

卓越工程师计划：软件工程专业系列丛书

Java

开发技术实践

李圣文 杨之江 龚君芳 编著



科学出版社

卓越工程师计划：软件工程专业系列丛书

Java 开发技术实践

李圣文 杨之江 龚君芳 编著

科学出版社

北 京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书在介绍了Java基本语法基础后,以主题方式展开,涉及数据库、图形化界面、多线程、网络编程、Web开发、分布式开发及其通用的SSH基础等内容。每个主题为一章,便于教学裁剪及展开。本书不注重Java的语法,力求以Java为例,较全面地介绍常用的开发技术以及企业应用开发的相关知识体系。

本书主要面向软件工程等信息类高年级学生和具有一定语言基础且准备自学Java语言的爱好者。

图书在版编目(CIP)数据

Java开发技术实践/李圣文,杨之江,龚君芳编著. —北京:科学出版社, 2015.1

(卓越工程师计划:软件工程专业系列丛书)

ISBN 978-7-03-042585-0

I. ①J… II. ①李… ②杨… ③龚… III. ①JAVA语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2014)第272409号

责任编辑:张颖兵 闫 陶/责任校对:肖 婷

责任印制:高 嵘/封面设计:陈明亮

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>

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2015年1月第一版 印张:15

2015年1月第一次印刷 字数:300 000

定价:36.00元

(如有印装质量问题,我社负责调换)

前 言

随着各种开发环境和技术的涌现,Java 被广泛应用到各种应用的开发中,从桌面应用程序到 Web 应用程序,进而扩展到 Android 等嵌入式应用的开发。在这些开发方法中,除了都利用了 Java 语言特性外,还或多或少地使用了 Java 的类库及一些扩展的开发技术。因而 Java 开发技术的学习,应该不仅限于 Java 语言语法的学习,更应该以相关技术为核心,建立完整的 Java 知识体系。

另外,软件工程等信息类专业的学生在学习了一门高级语言后,可能会对软件开发具有一定的兴趣,但是所具有的基础知识难以支撑其进行各种应用开发。很明显,高年级学生迫切需要一门能全面引入各种开发技术,以及在他们的基础和需求上有所帮助的课程。

本书的出发点就是建立在培养学生熟悉各种开发知识,建立企业应用开发知识体系的基础上,作为学生的后续软件实践及其软件项目开发的知识储备,也为软件架构、UML 和软件工程的教學提供理论和案例基础。

本书在编写中努力体现以下两点。

(1) 以建立知识体系为目标。以主题为核心,一个章节一个主题,分别介绍实际软件开发的各项技术,逐步建立学生在 Java 企业开发中的相关知识体系。

(2) 力求以实例为导向。每个知识点一个实例,力求实例简单,可实践性强,且对于同一功能在不同章节采用不同的技术实现,以引导学生理解各项技术,掌握其应用技巧。

另外,Java 相关技术开发具有一定的广泛性,对于其他语言相关技术的开发具有一定的借鉴意义。

本书共分 10 章,分别介绍 Java 开发工具与环境、Java 语言基础、数据库开发、图形用户界面、多线程、网络程序设计、Web 开发、Java EE 与分布式开发、SSH 集成框架以及 JForum 实例分析等。建议前 8 章各安排 4 学时,第 9 章 8 学时,共计 40 学时。

