”选项卡进行系统设置，在弹出的“环境变量”对话框中找到 Path 变量，单击“编辑”按钮，在“变量值”后输入 C:\Program Files\Java\jdk1.6.0_25\bin，如图 1.2 所示。

新建系统变量 CLASSPATH(类的路径)，在编译运行 Java 程序时，如果调用其他类时，在 CLASSPATH 中寻找需要的类。

输入变量值：“.;C:\Program Files\Java\jdk1.6.0_25\lib;C:\Program Files\Java\jdk1.6.0_25\lib\dt.jar;C:\Program Files\Java\jdk1.6.0_25\lib\tools.jar”。需要注意的是，前面用到的路径 C:\Program Files\Java\jdk1.6.0_25 是 JDK 的默认路径。操作如图 1.3 所示。

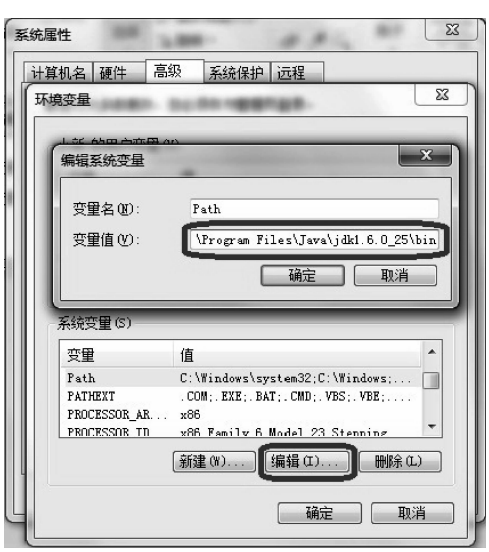

图 1.2　配置 Path 变量

图 1.3　配置 CLASSPATH 变量

4. 测试 JDK

执行 Windows 桌面的“开始”→“运行”命令，打开“运行”窗口。在“运行”文本框中输入 cmd 按 Enter 键，在 DOS 提示符下执行 javac 命令。如果出现 javac 的用法信息，说明环境变量配置成功，如图 1.4 所示。

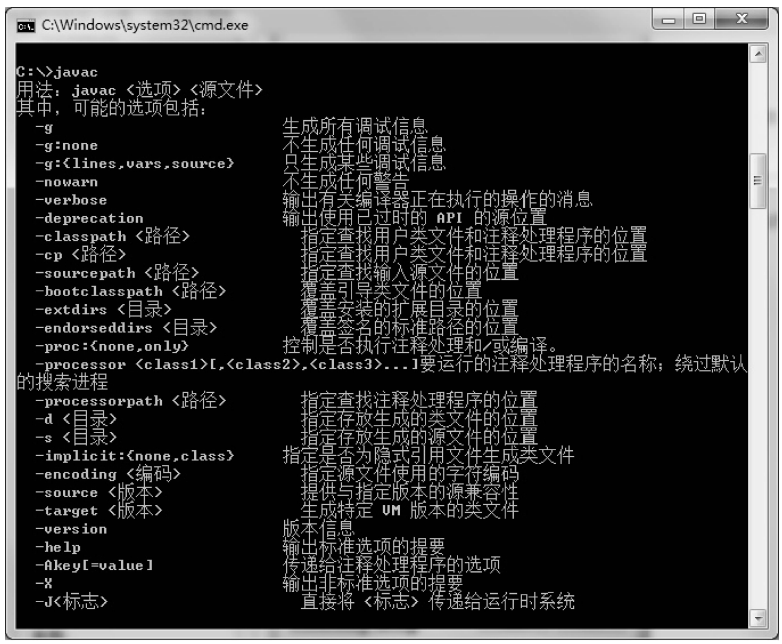

图 1.4 javac 的用法信息

如果出现错误信息则说明环境变量配置失败，如图 1.5 所示。此时需重新检查配置时输入是否有误，比如是否漏掉了“；”。

图 1.5 错误信息

1.1.2 Eclipse 下载与安装

1. 下载 Eclipse 集成开发环境

集成开发环境 Eclipse IDE 是一种辅助开发工具，它将程序设计开发需要的很多功能，例如代码编辑、代码调试、程序部署等一系列功能整合在一起，简化了程序设计开发中的很多操作，提高了程序开发效率。Eclipse 是著名的跨平台开源集成开发环境，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它由 Eclipse 项

