

第 1 章 Java 程序设计语言概述

本章内容：介绍 Java 语言的特点、目标、开发环境的搭建、运行原理以及开发工具的使用。

学习目标：

- 了解 Java 语言的几个主要特点及 Java 程序的运行原理
- 能够独立安装 JDK 开发工具、配置 Java 运行环境
- 熟悉 Eclipse 集成开发环境，编写并运行一个 Java Application 程序

1.1 Java 简介

Java 是由 Sun 公司开发的一种应用于分布式网络环境的程序设计语言，它已经成为了一个真正意义上的语言标准，Java 的标准指的是一种作为应用层封装的标准，使用 Java 可以调用一些底层的操作，例如 Android 开发就是利用了 Java 调用 Linux 内核操作形成的。一般的初学者认为 Java 是一种编程语言，实际上，Java 不仅是一种语言，它更是一个平台。提供了开发类库、运行环境、部署环境等一系列支持。

根据应用范围的不同，Java 分为三个版本：Java SE、Java EE 和 Java ME。

Java SE（Java Standard Edition）包含了标准的 JDK、开发工具、运行时环境和类库，适合开发桌面应用程序和底层应用程序。同时它也是 Java EE 的基础平台。

Java EE（Java Enterprise Edition）采用标准化的模块组件，为企业级应用提供了标准平台，简化了复杂的企业级编程。现在 Java EE 已经成为了一种软件架构和企业级开发的设计思想。

Java ME（Java Micro Edition）包含高度优化精简的 Java 运行时环境，主要用于开发具有有限的连接、内存和用户界面能力的设备应用程序。例如移动电话（手机）、PDA（电子商务）、能够接入电缆服务的机顶盒或者各种终端和其他消费电子产品。

1.2 Java 语言的特点

Java 语言具有简单性、面向对象、分布式、解释通用性、健壮性、安全性、可移植性、高性能、多线程、动态等特性。另外还提供了丰富的类库，方便用户进行自定义操作。

1.3 Java 的目标

Internet 的迅猛发展，使 Java 迅速成为了最流行的网络编程语言。最初设计 Java 有以下几个目标：

- （1）不依赖于特定的平台，一次编写到处运行。

- (2) 完全的面向对象。
- (3) 内置的对计算机网络的支持。
- (4) 借鉴 C++ 优点，尽量简单易用。

1.4 Java 开发环境的搭建

1.4.1 JDK 的下载安装

JDK (Java Development Kit) 是 Java 的开发工具包，是 Java 开发者必须安装的软件环境。JDK 包含了 JRE 和开发 Java 程序所需的工具，如编译器、调试器、反编译器和文档生成器等。JRE (Java Runtime Environment) 是 Java 运行时环境，包含了类库和 JVM (Java 虚拟机)，是 Java 程序运行的必要环境。如果仅仅运行 Java 程序只安装 JRE 就可以了。Sun 公司网站下载 JDK 的网址为：<http://java.sun.com/javase/downloads/index.html>。

需要注意，Java 是跨平台的开发语言，根据平台的不同要选择不同的 JDK。本书选择 Windows 平台，在这里 JDK 又分为在线安装包和离线安装包两种，选择离线安装方式。下载的 JDK1.6 安装包保存到硬盘上，文件名为 `jdk-6u2-windows-i586-p.exe`，执行该文件并按照向导安装。安装前最好关闭防火墙，关闭所有正在运行的程序，接受许可协议，设置 JDK 的安装路径及选择安装的组件对话框，如图 1-1 所示。



图 1-1 设置 JDK 的安装路径及选择安装的组件对话框

更改安装路径到 “D:\Java\jdk1.6”，选择要安装的组件。在安装过程中定义 JRE 安装路径的提示对话框，更改路径到 “D:\Java\jre1.6”。在弹出安装完成的提示对话框中，取消 “显示

