

Java 与JSP 快速入门

李 兵 杜学功 杨海军 叶燕文 编



科学出版社

Java 与 JSP 快速入门

李 兵 杜学功 杨海军 叶燕文 编

科学出版社

北 京

内 容 简 介

本书根据编者多年的教学与项目开发实践经验编写而成,主要包括四部分的内容,分别为 Java 编程的基础知识、网页设计的基础知识、Servlet 和 JSP,以及综合实例。本书内容简明扼要、重点突出、易于阅读,充分体现了教学的操作性、基础性和实用性等特点。

本书可以作为高等学校计算机、通信管理、电子商务专业和其他相关专业学生的教材,也可供读者自学使用。

图书在版编目(CIP)数据

Java 与 JSP 快速入门 / 李兵等编. —北京: 科学出版社, 2015.12
ISBN 978-7-03-046006-6

I. ①J… II. ①李… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 245989 号

责任编辑: 相 凌 王迎春 / 责任校对: 张怡君

责任印制: 徐晓晨 / 封面设计: 华路天然工作室

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京京华虎彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2015 年 12 月第 一 版 开本: 787×1092 1/16

2015 年 12 月第一次印刷 印张: 15 1/2

字数: 367 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换)

前 言

Java 是目前应用最广的编程语言,因其面向对象、跨平台、多线程、数据库应用和 Internet 网络应用等特性,在各个领域都有很多软件采用 Java 语言开发。Java 也是一个开发平台,它支持从移动设备、桌面到企业级等各个层次的应用,涵盖 Java 基础知识、Web 开发、框架技术、数据库、服务器和 Web 前端技术等技术领域,已成为软件开发人员的理想工具,是当今信息技术的主流之一。

目前国内高校的有关专业都开设了 Java 类课程。本书从应用的角度出发,以较少的篇幅介绍了 Java 语言和 JSP 开发的基础知识,适合于教学时数有限的 Java 课程,使学生经过一个学期的学习,就可以完成简单 Web 应用的设计工作,避免了仅学习 Java 基础知识而产生的用处不大的感觉,也为第二学期的 Java 课程进一步学习框架技术打下基础。

本书由长期从事计算机理论教学、实践教学与科研工作的优秀教师编写,适合计算机、信管和电子商务专业的学生使用。第 1、2、9 章由叶燕文编写,第 3、4、5、6、7、10、14 章由杜学功编写,第 11、12 章由杨海军编写,第 8、13 章由李兵编写并负责统稿。本书在编排上注意内容由浅入深、重点突出、结构合理清晰,语言准确精炼,内容详略适当,理论联系实际,注重利用所学解决实际问题。

由于时间仓促,书中难免存在不足与欠妥之处,为了便于今后的修订,恳请广大读者提出宝贵的意见与建议。

编 者

2015 年 10 月

目 录

第 1 章	Java 简介	1
1.1	运行环境	1
1.2	开发工具	5
1.3	程序错误及调试	12
第 2 章	Java 语言基础	15
2.1	基本数据类型	15
2.2	标识符、常量与变量	17
2.3	运算符与表达式	20
2.4	数据类型转换	25
2.5	控制语句	27
2.6	数组	36
第 3 章	类和包	41
3.1	类	41
3.2	引用对象模型	43
3.3	静态成员	46
3.4	初始化	48
3.5	内部类	51
3.6	包	53
第 4 章	继承	56
4.1	Java 的继承	56
4.2	访问控制	57
4.3	继承结构的初始化	57
4.4	多态性	59
第 5 章	接口	63
5.1	抽象类	63
5.2	接口的声明	64
5.3	接口的实现	65
5.4	接口作用原理	65
第 6 章	Java 集合	69
6.1	概述	69
6.2	List	70
6.3	Set 集合	76
6.4	Map	80
6.5	Collections 工具类	83
第 7 章	异常处理	86
7.1	什么是异常	86

7.2	Java 异常的分类层次	86
7.3	抛出异常与捕获异常	87
7.4	声明异常	89
7.5	自定义异常	90
7.6	Java 的异常处理原则	92
第 8 章	输入/输出设计	93
8.1	File	93
8.2	基本 I/O	95
8.3	Socket 编程	104
8.4	设计 Web 服务器	109
第 9 章	Web 基础知识	119
9.1	HTML 基础知识	119
9.2	JavaScript 基础知识	124
9.3	CSS 基础知识	132
第 10 章	Servlet 与 Tomcat	137
10.1	Tomcat	137
10.2	Servlet 简介	139
10.3	Servlet 生命周期	143
10.4	Servlet 加载	145
第 11 章	JavaBean	147
11.1	JavaBean 的概念	147
11.2	JavaBean 的特征与结构	147
11.3	在 JSP 中使用 JavaBean	157
第 12 章	JSP 中访问数据库	165
12.1	数据库及 SQL 语句	165
12.2	连接数据库的常用方法	172
12.3	JDBC 结构	173
12.4	JDBC 连接实例	177
12.5	使用连接池	191
第 13 章	JSP 基础与应用	195
13.1	JSP 简介	195
13.2	JSP 基本语法	197
13.3	JSP 内置对象	202
13.4	JSP 应用实例	207
第 14 章	并发性	230
14.1	Java 线程	230
14.2	线程间的同步	233
14.3	JSP 与多线程处理	237