编者谨向一切关心和帮助过本书编写的同事和朋友致以衷心的感谢,并感谢科学出版社在教材编写、插图、版式设计等方面所提出的宝贵意见和所做的大量工作。

由于水平有限,时间仓促,疏漏和不妥之处在所难免,敬请指正。

编者

2014 年 3 月

目 录

第 1 章 开发工具与环境	1
1.1 JDK 开发环境	1
1.1.1 JDK 安装	1
1.1.2 环境变量设置	1
1.2 基于命令行的开发	2
1.2.1 第一个 Java 程序	2
1.2.2 Java 程序的编译	3
1.2.3 程序执行	4
1.2.4 程序调试	5
1.3 基于 Eclipse 的开发	5
1.3.1 Eclipse 简介	5
1.3.2 Eclipse 项目实践	7
1.3.3 程序编译错误的消除	10
1.3.4 库的引用	11
1.3.5 Java 应用程序调试实例	11
1.3.6 相关窗口设置	14
1.3.7 Eclipse 插件	14
1.4 JDK 常用工具	15
1.4.1 JAR	16
1.4.2 JavaDoc	16
1.5 习题	18
第 2 章 Java 语言基础	19
2.1 基本语法	19
2.1.1 标识符与关键字	19
2.1.2 基本数据类型	20
2.1.3 运算符	21
2.1.4 Java 包	21
2.2 Java 中的面向对象	22
2.2.1 对象引用	22
2.2.2 java.lang.Object	23
2.2.3 java.lang.String	24
2.2.4 包装类	24
2.2.5 数组	26

2.3 构建类	27
2.3.1 接口	27
2.3.2 构造与析构	28
2.3.3 访问控制	29
2.3.4 继承	29
2.3.5 嵌套类	30
2.4 反射	33
2.5 异常处理	35
2.6 容器类	36
2.7 命名规范	37
2.8 Java 与 C++	37
2.8.1 语法与词汇	37
2.8.2 类型、值、变量、表达式	38
2.8.3 Java 类与对象	39
2.8.4 继承和多态	41
2.9 习题	42
第 3 章 数据库开发	44
3.1 JDBC 介绍	44
3.2 JDBC API	45
3.2.1 Connection 对象	45
3.2.2 DriverManager 类	45
3.2.3 Statement 对象	46
3.2.4 ResultSet 对象	47
3.2.5 PreparedStatement	49
3.2.6 CallableStatement	51
3.3 Java 数据库——HSQLDB	53
3.3.1 简介	53
3.3.2 服务器模式启动示例	54
3.3.3 客户端工具	55
3.4 JDBC 实例	56
3.5 习题	58
第 4 章 图形用户界面	59
4.1 事件处理	59
4.1.1 事件处理机制	59
4.1.2 Frame 示例	60
4.1.3 使用适配器类简化代码	62
4.2 AWT 组件	64
4.2.1 按钮	64

4.2.2	标签	64
4.2.3	文本组件	65
4.2.4	复选框	65
4.2.5	单选按钮	66
4.2.6	列表	66
4.2.7	滚动条	67
4.3	Swing	67
4.3.1	Swing 组件	67
4.3.2	组件的显示	69
4.3.3	JOptionPane	71
4.4	容器与布局	72
4.4.1	Swing 容器	73
4.4.2	JPanel 和 JScrollPane	74
4.4.3	布局	75
4.5	JNotePad 实践	77
4.6	习题	86
第 5 章	多线程	87
5.1	Java 中的线程	87
5.1.1	操作系统与进程	87
5.1.2	进程与线程	87
5.1.3	Java 的多线程机制	88
5.2	线程创建	89
5.2.1	Thread 类	89
5.2.2	实现 Runnable 接口	89
5.2.3	线程的结束与常用方法	90
5.3	线程同步	91
5.3.1	不相关的线程	91
5.3.2	相关但无须同步的线程	92
5.3.3	互斥线程	94
5.3.4	竞争条件问题	95
5.3.5	代码块的同步	96
5.3.6	相互通信的互斥线程	98
5.4	线程本地存储区	103
5.5	习题	105
第 6 章	网络程序设计	106
6.1	网络基础	106
6.1.1	IP 地址	107
6.1.2	TCP 和 UDP	108

6.2	Socket 基础	111
6.2.1	连接到服务器	111
6.2.2	发送邮件	114
6.2.3	服务器端套接字	115
6.3	HTTP 服务器实例开发	117
6.3.1	简单服务器程序	118
6.3.2	获取 HTTP 命令	119
6.3.3	支持并发的 HTTP 服务器	121
6.4	URL 类	122
6.5	习题	124
第 7 章	Web 开发基础	125
7.1	Web 基础	125
7.1.1	Post 与 Get	125
7.1.2	Web 项目实践	126
7.2	Servlet	130
7.2.1	Servlet 概述	131
7.2.2	Servlet 生命周期	131
7.2.3	Servlet 实践	132
7.3	JSP	137
7.3.1	JSP 概述	137
7.3.2	JSP 内置对象	138
7.3.3	JSP 实践	139
7.4	习题	142
第 8 章	Java EE 与分布式开发	143
8.1	分布式多层应用程序	143
8.1.1	安全性	144
8.1.2	Java EE 组件	144
8.1.3	Java EE 客户端	145
8.1.4	Web 组件	146
8.1.5	业务组件	147
8.1.6	企业信息系统层	148
8.1.7	Java EE 容器	148
8.2	Java EE 开发体系	149
8.2.1	Java EE 的核心 API 与组件	149
8.2.2	开发角色	151
8.2.3	Java EE 应用组装和部署	152
8.3	EJB 与企业 Bean	153
8.3.1	企业 Bean 的优点	154