自述文件”复选框的勾选，单击“完成”按钮，即可完成JDK的安装。安装到图1-2的目录：

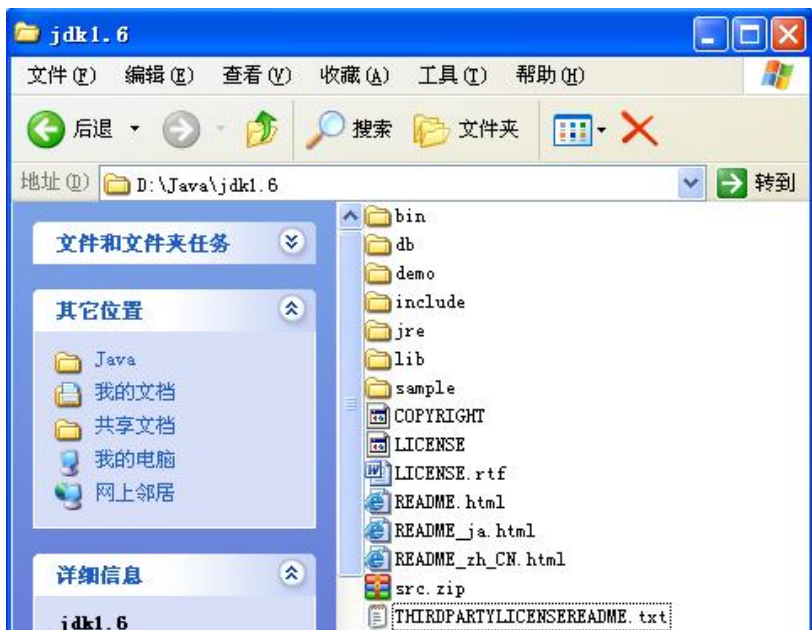


图 1-2 JDK 安装路径

主要目录和文件简介如下：

bin 目录：开发工具，包括开发、运行、调试和文档生成的工具，主要是*.exe 文件。

lib 目录：类库，开发时需要的一些类库和文件。

jre 目录：运行时环境，包括 Java 虚拟机、类库、辅助运行的支持文件。

demo 目录：演示文件，附源代码的 Java 文件，演示了 Java 的一些功能。

include 目录：C 语言头文件，支持 Java 本地方法调用的必要文件。

src.zip 文件：Java 核心类源文件，感兴趣的学习者可以解压后研究。

其中，bin 目录中的两个文件最重要，编程中经常使用：

javac.exe——Java 编译器。

java.exe——Java 解释器，调用 Java 虚拟机执行 Java 程序。

单击“开始”/“运行”输入cmd，如图1-3所示。进入DOS命令行，输入Java -version 出现如图1-4所示的界面，即为安装成功。



图 1-3 进入 DOS 命令行

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Owner>java -version
java version "1.6.0_02"
Java(TM) SE Runtime Environment (build 1.6.0_02-b06)
Java HotSpot(TM) Client VM (build 1.6.0_02-b06, mixed mode, sharing)