第 1 章 Java 简介

Java 来自 Sun 公司的一个叫 Green 的项目，其初始目的是为家用消费电子产品开发一个分布式代码系统，使用这种技术可以把 E-mail 发送给电冰箱、电视机等家用电器，并对它们进行控制或与它们进行信息交流。开始，Sun 公司的技术人员准备采用 C++ 来完成这个设想，但是 C++ 太复杂、安全性差，技术人员不得不放弃 C++，转而开发一种新的语言 Oak（Java 的前身）。Oak 是一种用于网络的精巧而安全的语言。

Internet 的出现给 Java 语言的发展提供了契机，Java 语言继承了前身 Oak 语言能够跨平台运行的特点，融合了面向对象编程的风格，广受当时程序开发人员的好评。Sun 公司又将 Java 语言设计为可以针对移动平台、桌面系统、企业级应用进行开发的综合平台，极大地提高了 Java 语言的生产力。当掌握了 Java 语言的基本语言特性后，再通过学习特定的开发包，就可以开发移动应用程序（如手机游戏）、桌面应用程序（QQ、MP3 播放器都属于桌面应用程序）和企业级的高级应用程序。Sun 公司将三种平台下的开发分别命名为 Java SE、Java ME 和 Java EE，它们是 Java 语言开发的三个分支。

（1）Java SE：对应于桌面开发，可以开发基于控制台或图形界面的应用程序。Java SE 中包括 Java 的基础类库，也是进一步学习其他两个分支的基础。本课程主要的学习内容就是 Java SE 中基于控制台的应用程序开发。

（2）Java ME：对应于移动平台（如手机、PDA 等）设备的开发，因为这类设备的硬件差异很大，而 Java 恰恰具有平台无关的特性，同样的 Java 代码可以在不同的设备上运行，所以在移动平台开发中，Java ME 非常流行。从技术角度上可以认为 Java ME 是 Java SE 的精简版。

（3）Java EE：对应于企业级开发，包括 B/S 架构开发、分布式开发、Web 服务等非常丰富的应用内容，在软件开发企业中被大量应用。

1.1 运行环境

在开发一个 Java 应用程序前，必须了解并搭建好所需要的开发环境。首先要确保计算机上安装有 JDK。JDK 全称是 Java SE Development Kit，又称 Java SE，是 Sun 公司提供的用于开发 Java 应用程序的标准开发工具包，它包括 Java 语言的编译环境、运行环境、调试环境等，并且提供了代码示例、Java 类库源代码等开发资源与参考资料。

1.1.1 下载 JDK

下面以目前最新版本的 JDK 8（Java SE Development Kit 8u40）为例介绍下载 JDK 的方法。由于 Sun 公司已于 2009 年被 Oracle 公司收购，所以 JDK 可以在 Oracle 公司的官网下载。下载 JDK 的具体步骤如下。

（1）打开浏览器，输入网址 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>，出现如图 1-1 所示页面，单击该页面上的方形“Java DOWNLOAD”按钮，或单击页面右下角的“JDK DOWNLOAD”按钮。

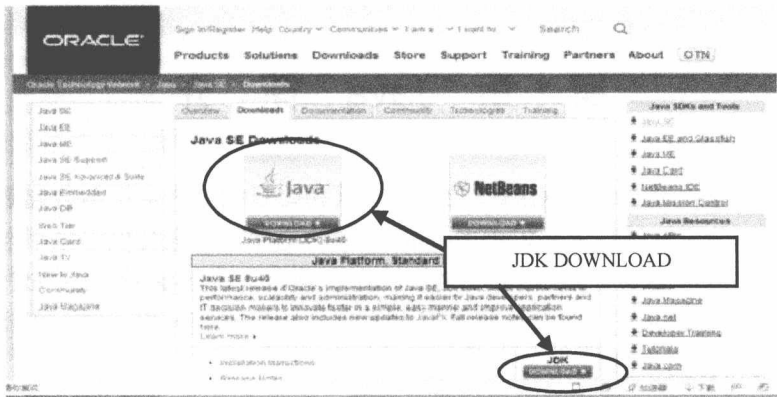


图 1-1 下载页面

(2) 在如图 1-2 所示的页面中选中“Accept License Agreement”单选按钮。



图 1-2 接受许可协议

(3) 在如图 1-3 所示的页面中选择适合自己操作系统平台的下载文件。作者使用的操作系统是 Windows 7，所以选择下载 jdk-8u40-windows-i586.exe 文件。