目、Eclipse 工具项目和 Eclipse 技术项目三个项目组成，每一个项目由一个项目管理委员会监督，并用它的项目章程管理。本书所用的版本是 Eclipse 3.5（Galileo），可以在 Eclipse 官方网站的下载栏目 http://www.eclipse.org/downloads/中免费下载 Eclipse 的最新版本，本书截稿时 Eclipse 3.6（Helios）已经正式发布。在下载页面上可以看到针对不同开发者有各种不同的插件可打包下载，如图 1.6 所示。建议初学者下载 Eclipse IDE for Java Developers。

图 1.6　下载 Eclipse

2. 安装 Eclipse

要安装 Eclipse，只须将下载的压缩包在原路径下直接解压，在解压缩之后找到 Eclipse.exe 运行就可以了。只要之前的 JDK 安装正确无误，一般不会有什么问题。Eclipse 的国际化工作由 babel 插件项目组维护，其官方网址为 http://babel.eclipse.org/babel/。

Eclipse 的汉化有两种方法，第一种方法是在 babel 的网站下载所有简体中文汉化包，注意每个 Eclipse 版本都有对应的下载链接。然后将这些汉化包解压缩到的 Eclipse 根目录下。第二种方法是在线安装，这种方法的好处是，如果汉化内容增加或者变化时可以及时地在线更新。在菜单栏中选择 Help→Install New Software 命令打开 Install 对话

框，在 Work with：文本框中输入 babel（插件）。项目在线更新与安装地址为 http://download.eclipse.org/technology/babel/update-site/R0.8.0/Galileo。

每个 Eclipse 版本都有不同的在线更新与安装的地址，在连接到项目地址并在下载了所有项目列表后，在列表中找到 Simplified Chinese（简体中文）并单击 Next 按钮，等待接收列表，然后单击 Next 按钮，选择“接受相关协议说明”，单击 Finish（完成）按钮等待自动安装。安装完成后会询问是否要重启 Eclipse，单击 Yes 按钮，重启之后就可以看到中文界面的 Eclipse 了。

1.1.3 知识拓展——jigloo 下载及安装

本书中，创建 SWT 图形用户界面使用的是 jigloo 插件，它和 Eclipse 的汉化插件 babel 一样，也有两种方式。在 jigloo 的官方网站 http://www.cloudgarden.com/jigloo/ 可以下载 jigloo 的打包文件。本书使用的版本是 4.6.2。把下载的打包文件直接解压到 Eclipse 的根目录下就完成了安装。jigloo 在线安装与更新的方法与汉化插件是一样的，下载地址为 http://cloudgarden1.com/update-site。

1.2 任务 2：Java 项目创建与运行

技能目标

（1）熟悉 Eclipse 集成开发环境。

（2）熟练使用 Eclipse 创建项目和类。

（3）熟练使用 Eclipse 的配置工具。

知识目标

（1）了解项目的概念。

（2）掌握 Java 源文件的概念。

（3）掌握 Java 类文件的概念。

（4）了解 Eclipse 的插件机制。

工作任务

（1）使用 Eclipse 创建项目。

（2）使用 Eclipse 创建类。

（3）编辑 main 方法。

（4）运行含有 main 方法的类。

1.2.1 创建 Java 项目

1. 启动 Eclipse

第一次运行 Eclipse 会弹出“工作空间启动程序”对话框，组合列表框中出现的是默认文件夹，可以自己重新建立，也可以将同一项目的子项目保存在这个目录下，如图 1.7 所示。