C:\Documents and Settings\Owner>
```

图 1-4 测试 JDK 是否安装成功

1.4.2 Java 开发环境配置

安装完 JDK 后，需要设置环境变量及测试 JDK 配置是否成功，Windows XP 系统下具体步骤如下：

(1) 在“我的电脑”上单击鼠标右键，选择“属性”菜单项。在打开的“系统属性”对话框中选择“高级”选项卡，单击“环境变量”按钮，打开“环境变量”对话框，选择针对所有用户的“系统变量”区域中的“新建”按钮。

(2) 在“变量名”文本框中输入“JAVA_HOME”，在“变量值”文本框中输入 JDK 的安装路径，单击“确定”按钮，如图 1-5 所示，完成环境变量 JAVA_HOME 的配置。



图 1-5 JAVA_HOME 环境变量的配置

(3) 在“系统变量”中查看 path 变量，如果不存在，则新建变量 PATH，否则选中该变量，单击“编辑”按钮，在“变量值”文本框的起始位置添加以下内容：%JAVA_HOME%\bin;，单击“确定”按钮，注意不要漏掉最后的“;”符号。

提示：尽量把“变量值”放在最前面，以避免安装有 Oracle 数据库或其他自带 JDK 存在的版本问题。

(4) 在“系统变量”中查看 classpath 变量，如果不存在，则新建变量 classpath，单击“新建”按钮，“变量值”为：.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar。

(5) 测试 JDK 是否能够在机器上运行，在 DOS 命令行窗口输入“javac”，输出帮助信息即为配置正确。通过 Java -version 查看 JDK 版本，判断是否安装完整。

1.5 Java 程序运行的原理

Java 程序分为两种类型，一种是 Application 程序，另一种是 Applet 程序，其中 Applet 程序主要应用在网页编程上，现在已经基本不再使用，本书不再做任何介绍。有 main 方法的程序是 Application 程序，本书主要讲解的是 Application 程序。