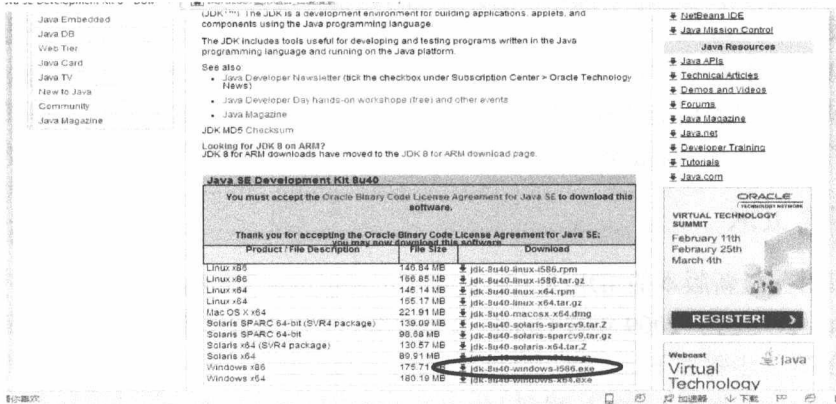


图 1-3 选择 Java 版本

1.1.2 安装 JDK

双击已经下载的 jdk-8u40-windows-i586.exe 文件，安装过程如图 1-4 所示。

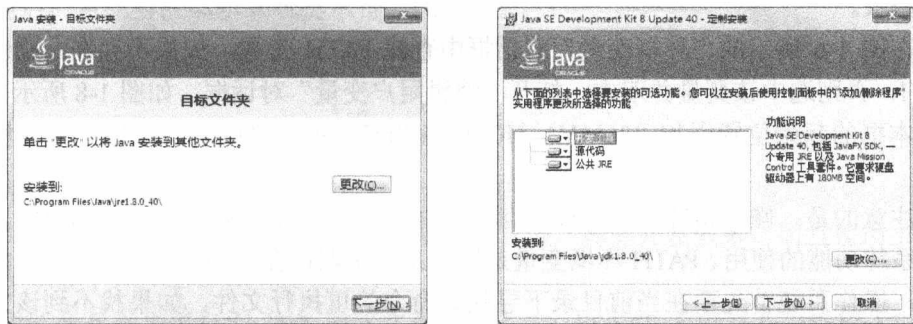


图 1-4 安装 JDK

1.1.3 配置环境变量

在基于 Java 控制台的单机软件开发中，JDK 一般需要进行一些环境变量的配置才能保证程序正常编译和运行。在 Java 程序运行的时候，这些环境变量会告诉操作系统 Java 被安装到了什么地方，怎样才能找到 Java 的编译和运行环境等。JDK 的配置涉及两个环境变量，假设 JDK 的安装目录为 C:\Program Files\Java\jdk1.8.0_40，则环境变量配置如下：

```
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_40
PATH=%JAVA_HOME%\lib;
```

其中，PATH 环境变量中需要配置多个目录，多个目录之间用分号（“;”）隔开。
在 Windows 7 操作系统下配置环境变量的步骤如下。

- (1) 右击桌面上的“计算机”图标，在弹出的快捷菜单中选择“属性”命令，在弹出的“属性”对话框左侧单击“高级系统设置”超链接，将出现如图 1-5 所示的“系统属性”对话框。
- (2) 单击图 1-5 对话框中的“环境变量”按钮，弹出如图 1-6 所示界面。

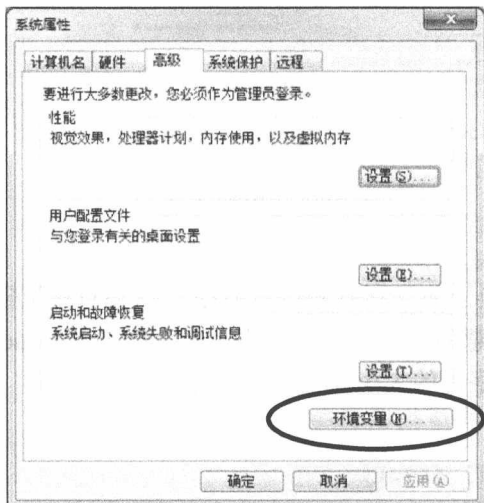


图 1-5 “系统属性”对话框

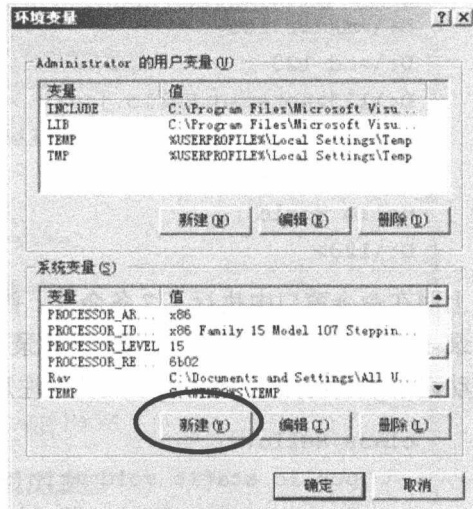


图 1-6 编辑环境变量

(3) 单击图 1-6 界面中的“新建”按钮，创建新的系统变量 JAVA_HOME。

(4) 在如图 1-7 所示的“新建用户变量”对话框中，分别输入变量名 JAVA_HOME 和变量值 (JDK 的安装路径)，图 1-7 中的变量值是作者的安装路径，读者需要根据自己的计算机环境进行修改。