8.3.2 企业 Bean 的分类	154
8.4 无状态会话 Bean 实践	157
8.4.1 创建无状态会话 Bean	157
8.4.2 Web 客户端	163
8.5 远程方法调用	170
8.5.1 RMI 原理	170
8.5.2 RMI 实例	171
8.6 Web 服务	173
8.6.1 XML	173
8.6.2 SOAP	174
8.6.3 WSDL 标准格式	174
8.6.4 Web Service 实践	174
8.7 习题	179
第 9 章 SSH 集成框架	180
9.1 Spring	180
9.1.1 IOC	181
9.1.2 AOP	187
9.2 ORM 与 Hibernate	192
9.2.1 Hibernate 概述	192
9.2.2 Hibernate 原理	193
9.2.3 开发示例	194
9.3 Struts	200
9.3.1 MVC 思想概述	200
9.3.2 Struts 开发实践	201
9.4 SSH 整合实践	207
9.4.1 概述	207
9.4.2 整合 Spring 与 Struts	208
9.4.3 集成 Hibernate	210
9.5 习题	213
第 10 章 JForum 实例分析	214
10.1 系统部署及功能介绍	214
10.1.1 下载与安装	214
10.1.2 JForum 功能介绍	217
10.2 源码分析	220
10.2.1 框架配置文件分析	220
10.2.2 MVC 和 ORM 实现	220
10.2.3 JForum 处理请求流程	220
10.2.4 文件监视	223

10.2.5	数据库访问	223
10.2.6	权限配置	223
10.3	二次开发	223
10.3.1	常用开发包	223
10.3.2	关键类	224
10.3.3	实例开发	224
参考文献		228

第 1 章 开发工具与环境

“工欲善其事，必先利其器”，各种工具在程序开发中的地位显得非常重要。本章分别从 JDK、基于命令行的开发、集成环境下的开发及其常用开发工具等方面介绍 Java 开发过程与步骤。

1.1 JDK 开发环境

Java 标准版开发工具包 (Java Standard Edition Development Kit, JDK) 是 Java 开发和运行的基本平台。换句话说，所有用 Java 语言编写的程序要编译和运行都离不开它，Java 集成开发环境也是建立在 JDK 的基础上。

1.1.1 JDK 安装

下载 JDK 可以访问官方网站 <http://www.oracle.com/technetwork/java>，一般来说，根据自己的操作系统平台下载最新版本即可，目前的稳定版本是 JDK 7。

Windows 下可以直接执行安装文件完成安装。

不要下载 Java 运行时环境 (Java Runtime Environment, JRE)，因为 JRE 不包含 Java 编译器和 JDK 类的源码。

1.1.2 环境变量设置

环境变量设置不是必须的，如果打算使用集成开发环境 (如 Eclipse) 来进行开发，而不是手工编译代码，可以完全忽略本节内容。

第一个需要配置的环境变量是 JAVA_HOME。在桌面上的“我的电脑”上右击，在弹出的快捷菜单中选择“属性”选项，在弹出的对话框中切换到“高级”选项卡，然后单击“环境变量”按钮，如图 1-1 所示。

在出现的“环境变量”对话框的“系统变量(S)”列表框下方单击“新建”按钮，出现新建系统环境变量的对话框，输入变量名 JAVA_HOME，变量值为 JDK 安装目录，例如，D:\Program Files\Java\jdk1.6.0_13 (例如，Tomcat 需要这个环境变量来查找 JDK)。