编写 Java Application 程序可以使用简单的 Windows 记事本程序来编写，下面是一个用 Windows 记事本编写的简单 Java 文件，如图 1-6 所示。

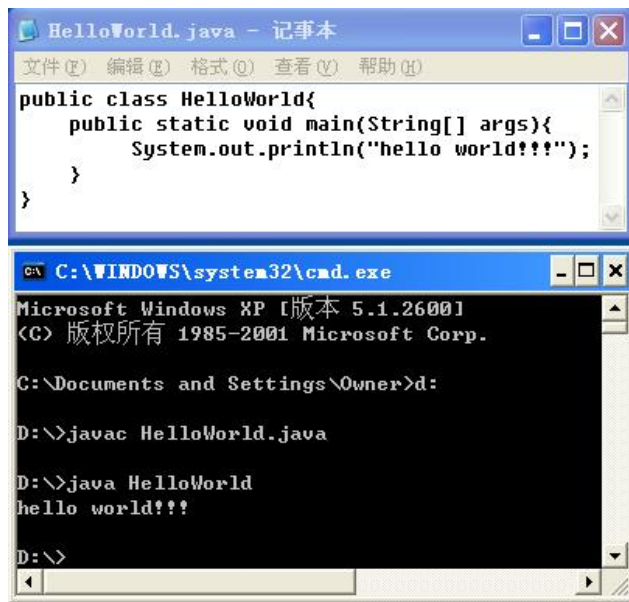


图 1-6 Java 的开发运行过程

把代码保存到 D 盘，命名为 HelloWorld.java，在 DOS 命令行编译源代码：Javac HelloWorld.java，编译正确生成 Hello World.class 文件，用 Java 解释器解释执行 class 文件：Java Hello World。如图 1-6 所示。

可以通过如上程序的运行过程来了解 Java 的运行原理，如图 1-7 所示。

通过如上的程序运行原理图可以发现，任何一个*.java 程序首先必须经过编译，编译之后会形成一个*.class 的文件（字节码文件），在电脑上执行的不是*.java 文件，而是编译之后的*.class 文件（这个文件可以理解为“加密”的文件），但是解释程序的电脑并不是一台真正意义上的电脑，而是一台由软件和硬件模拟出来的电脑——Java 虚拟机。

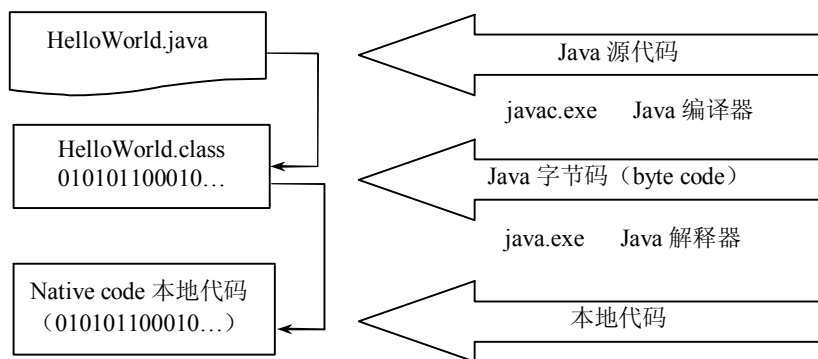


图 1-7 Java 程序运行原理

Java 虚拟机 (Java Virtual Machine) 是在一台计算机上由软件或硬件模拟的计算机。Java 虚拟机的最大作用是对各个平台的支持，通过如图 1-8 可以发现，所有要解释的程序在 JVM 上执行，但是由不同版本的 JVM 去匹配不同的操作系统，这样只要 JVM 的支持不变，程序可以任意地在不同的操作系统上运行。因此，Java 编译器针对 Java 虚拟机产生 class 文件，是独立于平台的，Java 解释器负责将 Java 虚拟机的代码在特定的平台上运行，实现了可移植性。即 Java “一次编写，到处运行”的特性。但是这种运行方式很明显没有直接运行在操作系统上性能高，不过随着硬件技术的发展，这些问题几乎可以忽略了。