(5) 在图 1-6 所示的“环境变量”对话框中查找 PATH 变量，如果不存在，则新建系统变量 PATH，否则选中该变量并双击，打开“编辑用户变量”对话框，如图 1-8 所示。在“变量值”文本框的起始位置添加“%JAVA_HOME%\lib;”。单击“确定”按钮完成环境变量的设置。

需要注意的是：修改系统的 PATH 环境变量的时候，一定不要删除已有的内容，否则会影响部分系统功能的使用。PATH 环境变量是操作系统中默认存在的环境变量，当执行一个外部命令时，操作系统首先要在当前目录下寻找该命令的可执行文件，如果找不到该文件就会到系统的 PATH 环境变量所指定的目录下寻找该命令。

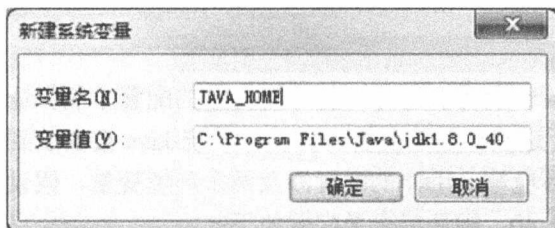


图 1-7 新建 JAVA_HOME 变量

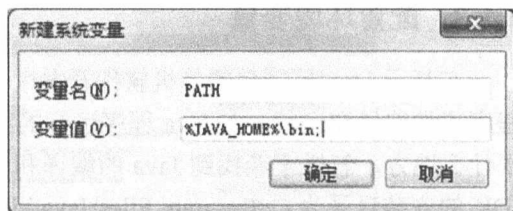


图 1-8 修改 PATH 变量

(6) JDK 程序的安装和配置完成后，需要测试 JDK 是否能够在计算机上运行。具体测试步骤如下：按 Win+R 键，打开“运行”对话框，输入 cmd 并单击“运行”按钮，将打开 Windows 的命令窗口。在命令提示符后面直接输入 javac，并按 Enter 键，系统将会输出 JDK 的编译器信息，这说明 JDK 环境搭建成功。

(7) 配置好 JDK 后就可以运行 Java 程序了。在命令窗口中执行下列命令：

```
C:\>D:
D:\>mkdir 123
D:\>cd 123
D:\123>notepad Hello.java
D:\123>javac Hello.java
D:\123>java Hello
Hello World
D:\123>
```

上面在命令窗口中执行了 7 条命令。首先切换当前盘到 D 盘，然后在 D 盘根目录下创建子目录 123，并用 cd 命令进入 123 子目录中，注意系统提示符的变化。接下来用 notepad 命令启动记事本编辑如下 Java 程序文件（注意英文字母的大小写）：

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

编辑完成后，保存文件，注意程序中 `class` 后面的类名 `Hello` 与文件名保持一致，而且文件扩展名为 `.java`。第 6 条命令用 `javac` 编译源程序，这时程序名 `Hello.java` 作为参数写在命令单词 `javac` 的后面。最后用 `java Hello` 来执行程序。命令的下一行是程序执行的结果。当程序执行完后，系统再次显示系统提示符 “`D:\123>`”。

1.2 开发工具

虽然使用记事本和 `JDK` 已经可以开发 `Java` 程序了，但是在项目开发过程中往往使用集成开发工具（`IDE`）来编写 `Java` 程序，以方便项目管理，提高开发效率，而且使用 `IDE` 工具的代码辅助功能可以快速地输入程序代码。

`Eclipse` 是目前最流行的 `Java` 集成开发工具之一。`Eclipse` 为编程人员提供了一流的 `Java` 程序开发环境。它的平台体系结构是在插件概念的基础上构建的，插件是 `Eclipse` 平台最具特色的特征之一，也是其区别于其他开发工具的特征之一。

1.2.1 安装 Eclipse

(1) 下载 `Eclipse`。打开浏览器，在地址栏中输入 `http://www.eclipse.org/downloads/`，访问 `Eclipse` 官网。根据不同的操作系统选择不同版本的下载链接。例如，作者的操作系统是 32 位的 `Windows 7`，因此选择 `Windows 32 Bit` 选项。

(2) 安装中文包。`Eclipse` 官方网站下载的是英文版，为降低读者的学习难度，在启动 `Eclipse` 之前可安装中文语言包。在浏览器的地址栏中输入 `http://www.eclipse.org/babel/downloads.php`，打开的下载页面如图 1-9 所示。在 `Babel Language Pack Zips and Update Sites-R0.12.1` 标题下选择对应 `Eclipse` 版本的超链接下载语言包。`Eclipse 4.3` 的名称是 `Luna`，所以单击 `Luna` 超链接，进入 `Eclipse 4.3` 的 `Babel` 语言包下载页面。