2. 新建 Java 项目

Eclipse IDE for Java Developers 默认情况下打开的是 Java 透视图。单击工具栏上

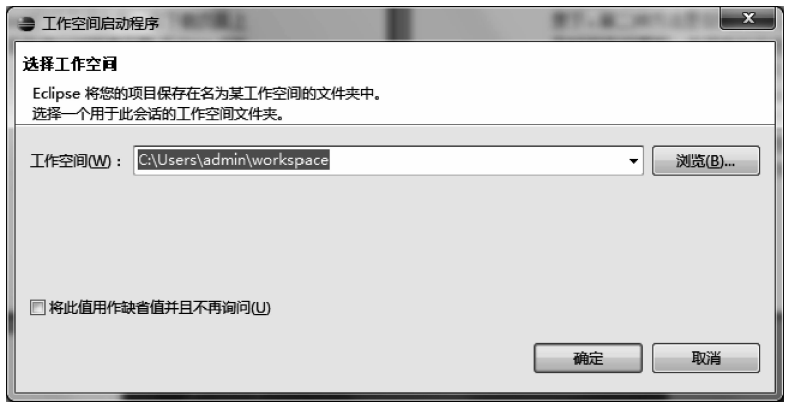

图 1.7　“工作空间启动程序”对话框

的“新建 Java 项目”按钮，打开“新建 Java 项目”对话框；或者选择“文件”→“新建”→“Java 项目”命令。在“项目名”文本框内输入 ch1，创建一个名为 ch1 的项目，保持对话框中的其他选项默认值不变，单击“完成”按钮创建该项目，如图 1.8 所示。

图 1.8　新建 Java 项目

3. 新建 Java 类

在左边“包资源管理器”中选中刚才所创建的项目，然后单击“新建 Java 类”按钮；或者选择“文件”→“新建”→“类”命令，打开“新建 Java 类”窗口。将要创建的类命名为 HelloWorld(注意单词的首字母大写)。在对话框的下半部选择“想要创建哪些方法存根”时，只选择 public static void main(String[ ] args)一项。在对话框的顶部会看到“建议不要使用缺省包”的警告信息。为了简单起见，现在允许这个问题存在。然后，单击“完成”按钮来创建该类，如图 1.9 所示。

图 1.9　新建 Java 类

4. 编辑 main()方法

在 main()方法中输入“System.out.println("Hello,World");”，在工具栏上单击“保存”按钮或者按 Ctrl+S 键将所作的修改保存到 HelloWorld.java 中，如图 1.10 所示。

5. 运行 Java 类

在“包资源管理器”中展开 HelloWorld.java。在类图标的右下角有个三角符号表示该类含有 main()方法，是一个可执行类。在“包资源管理器”中，右击文件 HelloWorld.java，从弹出的快捷菜单中选择“运行方式”→“Java 应用程序”命令。此时，类 HelloWorld 中的 main()方法运行，输出结果显示在下方的控制台视图中，如图 1.11 所示。

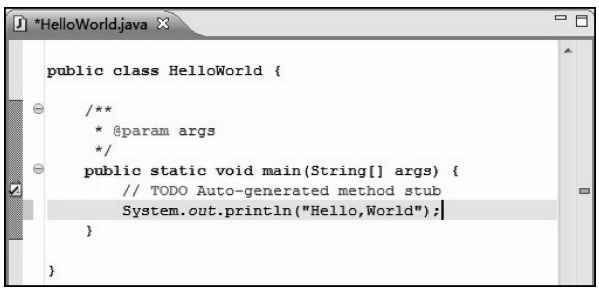

图 1.10　编辑 main()方法

图 1.11　运行 Java 类

1.2.2　问题探究——集成开发环境 Eclipse 的配置

Eclipse 是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目 3 个项目组成，每一个项目由一个项目管理委员会监督，并由它的项目章程管理。

在开展进一步学习之前，有必要了解首选项的设置。在菜单栏中选择“窗口”→“首选项”命令，在弹出的“首选项”窗口内可以根据用户的习惯对 Eclipse 中所列项目的外观和行为进行配置，从字体和颜色到打开新视图和新透视图的相关设置，再到为 Eclipse 中使用其他程序设置的文件类型关联等，涉及项目开发应用的各个方面。Eclipse 的首选项设置如图 1.12 所示。