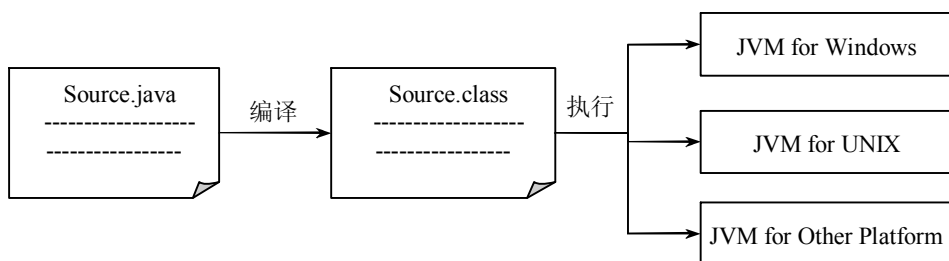


图 1-8 JVM 运行图

1.6 Java 开发工具 Eclipse

Eclipse 是基于 Java 的、开放源码的、可扩展的应用开发平台，它为编程人员提供了一流的 Java 集成开发环境 (Integrated Development Environment, IDE)。Eclipse 是利用 Java 语言编写的，因此 Eclipse 可以支持跨平台操作，是一个成熟的可扩展的体系结构。它的价值还体现在为创建可扩展的开发环境提供了一个开发源代码的平台。这个平台允许任何人构建与环境或其他工具无缝集成的工具，而工具与 Eclipse 无缝集成的关键是插件。通过不断地集成各种插件，Eclipse 的功能也在不断地扩展，以便支持各种不同的应用。

1.6.1 Eclipse 的安装与启动

安装 Eclipse 前需要先安装 JDK，关于 JDK 的安装和配置参见 1.4 节中的内容。可以从

Eclipse 的官方网站(<http://www.eclipse.org>)下载最新版本的 Eclipse。本书使用的 Eclipse Galileo 版本是 3.5。

Eclipse 下载完成后，解压，即完成了 Eclipse 的安装。

在 Eclipse 初次启动时，需要设置工作空间，本书中将 Eclipse 安装到 D 盘根目录下，将工作空间设置在“D:\eclipse\workspace”中，如图 1-9 所示。

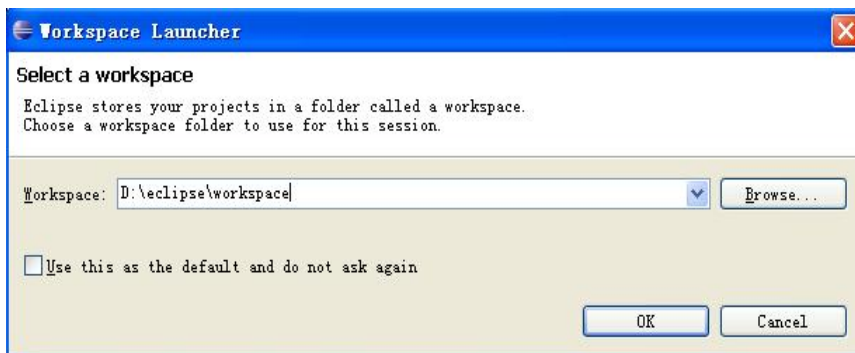


图 1-9 设置工作空间