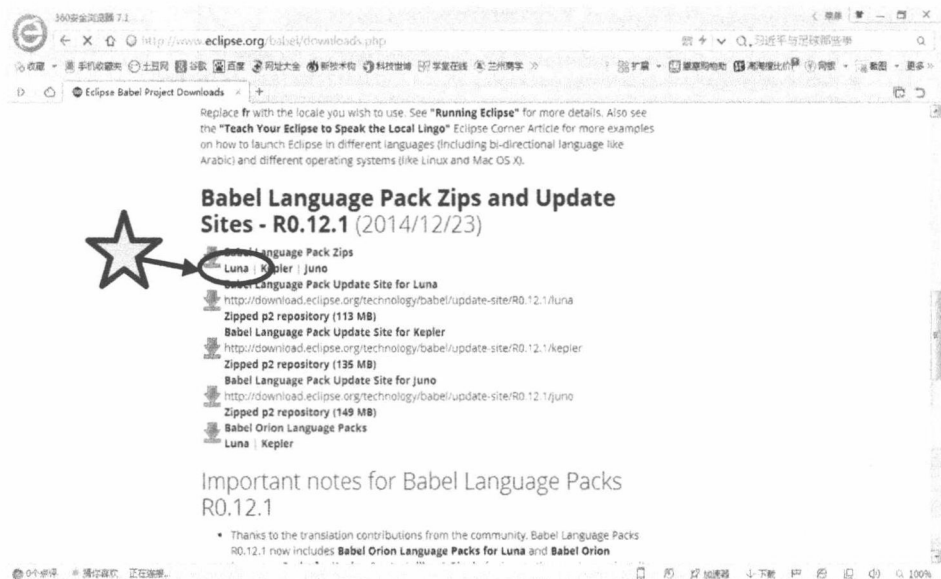


图 1-9 选择 Eclipse 版本

(3) 语言包下载页面包含对应各国语言的资源包，而每种语言的资源包又按插件与功能模块分为多个 zip 压缩包。单击图 1-10 中的 Chinese(Simplified)项，选择中文简体语言包，如图 1-11 所示。

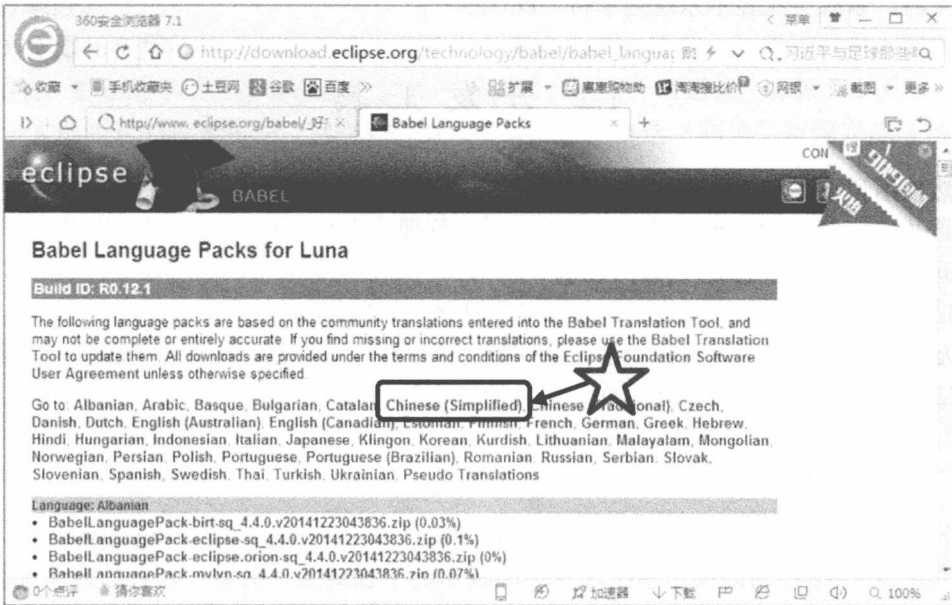


图 1-10 选择语言种类



图 1-11 下载语言包

(4) 将下载的所有语言包解压缩，并覆盖 Eclipse 文件夹中同名的两个文件夹 features 和 plugins，这样在启动 Eclipse 时就会自动加载语言包。

1.2.2 Eclipse 的启动与配置

在 Eclipse 的安装文件夹中运行 eclipse.exe 文件，即开始启动 Eclipse。Eclipse 在启动时会弹出“工作空间启动程序”对话框，该对话框用于设置 Eclipse 的工作空间（工作空间用于保存 Eclipse 建立的程序项目和相关设置）。本书的开发环境统一设置工作空间为 Eclipse 安装位置的 workspace 文件夹，在“工作空间启动程序”对话框的“工作空间”下拉列表框中输入“.\workspace”，如图 1-12 所示，单击“确定”按钮即可启动 Eclipse。

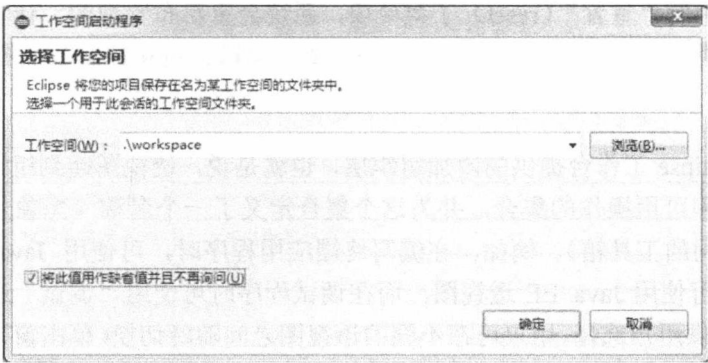


图 1-12 选择工作空间

1.2.3 Eclipse 工作台

Eclipse 首次启动时，会显示欢迎界面，其中包括 Eclipse 的概述、新增内容、样本、教程以及工作台相关按钮。Eclipse 工作台主要包括标题栏、菜单栏、工具栏、编辑器、透视图和相关视图等，如图 1-13 所示。注意工具栏中排在一起的三角形按钮和虫子按钮，它们分别表示运行程序（run）和调试（debug）程序。