接下来用类似的方法新建环境变量 CLASSPATH，取值为“.”，这个变量用来供 Java 虚拟机查找要加载的类。

把 JDK 的应用程序路径添加到系统的 PATH 变量中，拖动滚动条找到列表中名为 path 的变量，点击“编辑”按钮，即可修改 PATH 的变量值。一般来说只需要在开头加“%JAVA_HOME%\bin;”，如图 1-2 所示。



图 1-1 “系统属性”对话框



图 1-2 环境变量设置

在弹出的“环境变量”对话框中选中 path 变量,然后单击“编辑”按钮即可修改环境变量,如图 1-3 所示。

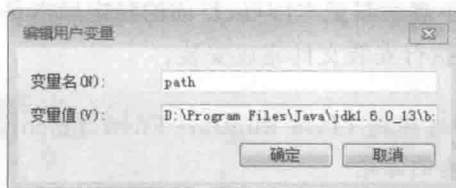


图 1-3 修改环境变量

编辑变量值,增加 JDK 下的 bin 目录,如 D:\Program Files\Java\jdk1.6.0_13\bin。

1.2 基于命令行的开发

正确安装了 JDK 后,就可以进行 Java 程序的开发了。虽然现在大多数情况下 Java 的开发已经采用了更为方便的集成环境,但是基于 JDK 的手工开发知识还是不可或缺的。一方面因为集成开发环境的功能有限,大多数 Java 系统部署、优化、分布式调试等需要手工方式的参与。另一方面由于集成开发环境是对 JDK 工具的封装,JDK 是集成开发环境的有机组成部分,所以这方面知识也是用户掌握开发集成开发环境的基础。

1.2.1 第一个 Java 程序

使用记事本或者其他文本编辑器编写以下内容,然后将其保存为 Hello.java 文件,并保证该文件在 firstdemo 目录下。

与其他语言的介绍类似,第一个程序将输出“Hello World!”,其中每一行的前两个字

符是行标注,不属于 Java 代码的内容,完整代码实例如下:

```
1. package firstdemo;  
2. public class Hello  
3. {  
4.     public static void main(String[] args)  
5.     {  
6.         System.out.println("Hello World!");  
7.     }  
8. }
```

(1) 代码第 1 行声明了包 firstdemo,说明此部分代码都在 firstdemo 名称空间中,并且本代码文件必须放到 firstdemo 目录中。

(2) 代码第 2 行声明了类 Hello,并且声明为 public 型,表示对包外面可见,即可从包外面访问(使用)这个类,构建类的对象。由于 Java 要求公有类要与所在文件名相同,所以该代码必须放入 Hello.java 文件中。

(3) 代码第 4 行声明了一个公有的静态的 main()函数,该函数接受 String 数组格式的参数。这里给出的是 Java 程序主入口的标准写法,也是唯一的写法。在 Java 语言中,没有全局函数,程序主入口必须写在一个公有的类中,并且以静态函数的方式声明。其中 args 用来获取启动此程序时传递的参数。

(4) 代码第 6 行调用系统库中 out 对象的 println()方法,在控制台或者输出窗口输出“Hello World!”。

1.2.2 Java 程序的编译

(1) 打开命令行。在“运行”对话框的“打开”框中输入 cmd,如图 1-4 所示,单击“确定”按钮,弹出命令行界面,如图 1-5 所示。

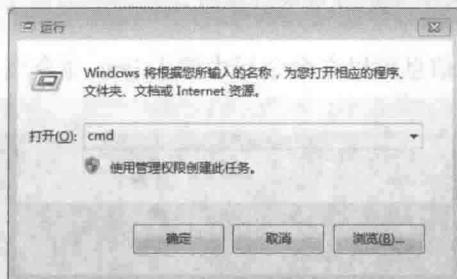


图 1-4 运行命令

(2) 使用命令行编译程序。在命令行中切换到 firstdemo 所在目录,然后使用 javac 命令编译上面编写的 Hello.java。示例如下:

```
javac firstdemo\Hello.java
```