每次启动 Eclipse 时，都会出现设置工作空间的对话框，如果不需要每次启动都出现该对话框，可以勾选“Use this as the default and do not ask again”选项将该对话框屏蔽。

单击“OK”按钮，进入到 Eclipse 工作台。如图 1-10 所示为 Eclipse 的欢迎界面。

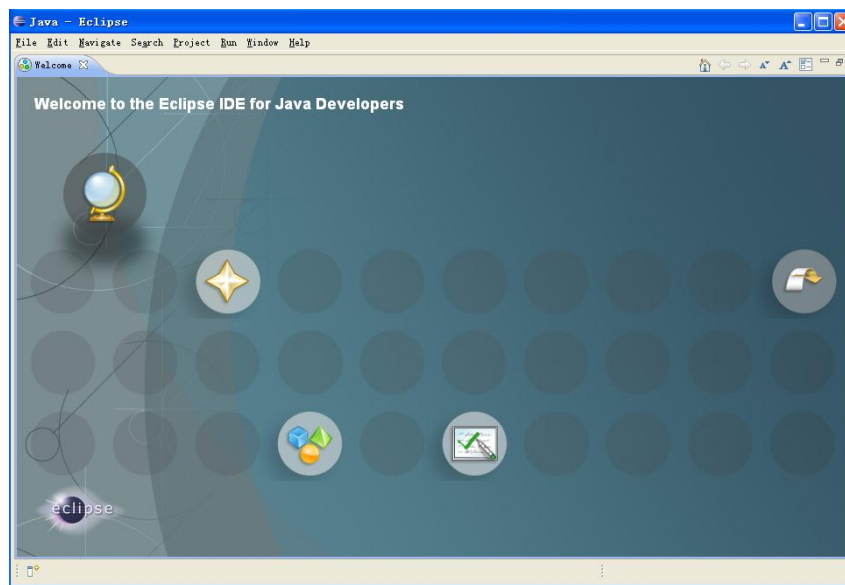


图 1-10 Eclipse 的欢迎界面

Eclipse 工作台是一个 IDE 开发环境。它可以通过创建、管理和导航工作空间资源提供公共范例来获得无缝工具集成。每个工作台窗口可以包括一个或多个透视图。透视图可以控制出现在某些菜单栏和工具栏中的内容。主要有标题栏、菜单栏、工具栏、透视图几部分组成（如图 1-11 所示），其中透视图包括视图和编辑器。

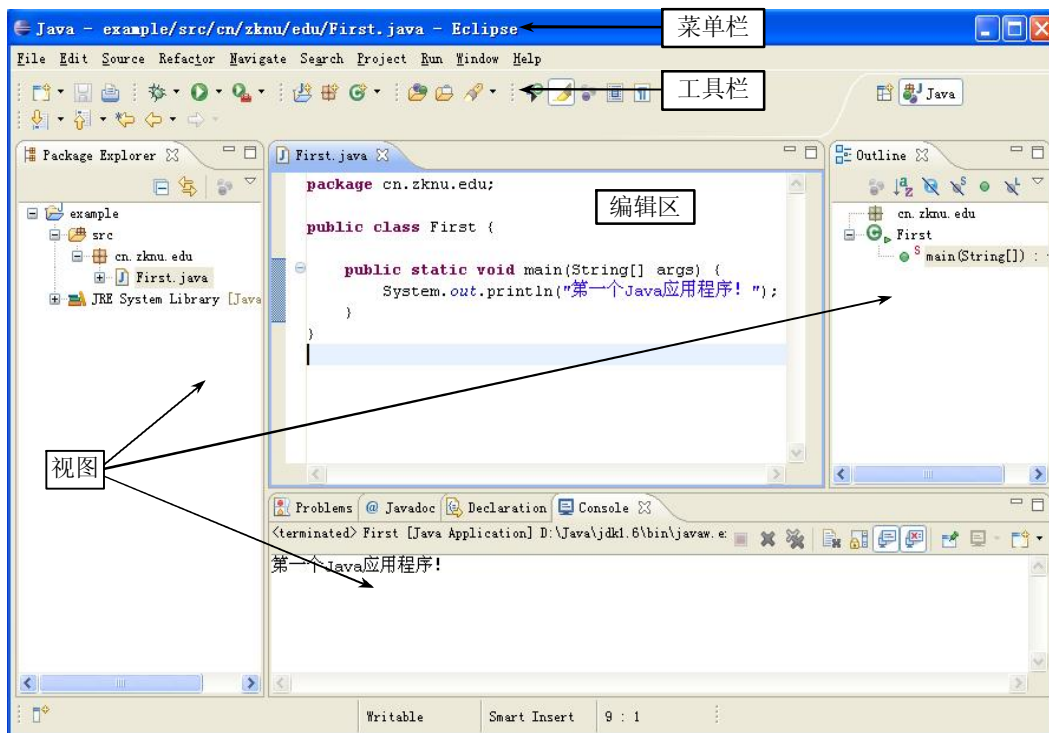


图 1-11 Eclipse 工作台

1.6.2 Eclipse 编写 Java 程序的流程

Eclipse 编写 Java 程序的流程必须经过新建 Java 项目、新建 Java 类、编写 Java 代码和运行程序 4 个步骤，下面分别介绍。

1. 新建 Java 项目

在 Eclipse 中选择“File”/“New”/“Java Project”菜单项，如图 1-12 所示。

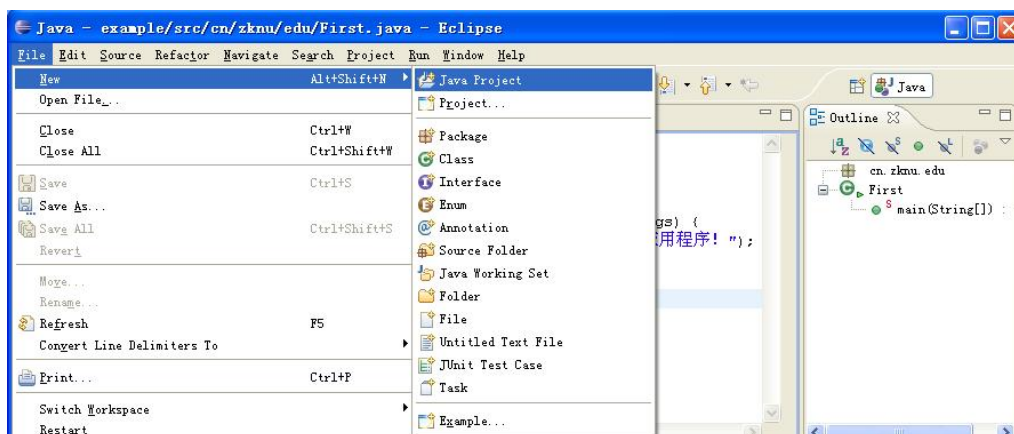


图 1-12 新建 Java 项目

打开新建项目对话框，如图 1-13 所示。



图 1-13 新建项目对话框

单击“Next”按钮，进入到 Java 项目构建对话框，配置 Java 的构建路径，如图 1-14 所示。

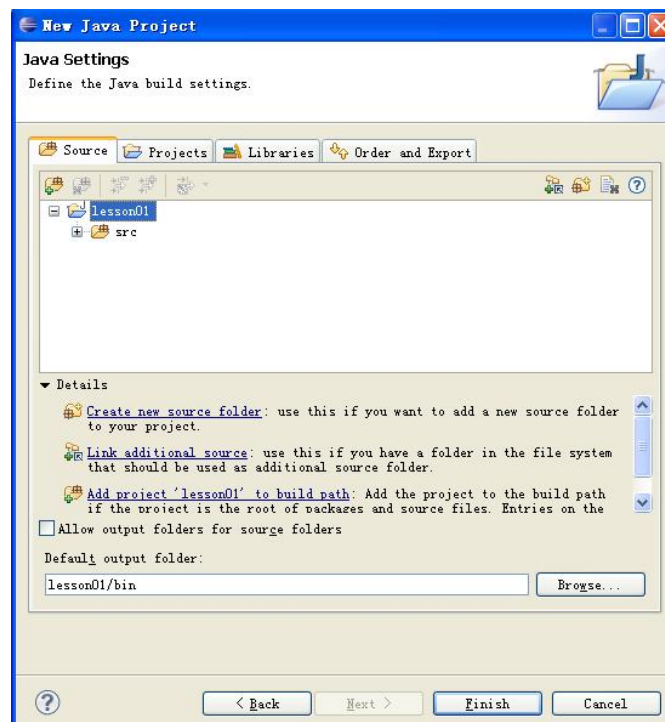


图 1-14 项目创建向导——Java 构建设置