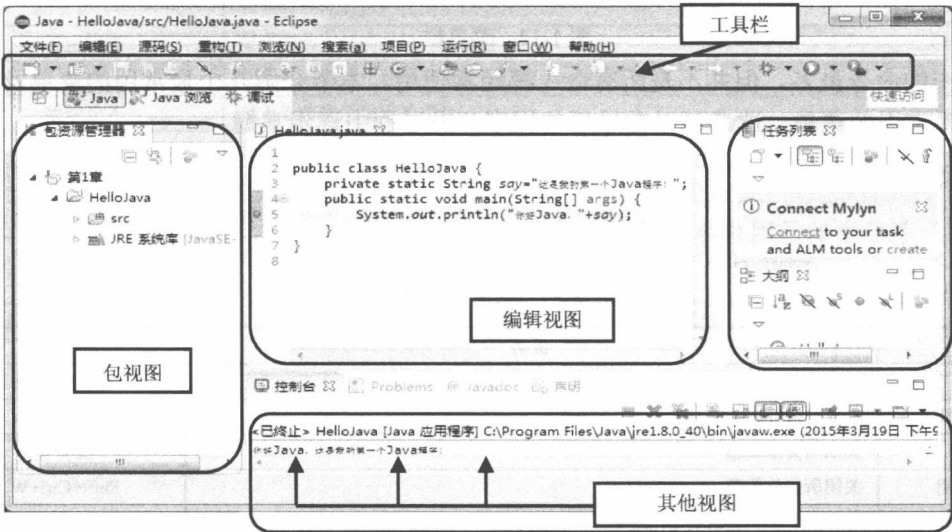


图 1-13 Eclipse 工作台

1.2.4 Eclipse 的透视图与视图

1. 视图

在图 1-13 中，用圆角矩形框标注的就是视图。各视图使用户可以从特定的角度查看或操作 Eclipse 中的源代码和其他数据。例如，“控制台”视图用于显示程序运行时的终端窗口中的信息，而“包资源管理器”视图便于在代码中导航，编辑视图自然是编辑源程序的窗口。每个视图都是一个小窗口，它们可以有自己独立的菜单和工具栏，并且可以通过拖拽随意改变布局位置、最小化或者被关闭。如果找不到某个视图，或者视图不在原来的位置上，则选

择“窗口”菜单中的“重置”(reset)子菜单项,系统会重新布置视图,让它们回到最初的位置上。视图的种类很多,初学者一开始用到的主要是编辑、包和控制台视图。

2. 透视图

透视图是 Eclipse 工作台提供的附加组织层,也就是说,透视图起到组织的作用,它实现多个视图的布局 and 可用操作的集合,并为这个集合定义了一个名称(想象一下,在修车和烹饪时显然需要不同的工具箱)。例如,在编写终端应用程序时,可使用 Java 透视图,在编写 Web 应用程序时可使用 Java EE 透视图,而在调试程序时可使用“调试”透视图。系统提供了近 20 种透视图供用户选择,用户可在不同的透视图之间随时切换(单击窗口右上角的按钮),但在同一时刻只能使用一个透视图。

1.2.5 Eclipse 的菜单栏

Eclipse 的菜单栏包含 Eclipse 的基本命令,在使用不同的编辑器时,还会动态地添加有关该编辑器的菜单。基本的菜单栏中除了常用的“文件”、“编辑”、“窗口”和“帮助”等菜单项以外,还提供了一些功能菜单,如“源码”和“重构”等,如图 1-14 所示。

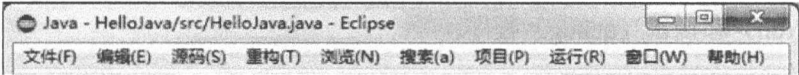


图 1-14 菜单栏

菜单中的命令虽多,但并不是所有命令都经常使用。本节介绍几个最常用的菜单及其命令,其他不常用的菜单,读者可以在日后的程序开发过程中逐步掌握。

1. “文件”菜单

“文件”菜单中的常用命令列于表 1-1 中。

表 1-1 “文件”菜单中的常用命令

命令	说明	快捷键
新建	创建新项目、元素或资源	Shift+Alt+N
打开文件	打开已经存在的文件	
关闭	关闭当前编辑器	Ctrl+W
全部关闭	关闭所有编辑器	Shift+Ctrl+W
保存	保存当前编辑器的内容	Ctrl+S
刷新	刷新所选元素的内容	F5
切换工作空间	切换工作空间到其他位置,这将导致 Eclipse 重启	
导入	打开导入向导对话框	
导出	打开导出向导对话框	
属性	打开所选元素的属性对话框	Alt+Enter