编译后会生成 Hello.class 文件。更多使用方式可以直接使用命令行中输入 Javac 查看相关帮助。

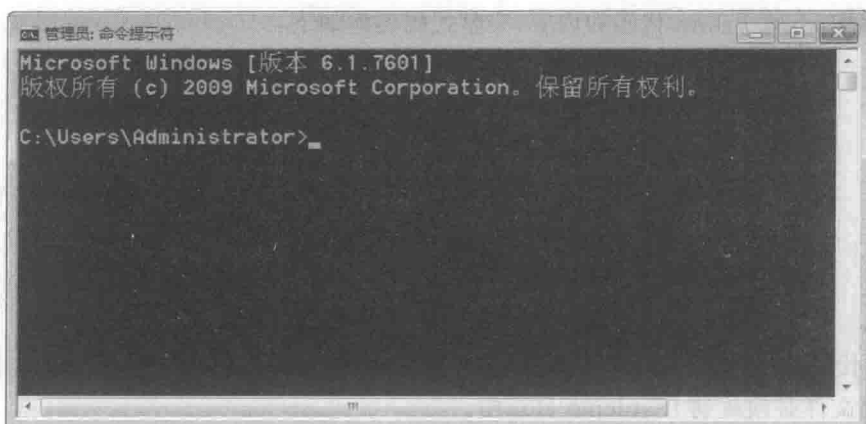


图 1-5 命令行界面

如果运行时程序提示“不是内部或外部命令,也不是可运行的程序或批处理文件”,那么应检查是否已经正确安装 JRE 和 JDK,同时是否正确配置了 path 环境变量。

1.2.3 程序执行

在命令行输入以下命令运行上面编译后的 Hello.class,执行结果如下:

```
D:\> Java firstdemo.Hello  
Hello World
```

如果出现错误提示“Exception in thread "main" java.lang.NoClassDefFoundError: Test”,则可能是在命令行下不存在 firstdemo 目录,或者 firstdemo 目录下不存在相应的.class 文件,也可能是大小写拼写错误引起的。