在对话框中，Java 的源文件（Java 文件）放在 src 文件夹，生成的 class 文件放在 bin 文件夹，一般不做修改。单击“Finish”按钮，完成 Java 项目的创建。

完成新建 Java 项目后，在“Package Explorer”（包资源管理器）视图将出现新建的项目 lesson01。如图 1-15 所示。包资源管理器视图中包含所有已经创建的 Java 项目。

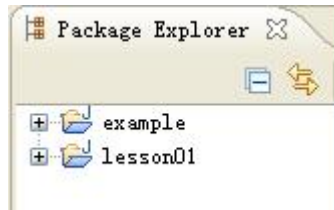


图 1-15 包资源管理器

2. 新建 Java 类

在 lesson01 中创建 Java 类，具体步骤如下：

（1）在 lesson01 上鼠标右击“New”/“Class”，弹出“New Java Class”（新建 Java 类）对话框，如图 1-16 所示。

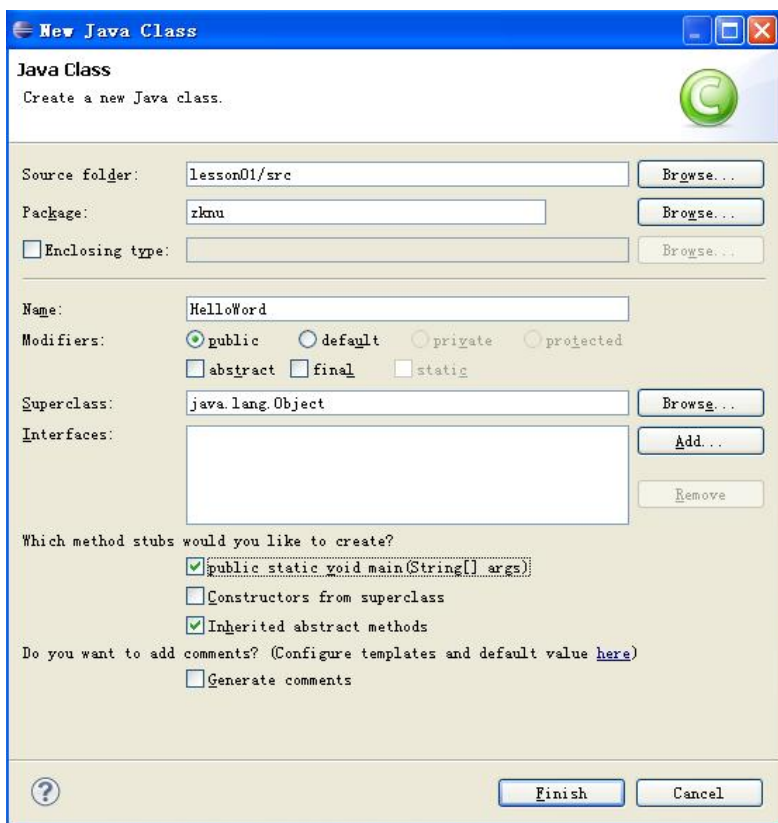


图 1-16 创建 Java 类

- Source folder（源文件夹）：用于输入新类的源代码文件夹。
- Package（包）：用于存放新类的包。
- Enclosing type（封闭类型）：选择此项用以选择要在其中封装新类的类型。
- Name（名称）：输入新建 Java 类的名称。
- Modifiers（修饰符）：为新类选择一个或多个访问修饰符。
- Superclass（超类）：为新类选择超类，默认为 java.lang.Object 类型。
- Interfaces（接口）：编辑新类实现的接口，默认为空。

(2) 在新类中选择创建哪些方法，默认选项为：

- 将 main 方法添加到新类中。
- 从超类复制构造方法到新类中。
- 继承超类或接口的方法

(3) 单击“完成”按钮，完成 Java 类的创建。

3. 编写 Java 代码

在 Eclipse 编辑区编写 Java 程序代码，Eclipse 会自动打开源代码编辑器。下面是编辑区里 HelloWorld 类的代码。