2. “编辑”菜单

“编辑”菜单用于辅助程序代码设计工作,“编辑”菜单中的常用命令列于表 1-2 中。

3. “源码”菜单

“源码”菜单中包含的命令都是和代码编写相关的一些命令,主要用于辅助编程,其功能说明如表 1-3 所示。

表 1-2 “编辑”菜单中的常用命令

命令	说明	快捷键
将选择范围扩展到	将选择编辑内容的范围扩大到外层元素、下一个元素、上一个元素或者恢复上一次选择的元素	
查找/替换	搜索编辑器中的内容片段，并根据需要替换为新的内容	Ctrl+F
查找下一个	搜索当前所选内容下一次出现的地方	Ctrl+K
查找上一个	搜索当前所选内容上一次出现的地方	Shift+Ctrl+K
添加书签	在当前光标所在行添加书签	
添加任务	在当前光标所在行添加任务	
灵活插入方式	切换插入方式。当禁用灵活插入方式时，将禁用自动缩进、添加右方括号等辅助功能	Shift+Ctrl+Insert
内容辅助	在当前光标位置打开内容辅助对话框	
文字补全	补全当前编辑器中正在输入的文字	Ctrl+Alt+/
快速修正	如果光标位于问题代码附近，则打开一个解决方案对话框	Ctrl+I

表 1-3 “源码”菜单中的常用命令

命令	说明	快捷键
切换注释	注释或取消注释当前选择的所有行	Ctrl+/
添加块注释	在当前选择的多行代码周围添加块注释	Shift+Ctrl+/
除去块注释	从当前选择的多行代码中除去块注释	Shift+Ctrl+\
更正缩进	更正当前选择的代码行的缩进	Ctrl+I
格式	使用代码格式化程序来格式化当前 Java 代码	Shift+Ctrl+F
组织导入	导入当前类所使用的类包	Shift+Ctrl+O
覆盖/实现方法	使用向导覆盖父类或实现接口中的方法	
生成 Getter 和 Setter	使用向导创建成员变量的 get×××()/set×××()方法	
生成 hashCode()和 equals()	打开“生成 hashCode()和 equals()”对话框	
使用字段生成构造函数	添加构造函数，这些构造函数初始化当前所选类型的字段，可用于类型、字段或类型中的文本选择	
从超类中生成构造函数	对于当前所选类型，按照超类中的定义来添加构造函数	
包围方式	使用代码模板包围所选语句	Shift+Alt+Z
外部化字符串	打开“将字符串外部化”向导，此向导允许通过使用语句访问属性文件来替换代码中的所有字符串	

4. “重构”菜单

“重构”菜单是 Eclipse 最关键的菜单，主要包括对项目重构的相关命令，是读者最应该掌握的。其功能说明如表 1-4 所示。

表 1-4 “重构”菜单中的常用命令

命令	说明	快捷键
重命名	重命名所选择的 Java 元素	Shift+Alt+R
移动	移动所选择的 Java 元素	Shift+Alt+V
抽取方法	创建包含当前所选语句或表达式的新方法，并相关地引用	Shift+Alt+M
抽取局部变量	创建为当前所选表达式指定的新变量，并将选择替换为对新变量的引用	Shift+Alt+L
抽取常量	从所选表达式创建静态终态字段并替换字段引用，并且可以选择重写同一表达式的其他出现位置	
内联	直接插入局部变量、方法或常量	Shift+Alt+I
将匿名类转换为嵌套类	将匿名内部类转换为成员类	

续表

命令	说明	快捷键
将成员类型转换为顶级	为所选成员类型创建新的 Java 编译单元，并根据需要更新所有引用	
将局部变量转换为字段	如果该变量是在创建时初始化的，则此操作将把初始化移至新字段的声明或类的构造函数	
抽取超类	从一组同类型中抽取公共超类	
抽取接口	根据当前类的方法创建接口，并使该类实现这个接口	
包括字段	将对变量的所有引用替换为 get×××()/set×××()方法	
历史记录	浏览工作空间重构历史记录，并提供用于从重构历史记录中删除重构的选项	

1.2.6 使用 Eclipse

下面用 Eclipse 来编写一个 Java 程序。

1. 创建 Java 项目

要在 Eclipse 中编写程序，必须先创建项目。Eclipse 中有很多种项目，其中 Java 项目用于管理和编写 Java 程序。创建该项目的步骤如下。

(1) 在菜单中选择“文件”→“新建”→“项目”命令（第一个是主菜单项，后面是子菜单项，以此类推），打开“新建项目”对话框。该对话框包含创建项目的向导，在向导中选择“Java 项目”节点，单击“下一步”按钮。

(2) 在“项目名”文本框中输入 HelloJava，在“项目布局”选项区中选中“为源文件和类文件创建单独的文件夹”单选按钮，如图 1-15 所示，单击“完成”按钮。

2. 创建 Java 类文件

创建 Java 类文件时，自动打开 Java 编辑器。创建 Java 类文件可以通过“新建 Java 类”向导来完成。在 Eclipse 菜单栏中选择“文件”→“新建”→“类”命令，打开“新建 Java 类”向导对话框，如图 1-16 所示。