Java 命令的详细帮助信息可以在命令行中输入 java 命令完成,如图 1-6 所示。

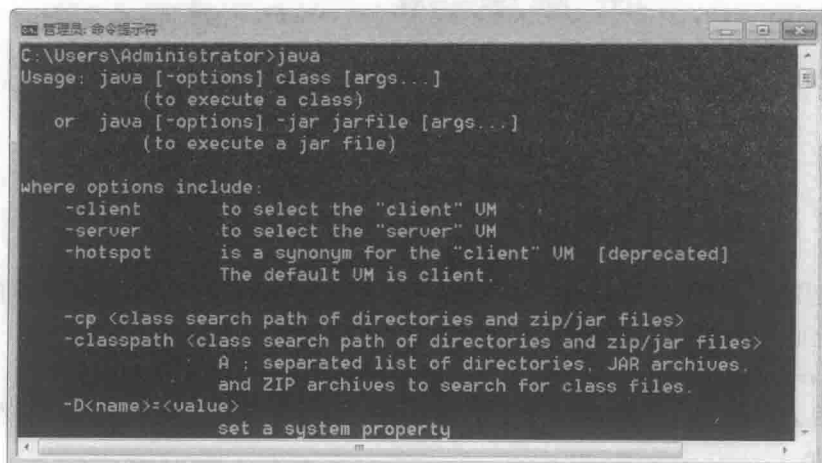


图 1-6 Java 命令提示

1.2.4 程序调试

在命令行输入 jdb, 进入 JDB 模式, 可以直接输入 help 查看详细调试方法, 如图 1-7 所示。

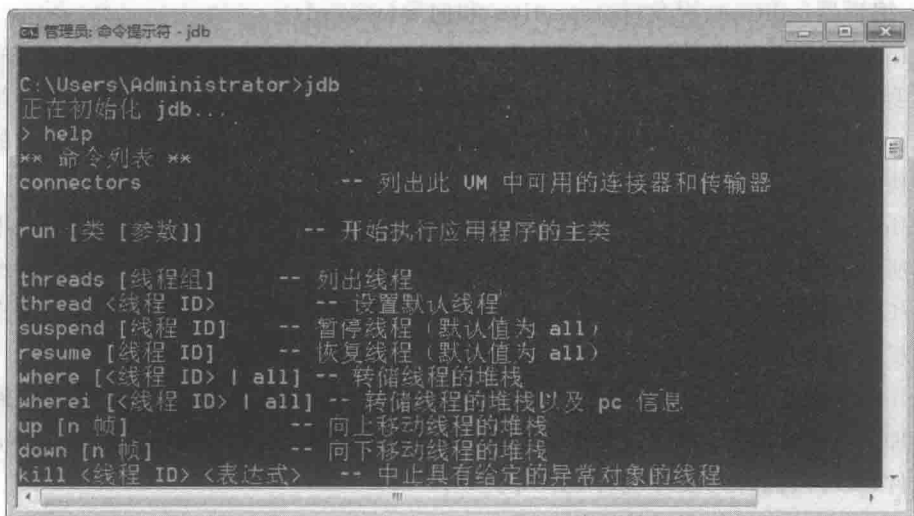


图 1-7 JDB 命令示例

1.3 基于 Eclipse 的开发

集成开发环境(Integrated Development Environment, IDE)是用于提供程序开发环境的应用程序,一般包括代码编辑器、编译器、调试器和图形用户界面工具,集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。Java 有比较多的集成开发环境,用户较多的有 Eclipse、Netbeans、JCreator 等。

1.3.1 Eclipse 简介

Eclipse 是著名的 IDE,最初主要用来进行 Java 语言开发,但是目前也有人通过插件使其作为其他计算机语言如 C++ 和 Python 的开发工具。

Eclipse 本身只是一个框架平台,但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

Eclipse 最初是由 IBM 公司开发的替代商业软件 Visual Age for Java 的下一代 IDE 开发环境,2001 年 11 月贡献给开源社区,现在它由非营利软件供应商联盟 Eclipse 基金会(Eclipse Foundation)管理。2003 年,Eclipse 3.0 选择 OSGi 服务平台规范为运行时架构。2012 年 6 月发布版本 4.2,代号 Juno。

Eclipse 的官方网站地址是 <http://www.eclipse.org>,本书示例采用 Eclipse IDE for

Java EE Developers 4.2 版本。

Eclipse 的基础是富客户机平台(Rich Client Platform, RCP), RCP 包括下列组件: 核心平台(启动 Eclipse, 运行插件); OSGi(标准集束框架); SWT(可移植构件工具包); JFace(文件缓冲、文本处理、文本编辑器); Eclipse 工作台, 即 Workbench, 包含视图(view)、编辑器(editor)、视角(perspective)和向导(wizard)。

Eclipse 采用的技术是 IBM 公司开发的 SWT, 这是一种基于 Java 的窗口组件, 类似 Java 本身提供的 AWT 和 Swing 窗口组件, 不过 IBM 声称 SWT 比其他 Java 窗口组件更有效率。Eclipse 的用户界面还使用了 GUI 中间层 JFace, 从而简化了基于 SWT 的应用程序的构建。

Eclipse 的插件机制是轻型软件组件化架构。在富客户机平台上, 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。已有的分离的插件已经能够支持 C/C++(CDT)、Perl、Ruby、Python、Telnet 和数据库开发。插件架构能够支持将任意的扩展加入现有环境中, 如配置管理, 而绝不仅限于支持各种编程语言。

Eclipse 的设计思想是“一切皆插件”。Eclipse 核心很小, 其他所有功能都以插件的形式附加于 Eclipse 核心之上。Eclipse 基本内核包括图形 API (SWT/JFace)、Java 开发环境插件(JDT)、插件开发环境(PDE)等。

Eclipse 由以下几种不同的计划组成。

(1) Eclipse 计划: 本身包括 Eclipse 平台、Eclipse 富客户端平台和 Java 开发工具。

(2) Eclipse 测试和性能工具平台(TPTP): 提供一个允许软件开发者构建诸如测试调试、概况分析、基准评测等测试和性能工具的平台。

(3) Eclipse Web 工具平台计划(WTP): 用 Java 企业版 Web 应用程序开发工具来扩展 Eclipse 平台。它由以下部分组成: HTML、JavaScript、CSS、JSP、SQL、XML、DTD、XSD 和 WSDL 的源代码编辑器; XSD 和 WSDL 的图形界面编辑器; Java 企业版的“项目性质”(projectnature)、建构器(builder)和模型(model), 以及一个 Java 企业版的导航(navigator); 一个 Web 服务(Web service)向导和浏览器, 还有一个 WS-I 测试工具; 最后是数据库访问查询的工具与模型。