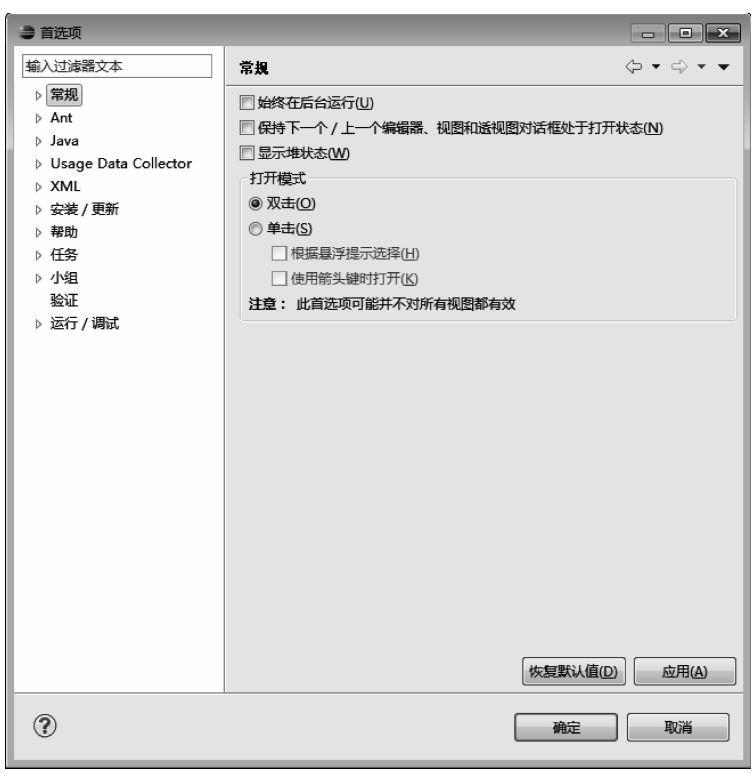

图 1.12　“首选项”窗口

由于 Eclipse 提供了用户界面的渐进显示，即 Eclipse 的功能是基于当前的活动或任务逐步导入的。这些活动有助于使用向导、透视图以及首选项等操作。所以，根据已经做的和正在做的操作，在计算机上看到的“首选项”窗口可能与图 1.12 不同。

可以共享首选项设置，对于开发小组中的一个成员，这是一个非常有用的特点。通过选择菜单里“文件”→“导出”命令，将首选项设置保存到一个文件里。然后通过“导入”命令，开发小组成员可以将该文件导入自己的当前开发环境改变首选项的配置，使用共同的配置环境。

对于构建在 Eclipse 之上的其他工具或插件来说，他们的首选项也将在 Eclipse 的“首选项”对话框中出现，对于已经安装配置并运行在 Eclipse 上的任何工具或插件来说，可以在“首选项”对话框中对其首选项进行管理。在本书接下来的部分，除非另有说明，当对某一行为进行描述时，指的是默认首选项下的系统行为。

在任务 2 中新建的 Java 类里有一些自动生成的注释以及任务标记，在首选项中可以对这些内容进行定制。打开“首选项”对话框，展开 Java→“代码样式”→“代码模板”。选中注释下的“类型模板”，单击“编辑”按钮，在弹出的“编辑模板”对话框中修改其中的内容，例如加入版权信息。接下来选中代码中的“方法主体模板”，单击“编辑”按钮，删除其中的内容。// ${todo} Auto-generated method stub 和 ${body_statement}，这样在新建的类中就看不到任务标记以及这行注释了。再创建一个新的 Java 类，以检验对这两个模板的更改是否已经生效。

1.2.3 知识拓展——Eclipse 插件

Eclipse 的设计思想是“一切皆插件”。Eclipse 内核很小，其他所有功能都以插件的形式附加于 Eclipse 内核之上。Eclipse 基本插件包括图形 API(Application Programming Interface)、Java 开发环境插件 JDT(Java Development Tooling)和插件开发环境 PDE(Plug-in Development Environment，它本身也是一个插件)等。

Eclipse 的插件机制是轻型软件组件化架构。Eclipse 使用插件来提供所有的附加功能，例如支持 Java 以外的其他语言。目前已有的插件能够支持 C/C++(CDT)、PHP、Perl、Ruby、Python、Telnet 和数据库开发。Eclipse 插件架构能够支持将任意的扩展加载到现有环境中，例如配置管理，而绝不仅限于支持各种编程语言。

1.2.4 知识拓展——Java 类文件

Java 语言的源程序代码由一个或多个编译单元组成。这里所说的编译单元是指保存源代码的.java 文件。每个编译单元只能包含下列内容(空格和注释除外)。

(1) 一个程序包语句(package statement)。

(2) 入口语句(import statement)。

(3) 类的声明(class declaration)。

(4) 接口声明(interface declaration)。

Java 的运行系统工作起来如同一个虚拟机，当启动一个 Java 程序时，一个虚拟机实例也就诞生了。当程序关闭退出，这个虚拟机实例也就随之消亡。Java 的源程序代码被编译后，便产生了 Java 字节代码文件。Java 的字节代码由于不依赖于机器的指令组成，