```
package zknu;
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("第一个Java应用程序!");
    }
}
```

4. 运行 Java 程序

单击工具栏  按钮右侧的小箭头，在弹出的下拉菜单中选择“Run As/Java Application”菜单项，如图 1-17 所示，程序开始运行，运行结束后，在控制台视图中将显示程序的运行结果，如图 1-18 所示。

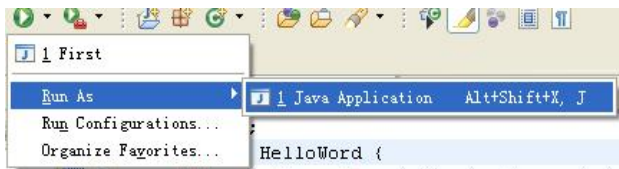


图 1-17 运行 Java 程序



图 1-18 程序执行结果

关键字 class 声明了类的定义，HelloWorld 是描述类名的标识符，整个类的定义包括其所有成员都是在一大括号中（即{ }之间）定义完成的，这标志着类定义块的开始和结束。

程序从 main()方法开始执行，所有的 Java 应用程序都必须有一个 main()方法，main()方法是所有 Java 应用程序的起始点。

注意：Java 是区分大小写的，所以 main 与 Main 不同。

关键字 public 是一个访问修饰符，它控制类成员的可见度和作用域。

关键字 void 告诉编译器在执行此 main()方法时，它不会返回任何值。

关键字 static 允许调用 main()方法，而无需创建类的实例。

String []args 是传递给 main()方法的参数，Args[]是 String 类型的数组，String 类型的对象用来存储字符串。Print()方法在屏幕上输出以参数方式传递给它的字符串，System 是一个预定义的类，它提供对系统类的访问，out 是连接到控制台的输出流。

本章小结

本章首先介绍了 Java 的特点和目标，然后引导读者完成 Java 开发环境的搭建，其中包括

JDK 的下载和安装, Java 运行环境, 又介绍了 JDK 相关环境变量的配置和 JDK 环境的测试方法。通过 Java 程序的运行过程让读者理解 Java 程序的运行原理。最后, 为使读者能够快速掌握 Java 语言程序设计的相关语法、技术以及其他知识点, 为大家介绍了目前流行的 IDE 集成开发工具 Eclipse 及其使用方法, 开发 Java 程序的流程。

习 题

- 1-1 何为字节码? Java 程序平台无关性的实现原理是什么?
- 1-2 实践安装 JDK, 并配置系统环境变量。
- 1-3 简述 Java 程序的运行原理。
- 1-4 在下面情况下, 程序能不能通过编译, 能不能运行, 若能运行其结果是什么?

```
public class J_test {  
    public static void main() {  
        System.out.println("hello,world");  
    }  
}
```