(4) Eclipse 商业智能和报表工具计划(BIRT): 提供 Web 应用程序(特别是基于 Java 企业版的)的报表开发工具。

(5) Eclipse 可视化界面编辑器计划(VEP): 一个 Eclipse 下创建图形用户界面代码生成器的框架。

(6) Eclipse 建模框架(EMF): 依据使用 XMI 描述的建模规格, 生成结构化数据模型的工具和其他应用程序的代码。

(7) 图形化编辑器框架(GEF): 能让开发者采用一个现成的应用程序模型来轻松地创建富图形化编辑器。

(8) UML2: Eclipse 平台下的一个 UML 2.0 元模型的实现, 用以支持建模工具的开发。

(9) AspectJ:一种针对 Java 的面向切片语言扩展。

(10) Eclipse 通信框架(ECF):专注于在 Eclipse 平台上创建通信应用程序的工作。

(11) Eclipse 数据工具平台计划(DTP)。

(12) Eclipse 设备驱动软件开发计划(DSDP)。

(13) C/C++ 开发工具计划(CDT):努力为 Eclipse 平台提供一个全功能 C 和 C++ 的集成开发环境,它使用 GCC 作为编译器。

(14) Eclipse 平台 COBOL 集成开发环境子计划(COBOL):将构建一个 Eclipse 平台上的全功能 COBOL 集成开发环境。

(15) 并行工具平台(PTP):将开发一个对并行计算机架构下的一组工具进行集成的并行工具平台,而且这个平台是可移植的、可伸缩的,并基于标准的。

(16) 嵌入式富客户端平台(eRCP):计划将 Eclipse 富客户端平台扩展到嵌入式设备上,这个平台主要是一个富客户端平台组件子集的集合,它使得桌面环境下的应用程序模型能够大致同样地运用在嵌入式设备上。

1.3.2 Eclipse 项目实践

1. 选择工作区

启动 Eclipse,指定 Java 工作区所在目录。Eclipse 以目录为工作区,工作区中可以包含多个项目,工作区选择界面如图 1-8 所示。

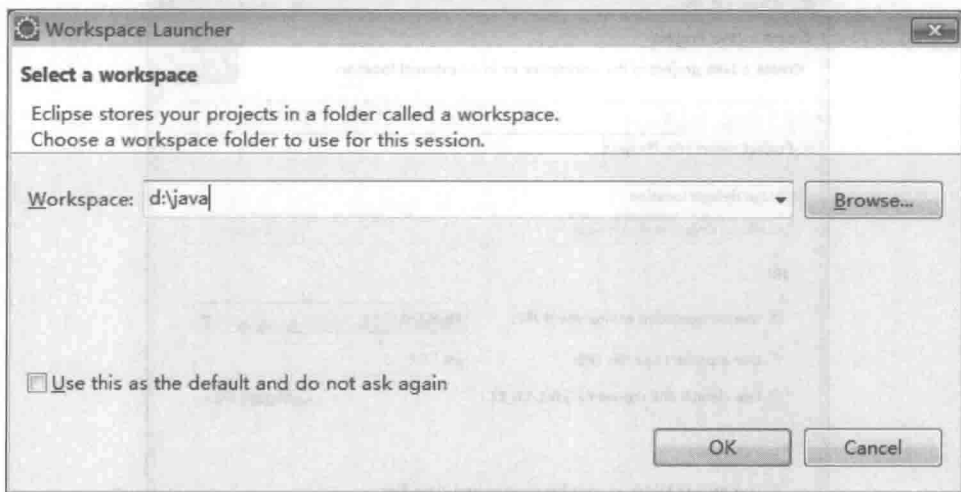


图 1-8 选择工作区

2. 新建项目

通过执行 File→New→Project...命令打开创建项目的向导,如图 1-9 所示。

单击 Next 按钮,进入输入项目信息的界面,在 Project Name 文本框中输入 FirstProject,其他内容保持缺省值,如图 1-10 所示,单击 Finish 按钮完成项目的创建。