使用该向导对话框创建 Java 类的步骤如下。

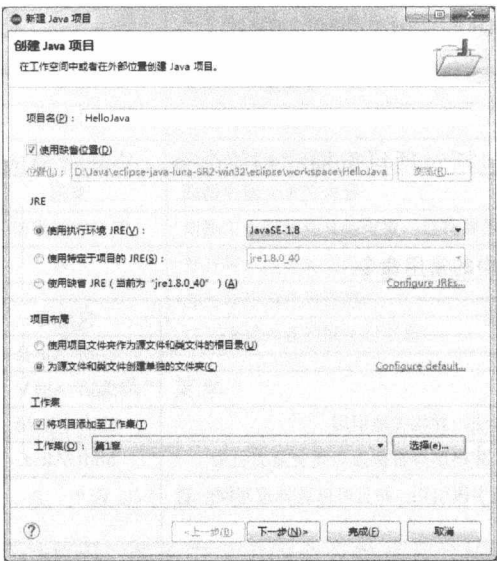


图 1-15 新建项目

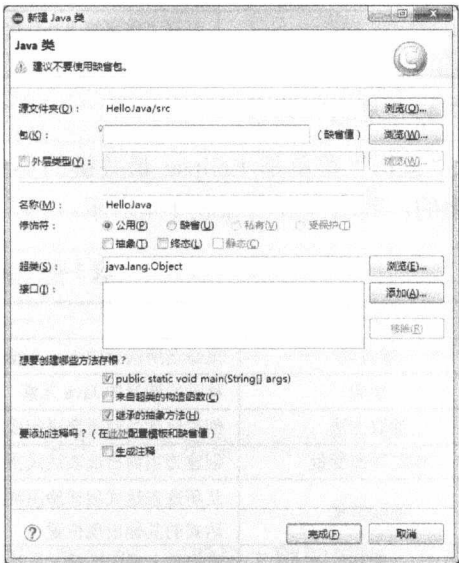


图 1-16 创建类文件

(1) 在“源文件夹”文本框中输入项目源程序文件夹的位置。通常向导会自动填充该文本框，没有特殊情况，不需要修改。

(2) 在“包”文本框中输入类文件想要使用的包名，这里暂时默认为空，不输入任何信息，这样就会使用 Java 工程的默认包。

(3) 在“名称”文本框中输入新建类的名称，如 HelloJava（注意：虽然 HelloJava 类名与 Java 项目同名，但是它们分别代表类文件和 Java 项目，读者需要注意区分它们的含义）。

(4) 选中 `public static void main(String[] args)` 复选框，向导在创建类文件时，会自动为该添加 `main()` 方法，使该类成为可以运行的主类。

3. 使用编辑器编写程序代码

编辑器总是位于 Eclipse 工作台的中间区域，该区域可以重叠放置多个编辑器。编辑器的类型可以不同，但是它们的主要功能都是完成 Java 程序、XML 配置等代码编写或可视化设计工作。

(1) 打开 Java 编辑器。在使用向导创建 Java 类文件之后，会自动打开编辑 Java 类文件的编辑窗口。除此之外，打开 Java 编辑器最常用的方法是在“包资源管理器”视图中双击 Java 源文件或者右击 Java 源文件，在弹出的快捷菜单中选择“打开方式”→“Java 编辑器”命令。

(2) 编写 Java 代码。在使用向导创建 HelloJava 类之后，向导会自动构建 HelloJava 类结构的框架代码，并且已经建立了 `main` 方法。程序开发人员需要做的就是将代码补全，为程序添加 `main` 的具体细节。本程序的完整代码如图 1-17 所示。

```
1 public class HelloJava{
2     private static String say="这是我的第一个 Java 程序! ";
3     public static void main(string[] args){
4         System.out.println("你好 Java, "+say);
5     }
6 }
```

图 1-17 编辑源程序

在 HelloJava 程序代码中。第 1 行、3 行、5 行、6 行是由向导创建的。完成这个程序只要编写第 2 行和第 4 行代码即可。

首先看第 2 行代码，它包括 `private`、`static`、`String` 3 个关键字，这 3 个关键字如果手动输入可能不会费多长时间，但是无法避免出现输入错误的情况，例如，将 `private` 关键字输入为 `privte`，缺少了字母 `a`，这个错误可能在程序编译时才会被发现。在 Eclipse 的 Java 编辑器中可以输入关键字的部分字母，然后使用 `Ctrl+Alt+/` 快捷键自动补全 Java 关键字，如图 1-18 所示。

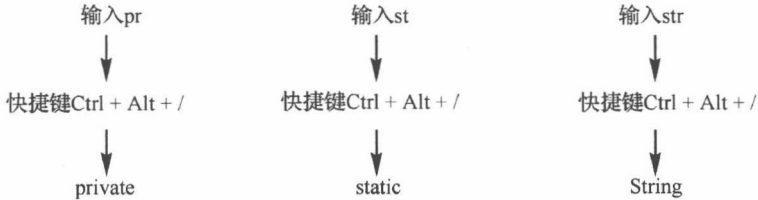


图 1-18 自动补全代码